BDRPC187**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG** 

**ĐỒ ÁN LẬP TRÌNH ỨNG DỤNG MẠNG**

Đề tài:

**Ứng dụng “Nuôi bò số” trên hệ điều hành Android**

**Giảng Viên Hướng Dẫn: Thạc sĩ Vũ Trí Dũng**

**Nhóm sinh viên thực hiện:**

1. **12520018 - Đinh Nhật Băng**
2. **12520677 - Huỳnh Đoàn Hữu Phúc**

***TP. Hồ Chí Minh, 2 tháng 6 năm 2016***

**MỞ ĐẦU**

Trong sự phát triển ko ngừng của xã hội thì nhu cầu của con người về cuộc sống ngày càng cao. Từ ý nghĩa thực tế đó mà đã nảy sinh ra nhiều loại hình dịch vụ để phần nào đáp ứng được nhu cầu thực tế của con người. Các dịch vụ đó đa phần về nhiều mặt với nhiều cách đáp ứng khác nhau. Và 1 loại hình dịch vụ giúp người dân cải thiện được khả năng cũng như hiệu suất làm việc ví dụ các ứng dụng quản lý sách, quản lý học sinh, sinh viên ra đời. Giúp chúng ta tiết kiệm thời gian cũng như giảm thiểu chi phí cũng như tăng năng suất làm việc ...

Ngày nay, khi cuộc sống của mọi người ngày càng được cải thiện, cho nên nhu cầu giải trí của con người cũng tăng lên. Bên cạnh đó, số lượng người sở hữu smartphone ngày càng tăng lên, cho nên, ứng dụng này ra đời nhằm mục đích cải thiện chất lượng cuộc sống của người nông dân, tăng khả năng làm việc, giảm thiểu rủi ro trong công việc. Và không chỉ đơn thuần là chỉ phục vụ nhu cầu quản lý, chăm sóc bò. Ứng dụng này còn tạo nên 1 cộng đồng lớn mạnh, qua đó các thành viên có thể tương tác, chia sẻ với nhau, trao đổi cách thức nuôi bò, cách chăm sóc và phòng chóng bệnh, hoặc có thể bán sản phẩm của mình thông qua ứng dụng.

Đồ án Ứng dụng “Nuôi bò số” trên Android bao gồm 4 chương:

**Chương I: Giới thiệu** – Giới thiệu khái quát về đồ án và mục đích của đồ án.

**Chương II: Phân tích và thiết kế hệ thống** – Phân tích, thiết kế và phát triển hệ thống, bao gồm: tổng quan về hệ thống, yêu cầu của hệ thống, phân tích và thiết kế hệ thống.

**Chương III: Công nghệ** – Giới thiệu đầy đủ về công nghệ và phần mêm được sử dụng trong đồ án bao gồm FULL stack application.

Server : Php với Yii FrameWork

Client : Java trên Android.

Cơ sở dữ liệu : MySQL.

**Chương IV: Ứng dụng “Nuôi bò số”** – Trình bày về giao diện, sơ đồ chức năng và sơ đồ xử lý chức năng của ứng dụng.

Kết luận: Tổng kết quá trình thực hiện đồ án và rút ra hướng phát triển sau này.

**LỜI CẢM ƠN**

Nhóm xin trân trọng cám ơn sự giúp đỡ tận tình của các thầy cô thuộc trường Đại học Công Nghệ Thông Tin – Đại học quốc gia thành phố Hồ Chí Minh đã nhiệt tình giúp đỡ nhóm trong quá trình thực hiện đồ án. Đặc biệt là sự hỗ trợ, hướng dẫn tận tình của thầy Vũ Trí Dũng – người trực tiếp hướng dẫn nhóm thực hiện đồ án này.

Trong quá trình thực hiện khóa luận, mặc dù nhóm đã cố gắng hoàn thành một cách tốt nhất nhưng cũng ko thể tránh khỏi những thiếu sót. Nhóm kính mong nhận được sự đóng góp quý báu từ các thầy cô để khóa luận được hoàn thiện hơn.

Nhóm sinh viên thực hiện:

Đinh Nhật Băng – Huỳnh Đoàn Hữu Phúc

Tháng 06 - 2015

**MỤC LỤC**

**MỞ ĐẦU** 2

**LỜI CẢM ƠN** 3

**MỤC LỤC** 4

**CHƯƠNG I: GIỚI THIỆU**

* 1. Giới thiệu đề tài 6
  2. Mục tiêu đề tài 6

**CHƯƠNG II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

**2.1 Giới thiệu**......................................................................................................7

2.1.1 Tổng quan về hệ thống.......................................................................7

2.1.2 Mục tiêu và phạm vi của hệ thống.....................................................7

**2.2 Yêu cầu hệ thống**.........................................................................................7

2.2.1 Xác định yêu cầu................................................................................7

2.2.2 Các yêu cầu chức năng.......................................................................8

2.2.3 Use Case Diagram..............................................................................9

2.2.4 Use Case Narrative...........................................................................10

**CHƯƠNG III: CÔNG NGHỆ**

**3.1 Giới thiệu về PHP YII Framework** ……………………………………28

3.1.1 Ưu và nhược điểm............................................................................28

3.1.2 Tại sao nên sử dụng Yii framework ................................................29

**3.2 Giới thiệu về MySQL**................................................................................34

3.2.1 MySQL là gì.....................................................................................34

3.2.2 Cách thức lưu trữ dữ liệu của MySQL DB.......................................37

**3.3 VPS**..............................................................................................................38

3.3.1 Tổng quan về DigitalOcean..............................................................39

3.3.2 Khả năng hoạt động, chịu tải, nâng cấp của VPS.............................40

**3.4 Lập trình Android với Android Studio**....................................................43

3.4.1 Android Studio là gì ?.......................................................................43

3.4.2 Xây dựng ứng dụng Android với Android Studio.............................44

**CHƯƠNG IV: ỨNG DỤNG NUÔI BÒ SỐ**

4.1 Giới thiệu về ứng dụng “Nuôi bò số”...........................................................47

4.2 Thiết kế giao diện, các chức năng của ứng dụng..........................................47

4.3 Cấu trúc CSDL..............................................................................................61

**KẾT LUẬN**........................................................................................................................65

**TÀI LIỆU THAM KHẢO**................................................................................................66

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**..........................................................67

**CHƯƠNG I: GIỚI THIỆU**

* 1. **Giới thiệu đề tài**

Ngày nay, khi cuộc sống của mọi người ngày càng được cải thiện, mọi công việc ngày càng được đơn giản hóa nhưng đem lại chất lượng ngày càng được nâng cao. Bên cạnh đó, số lượng người sở hữu smartphone ngày càng tăng lên, theo thống kê số lượng người sử dụng smartphone chiếm 52% tổng số người sử dụng di động (theo thống kê của Nelsen 2014) và con số này ngày càng tăng nhanh. Con số ngày cho thấy lượng người sử dụng smartphone ngày càng nhiều, bất kể bất kì ngành nghề hay độ tuổi nào. Chính vì vậy ứng dụng này ra đời nhằm hỗ trợ những người nông dân tiết kiệm được thời gian, chi phí quản lý bò. Nhờ đó tăng năng suất lao động của những người nông dân, cải thiện được chất lượng cuộc sống.

Về mặt lý thuyết, có thể nói ứng dụng này có như 1 số ứng dụng quản lý ở các trung tâm thành phố như ứng dụng quản lý học sinh, sinh viên, quản lý thư viện, v…v…. . Thế nhưng điều đặc biệt của ứng dụng là nó có khả năng tận dụng được sự tiện lợi của các thiết bị di động. Hơn thế nữa là nó có thể sử dụng mọi lúc mọi nơi, không nhất thiết phải sử dụng máy tính cồng kềnh khó di chuyển.

* 1. **Mục tiêu đề tài**

Xây dựng nên 1 ứng dụng quản lý, chăm sóc bò cho người sử dụng điện thoại Android ( Tương lai có thể phát triển iOS). Nhằm phục nhu cầu đổi mới, phát triển của người nông dân. Người nông dân có thể kiểm tra được quá trình phát triển của từng con bò, đặc điểm của từng con bò, từ đó có thể chăm sóc 1 cách hiệu quả và tiết kiệm nhất. Ngoài ra , ứng dụng còn có thể giúp cho người sử dụng có thể trao đổi, buôn bán trên chính chiếc điện thoại của mình mà không phải qua các bên trung gian. Giúp đơn giản hóa việc mua bán, tạo hiệu quả kinh tế vô cùng lớn.

**CHƯƠNG II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

**2.1 Giới thiệu**

Ứng dụng được xây dựng theo mô hình và chức năng giống các hệ thống quản lý thư viện, sinh viên. Tuy nhiên, phù hợp theo đề tài mà có 1 số chức năng đặc trưng trên điện thoại smartphone mới có.

**2.1.1 Tổng quan về hệ thống.**

Có hệ thống đăng nhập, đăng ký và server lưu trữ CSDL. Người dung khi sử dụng có thể xem danh sách các con bò của mình, có thể tạo bò, tìm kiếm bò theo mã QR code hoặc NFC tag, xem thư viện hướng dẫn nuôi bò, xem danh sách công việc trong ngày, danh sách các con bò cần tiêm phòng trị bệnh, đăng thông tin bán bò và xem danh sách các con bò bán trên chợ bò. Về phần admin có thể đăng thông tin trong thư viện hướng dẫn nuôi bò, kiểm duyệt nội dung trong chợ bò xà xử lý vi phạm.

**2.1.2 Mục tiêu và phạm vi hệ thống**

Ứng dụng xây dựng phục vụ cho tất cả những người dùng điện thoại Android đang nuôi bò trên đất nước Việt Nam. Đặc biệt là những người nông dân hay cho các hệ thống nuôi bò với số lượng lớn.

**2.2 Yêu cầu hệ thống**

Ứng dụng phải đảm bảo cung cấp đầy đủ cho người dùng những công cụ cũng như server hỗ trợ tốt nhất khi người dùng sử dụng.

**2.2.1 Xác định yêu cầu.**

Ứng dụng được xây dựng dựa trên giao diện và chức năng quản lý sinh viên viên nên phải tích hợp được tất cả các chức năng cơ bản của quản lý như tìm kiếm, thêm mới, chỉnh sửa và xóa. Ngoài ra còn có các chức năng mới như tìm kiếm theo QR code hoặc NFC tag.

**2.2.2 Các yêu cầu chức năng**

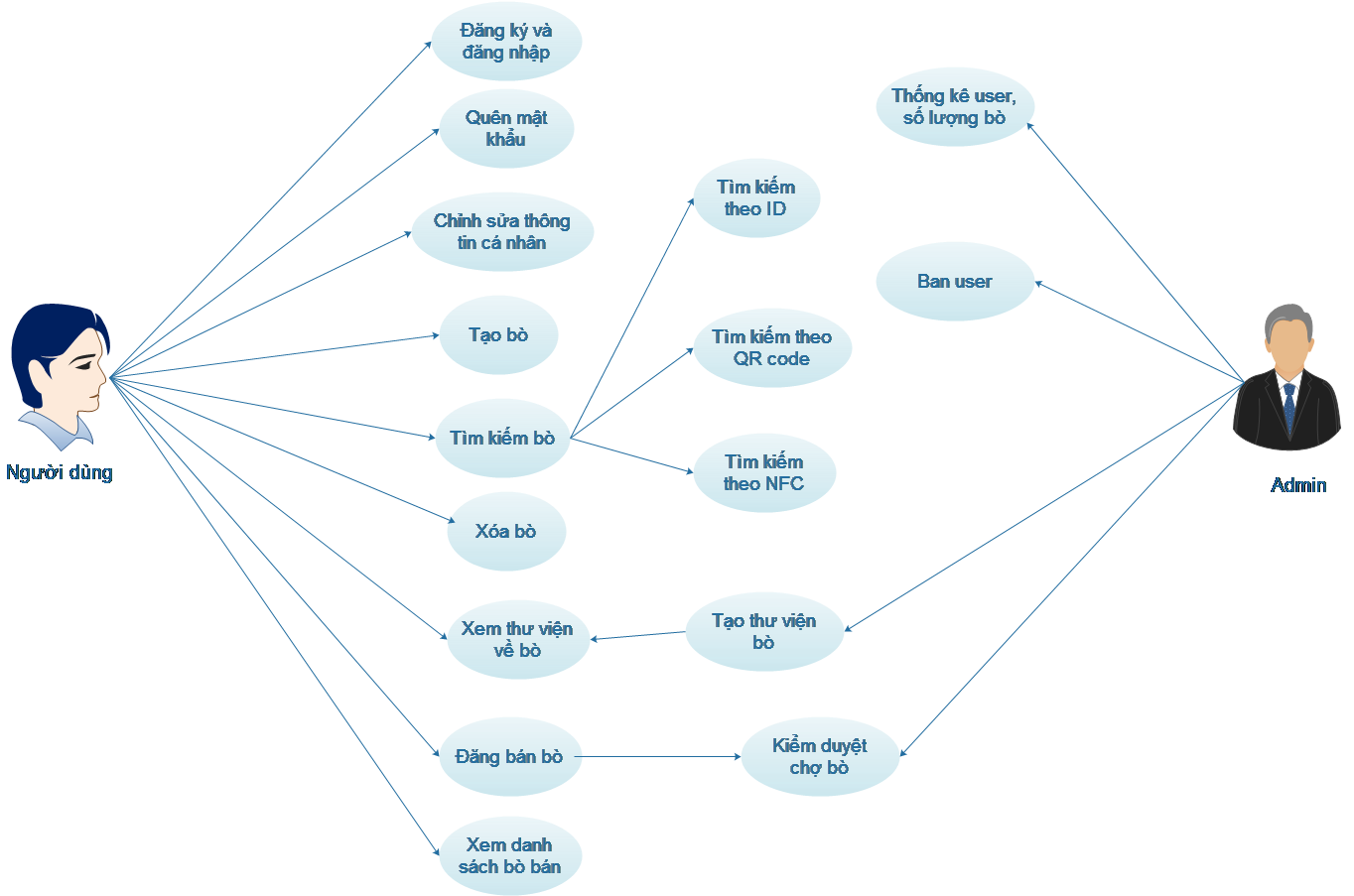
Nhóm chức năng đăng ký, đăng nhập, quên mật khẩu, chỉnh sửa thông tin cá nhân.

Nhóm chức năng tạo bò, xem danh sách bò, xem chi tiết từng con bò, thêm bò và xóa bò.

Nhóm chức năng xem thông tin trong thư viện bò.

Nhóm chức đăng thông tin bò để bán, xem bò trong danh sách chợ bò.

**2.2.3 Use Case Diagram**



**2.2.4 Use Case Narrative**

|  |  |  |
| --- | --- | --- |
| **Tên Use Case** | **Đăng ký & đăng nhập** | |
| **Mô tả** | Người dùng phải bắt buộc đăng ký và đăng nhập để sử dụng app | |
| **Người sử dụng** | Người dùng | |
| **Điều kiện**  **tiên quyết** | Người dùng mới, lần đầu tiên tải và sử dụng ứng dụng. | |
| **Chi tiết các chức năng** | **Người dùng thực hiện** | **Hệ thống phản hồi** |
| 1. Người dùng mở ứng dụng lên.  3. Người dùng sẽ bấm nút đăng ký, ứng dụng chuyển sang mục Đăng ký để tạo tài khoản.  4. Người dùng điền đầy đủ thông tin đăng ký, và bấm nút Tạo tài khoản.  6. Người dùng bấm nút Đăng nhập nếu đã có tài khoản, ứng dụng chuyển sang mục Đăng nhập.  7. Người dùng sử dụng id, mật khẩu của mình để điền vào khung và bấm nút Đăng nhập.  9. Ứng dụng nhận thông báo từ server, tự động chuyển người dùng sang mục Bảng điều khiển. | 2. Server yêu cầu người dùng đăng nhập.  5. Server lưu trữ tất cả thông tin đăng ký tài khoản mới do người dùng cung cấp vào CSDL.  8. Server kiểm tra thông tin đăng nhập, nếu đúng với CSDL, server sẽ gửi thông báo đăng nhập thành công tới ứng dụng. |
| **Kết quả**  **mong đợi** | Sau khi đăng ký, người dùng có thể đăng nhập và sử dụng ứng dụng. | |

|  |  |  |
| --- | --- | --- |
| **Tên Use Case** | **Tự động đăng nhập** | |
| **Mô tả** | Người dùng đã có tài khoản, đăng nhập vào và sử dụng ứng dụng. | |
| **Người sử dụng** | Người dùng | |
| **Điều kiện**  **tiên quyết** | Người dùng đã đăng ký tài khoản. | |
| **Chi tiết các chức năng** | **Người dùng thực hiện** | **Hệ thống phản hồi** |
| 1. Người dùng đã đăng nhập từ trước, mở lại ứng dụng.  3. Ứng dụng nhận phản hồi từ server, nếu biến đăng nhập có giá trị “Yes”, ứng dụng tự động chuyển người dùng sang Bảng điều khiển.  4. Nếu biến đăng nhập có giá trị “No”, người dùng phải đăng nhập lại.  6. Ứng dụng nhận thông báo từ server, tự động chuyển người dùng sang mục Bảng điều khiển. | 2. Server kiểm tra biến đăng nhập tương ứng với người dùng, nếu biến có giá trị “Yes”, server sẽ gửi thông báo đến ứng dụng.  5. Server kiểm tra thông tin đăng nhập, nếu đúng với CSDL, server sẽ gửi thông báo đăng nhập thành công tới ứng dụng.  7. Server tự động set biến đăng nhập với giá trị “Yes” để kiểm tra cho lần đăng nhập sau. |
| **Kết quả**  **mong đợi** | Sau khi đăng nhập, người dùng sẽ bắt đầu sử dụng ứng dụng. | |

|  |  |  |
| --- | --- | --- |
| **Tên Use Case** | **Quên mật khẩu** | |
| **Mô tả** | Người dùng quên mật khẩu đăng nhập | |
| **Người sử dụng** | Người dùng | |
| **Điều kiện**  **tiên quyết** | Người dùng đã đăng ký tài khoản. | |
| **Chi tiết các chức năng** | **Người dùng thực hiện** | **Hệ thống phản hồi** |
| 1. Người dùng bấm nút Quên mật khẩu.  2. Ứng dụng chuyển sang mục “Quên mật khẩu” để phục hồi mật khẩu.  4. Sau khi nhập số điện thoại, ứng dụng bắt buộc người dùng nhắn tin theo đúng cú pháp đến tổng đài để lấy mã xác nhận.  6. Nếu đúng mã nhận, người dùng tự động chuyển sang mục nhập lại mật khẩu mới.  7. Sau khi phục hồi mật khẩu, người dùng tiến hành bước đăng nhập lại | 3. Server yêu cầu người dùng nhập số điện thoại khi đăng ký tài khoản.  5. Server nhận tin nhắn từ người dùng và gửi đến tổng đài SMS.  8. Server cập nhật lại mật khẩu mới cho người dùng vào CSDL. |
| **Kết quả**  **mong đợi** | Sau khi phục hồi mật khẩu thành công, người dùng sẽ đăng nhập lại. | |

|  |  |  |
| --- | --- | --- |
| **Tên Use Case** | **Chỉnh sửa thông tin cá nhân** | |
| **Mô tả** | Người dùng thay đổi thông tin cá nhân | |
| **Người sử dụng** | Người dùng | |
| **Điều kiện**  **tiên quyết** | Người dùng đã đăng nhập thành công và đang ở mục Bảng điều khiển. | |
| **Chi tiết các chức năng** | **Người dùng thực hiện** | **Hệ thống phản hồi** |
| 1. Người dùng chuyển sang mục thông tin cá nhân  2. Người dùng được tùy ý thay đổi tất cả thông tin cá nhân của mình.  4. Người dùng sẽ trở về Bảng điều khiển với những tùy chỉnh mới. | 3. Server tiến hành cập nhật lại tất cả những thông tin mới mà người dùng thay đổi. |
| **Kết quả**  **mong đợi** | Người dùng sẽ thay đổi hoặc cập nhật một số thông tin cá nhân. | |

|  |  |  |
| --- | --- | --- |
| **Tên Use Case** | **Tìm kiếm bài viết** | |
| **Mô tả** | Người dùng tìm kiếm những bài viết với nội dung cần tìm. | |
| **Người sử dụng** | Người dùng | |
| **Điều kiện**  **tiên quyết** | Người dùng đang ở trong mục Cộng đồng cười | |
| **Chi tiết các chức năng** | **Người dùng thực hiện** | **Hệ thống phản hồi** |
| 1. Người dùng chuyển sang mục Cộng đồng cười từ Bảng điều khiển.  2. Người dùng sẽ nhập nội dung tìm kiếm vào thanh tìm kiếm trong ứng dụng.  Bước 3: Ứng dụng gửi nội dung tìm kiếm lên server  4. Ứng dụng sau đó nhận gói tin từ server trả về và hiển thị tất cả những bài viết có chứa nội dung tìm kiếm của người dùng. | 3. Server kiểm tra tất cả những bài viết có tiêu đề chứa nội dung tìm kiếm mà người dùng yêu cầu và trả về dưới dạng gói tin. |
| **Kết quả**  **mong đợi** | Người dùng sẽ truy cập vào những bài viết có tiêu đề chứa với nội dung cần tìm kiếm. | |

|  |  |  |
| --- | --- | --- |
| **Tên Use Case** | **Tạo bò** | |
| **Mô tả** | Người dùng tạo bò mới | |
| **Người sử dụng** | Người dùng | |
| **Điều kiện**  **tiên quyết** | Người dùng đang ở trong mục Quản lý bò | |
| **Chi tiết các chức năng** | 1. Người dùng chuyển sang mục tạo bò  2. Người dùng nhập đầy đủ các thông tin cần thiết của 1 con bò và bấm nút tạo bò  4. Người dùng sẽ được chuyển sang bảng danh sách bò với con bò vừa được tạo | 3. Server lưu thông tin của con bò người dùng vừa nhập vào CSDL. |
| **Kết quả**  **mong đợi** | Người dùng tạo bò thành công | |

|  |  |  |
| --- | --- | --- |
| **Tên Use Case** | **Tìm kiếm bò** | |
| **Mô tả** | Người dùng tìm kiếm bò | |
| **Người sử dụng** | Người dùng | |
| **Điều kiện**  **tiên quyết** | Người dùng đang ở trong mục quản lý bò | |
| **Chi tiết các chức năng** | **Người dùng thực hiện** | **Hệ thống phản hồi** |
| 1. Người dùng chuyển sang mục tìm kiếm bò  2. Ứng dụng sẽ cho người dùng lựa chọn 3 cách để tìm kiếm bò là tìm kiếm bò theo ID, tìm kiếm bò theo QR code, tìm kiếm bò theo NFC tag.  3. Người dùng chọn 3 cách trên để tìm kiếm bò |  |
| **Kết quả**  **mong đợi** | Người dùng chọn 1 trong 3 cách trên để tìm kiếm bò | |

|  |  |  |
| --- | --- | --- |
| **Tên Use Case** | **Tìm kiếm bò theo ID** | |
| **Mô tả** | Người dùng tìm kiếm thông tin bò theo ID | |
| **Người sử dụng** | Người dùng | |
| **Điều kiện**  **tiên quyết** | Người dùng đang ở trong mục tìm kiếm | |
| **Chi tiết các chức năng** | **Người dùng thực hiện** | **Hệ thống phản hồi** |
| 1. Người dùng chuyển sang mục tìm kiếm bò theo ID  2. Người dùng nhập ID của bò và bấm tìm kiếm  4. Ứng dụng nhận phản hồi từ server, chuyển người dùng tới mục thông tin của bò | 3. Hệ thống sẽ tìm kiếm theo ID của bò và ID của người dùng để trả về chính xác con bò của người dùng yêu cầu và trả về gói tin bao gồm thông tin của con bò đó. |
| **Kết quả**  **mong đợi** | Người dùng truy cập vào con bò có id đúng với con bò cần tìm kiếm | |

|  |  |  |
| --- | --- | --- |
| **Tên Use Case** | **Tìm kiếm bò theo QRcode** | |
| **Mô tả** | Người dùng tìm kiếm con bò theo QR code trên thân bò | |
| **Người sử dụng** | Người dùng | |
| **Điều kiện**  **tiên quyết** | Người dùng đang ở trong mục Tìm kiếm | |
| **Chi tiết các chức năng** | **Người dùng thực hiện** | **Hệ thống phản hồi** |
| 1. Người dùng chuyển sang mục tìm kiếm bò theo QRcode  2. Người dùng sẽ được hiển thị màn hình scan QRcode với camera.  3. Người dùng scan camera vào tấm hình QR code.  5. Ứng dụng nhận phản hồi từ server, chuyển người dùng tới mục thông tin của bò | 4. Hệ thống sẽ tự động detect tấm hình QR code và trả về gói tin bao gồm thông tin chính xác con bò có mã QR code đó. |
| **Kết quả**  **mong đợi** | Người dùng truy cập vào con bò có QRcode đúng với con bò cần tìm kiếm | |

|  |  |  |
| --- | --- | --- |
| **Tên Use Case** | **Tìm kiếm bò theo NFC tag** | |
| **Mô tả** | Người dùng tìm kiếm bò theo thẻ NFC tag | |
| **Người sử dụng** | Người dùng | |
| **Điều kiện**  **tiên quyết** | Người dùng đang ở trong mục Tìm kiếm | |
| **Chi tiết các chức năng** | **Người dùng thực hiện** | **Hệ thống phản hồi** |
| 1. Người dùng chuyển sang mục tìm kiếm bò theo QRcode  2. Người dùng sẽ được hiển thị màn hình đọc thẻ NFC tag  3. Người dùng dùng tấm thẻ NFC tag chạm vào điện thoại.  5. Ứng dụng nhận phản hồi từ server, chuyển người dùng tới mục thông tin của bò | 4. Server tìm ID của con bò với số ID nhận được trên thẻ NFC và trả về gói tin đúng với con bò cần tìm kiếm. |
| **Kết quả**  **mong đợi** | Người dùng truy cập vào con bò có NFC tag đúng với con bò cần tìm kiếm | |

|  |  |  |
| --- | --- | --- |
| **Tên Use Case** | **Xem thư viện bò** | |
| **Mô tả** | Người dùng xem các thông tin hữu ích về nuôi bò | |
| **Người sử dụng** | Người dùng | |
| **Điều kiện**  **tiên quyết** | Người dùng đang ở bảng điều khiển | |
| **Chi tiết các chức năng** | **Người dùng thực hiện** | **Hệ thống phản hồi** |
| 1. Người dùng nút xem thư viện bò để vào xem danh sách về các loại giống bò, cách nuôi và chăm sóc bò  2. Ứng dụng sẽ gửi request vào thư viện bò để server xử lý. | 3. Server trả về tất cả các bài viết trên thư viện do admin tạo. |
| **Kết quả**  **mong đợi** | Người dùng xem thư viện nuôi bò thành công | |

|  |  |  |
| --- | --- | --- |
| **Tên Use Case** | **Chợ bò** | |
| **Mô tả** | Người dùng vào khu vực chợ bò | |
| **Người sử dụng** | Người dùng | |
| **Điều kiện**  **tiên quyết** | Người dùng đang ở bảng điều khiển | |
| **Chi tiết các chức năng** | **Người dùng thực hiện** | **Hệ thống phản hồi** |
| 1. Người dùng nhấp nút chợ bò trên bảng điều khiển để vào chợ bò.  2. Ứng dụng cho người dùng chọn mục đăng bán bò hoặc là xem thông tin những con bò được bán trong chợ  3. Người dùng chọn mục mà mìn muốn sử dụng. |  |
| **Kết quả**  **mong đợi** | Người dùng chọn đăng bán bò hoặc xem thông tin bò bán để mua | |

|  |  |  |
| --- | --- | --- |
| **Tên Use Case** | **Đăng bán bò** | |
| **Mô tả** | Người dùng đăng bán bò trong danh sách bò của mình | |
| **Người sử dụng** | Người dùng | |
| **Điều kiện**  **tiên quyết** | Người dùng ở trong mục Chợ bò | |
| **Chi tiết các chức năng** | **Người dùng thực hiện** | **Hệ thống phản hồi** |
| 1. Người dùng nhấp nút chợ bò để đăng bán bò  2. Ứng dụng hiển thị cho người dùng danh sách bò hiện có của người dùng.  3. Người dùng chọn 1 con bò muốn bán trong danh sách bò  5. Người dùng nhập tiêu đề, giá của bò và chi tiết liên hệ và bấm nút đăng bò  6. Ứng dụng hiển thị thông báo đăng bán bò thành công | 4. Server trả về thông tin cá nhân của bò mà người dùng chọn.  5. Server sẽ lưu trữ thông tin người dùng vừa nhập vào cơ sở dữ liệu và truyền vào danh sách bò đang bán. |
| **Kết quả**  **mong đợi** | Người dùng đăng bán bò thành công. | |

|  |  |  |
| --- | --- | --- |
| **Tên Use Case** | **Xem danh sách bò đang bán** | |
| **Mô tả** | Người dùng xem tất cả những con bò đang bán trên hệ thống | |
| **Người sử dụng** | Người dùng | |
| **Điều kiện**  **tiên quyết** | Người dùng đang ở mục chợ bò và nhấn vào nút xem danh sách bò đang bán | |
| **Chi tiết các chức năng** | **Người dùng thực hiện** | **Hệ thống phản hồi** |
| 1. Người dùng nhấp nút xem danh sách bò đang bán.  2. Ứng dụng sẽ gửi yêu cầu xem danh sách bò đang bán.  4. Người dùng sẽ được hiển thị tất cả bò trong danh sách bò đăng bán.  5. Người dùng chọn vào con bò muốn mua để xem thông tin liên hệ và liên hệ người đăng nếu muốn giao dịch. | 3. Server nhận request và trả về cho người dùng danh sách những con bò đang được đăng bán trên hệ thống. |
| **Kết quả**  **mong đợi** | Hiển thị cho người dùng danh sách những con bò đang bán | |

|  |  |  |
| --- | --- | --- |
| **Tên Use Case** | **Thống kê user, số lượng bò** | |
| **Mô tả** | Admin xem bảng danh sách thống kê số lượng user và bò | |
| **Người sử dụng** | Admin | |
| **Điều kiện**  **tiên quyết** | Admin đã đăng nhập vào trang web quản trị viên | |
| **Chi tiết các chức năng** | **Người dùng thực hiện** | **Hệ thống phản hồi** |
| 1. Admin chọn vào mục thống kê để xem thống kê user và bò hiện tại của hệ thống  3. Admin xem được bảng thống kê user và bò | 2. Server trả về cho admin số lượng user và số lượng bò của hệ thống |
| **Kết quả**  **mong đợi** | Hiển thị bảng thống kê số lượng user và số lượng bò | |

|  |  |  |
| --- | --- | --- |
| **Tên Use Case** | **Ban user** | |
| **Mô tả** | Admin kiểm tra những tài khoản không hợp lệ hoặc lừa đảo để chặn truy cập | |
| **Người sử dụng** | Admin | |
| **Điều kiện**  **tiên quyết** | Admin đã đăng nhập vào trang web quản trị viên | |
| **Chi tiết các chức năng** | **Người dùng thực hiện** | **Hệ thống phản hồi** |
| 1. Admin vào trang quản trị để xem thông tin user.  2. Admin có thể chặn user truy cập nếu thấy user có dấu hiệu lừa đảo, không hợp lệ. | 4. Server xóa thông tin user đó ra khỏi hệ thống |
| **Kết quả**  **mong đợi** | Admin sẽ thực hiện thao tác ban user | |

|  |  |  |
| --- | --- | --- |
| **Tên Use Case** | **Kiểm duyệt chợ bò** | |
| **Mô tả** | Admin kiểm duyệt những con bò được đăng bán trong chợ bò | |
| **Người sử dụng** | Admin | |
| **Điều kiện**  **tiên quyết** | Admin đang xem phần xem danh sách bò bán | |
| **Chi tiết các chức năng** | **Người dùng thực hiện** | **Hệ thống phản hồi** |
| 1. Admin kiểm tra con bò đang được đăng bán sai thông tin  2. Admin thực hiện thao tác xóa bò đang được bán nếu có sai phạm  4. Ứng dụng tự động cập nhật lại nội dung mới. | 3. Server nhận yêu cầu từ admin, xóa CSDL con bò đang ở trong chợ bò. |
| **Kết quả**  **mong đợi** | Con bò được đăng bán bị xóa khỏi hệ thống | |

**CHƯƠNG III: CÔNG NGHỆ**

* 1. **Giới thiệu về NodeJS**

Node.js là 1 nền tảng chạy trên môi trường V8 Javascript runtime. Node.js cho phép lập trình viên xây dựng các ứng dụng có tính mở rộng cao sử dụng Javascript trên server. Và vì được porting từ C nên về mặt tốc độ xử lý thì rất nhanh. Các ứng dụng (thật sự mạnh khi được xây dựng với Node.js) bao gồm: xây dựng websocket server (Chat server), ứng dụng upload file rất nhanh trên client, các máy chủ quảng cáo hoặc bất kỳ ứng dụng dữ liệu thời gian thực nào.

**3.1.1 Ưu và nhược điểm**

|  |  |
| --- | --- |
| **Ưu điểm** | **Nhược điểm** |
| Bởi lẽ REST/JSON APIs gọn nhẹ là điều khiến NodeJS tỏa sáng. Với cơ chế event-driven, non-blocking I/O (Input/Output) và mô hình kết hợp với Javascript là sự lựa chọn tuyệt vời cho các dịch vụ Webs làm bằng JSON.  Nếu ta định viết 1 ứng dụng thể hiện trên 1 trang (Gmail?) NodeJS rất phù hợp để làm. Với khả năng xử lý nhiều Request/s đồng thời thời gian phản hồi nhanh. Các ứng dụng ta định viết không muốn nó tải lại trang, gồm rất nhiều request từ người dùng cần sự hoạt động nhanh để thể hiện sự chuyên nghiệp thì NodeJS sẽ là sự lựa chọn của ta.  NodeJS sẽ tận dụng tối đa Unix để hoạt động. Tức là NodeJS có thể xử lý hàng nghìn Process và trả ra 1 luồng khiến cho hiệu xuất hoạt động đạt mức tối đa nhất và tuyệt vời nhất. Các web thông thường gửi HTTP request và nhận phản hồi lại (Luồng dữ liệu). Giả xử sẽ cần xử lý 1 luồng giữ liệu cực lớn, NodeJS sẽ xây dựng các Proxy phân vùng các luồng dữ liệu để đảm bảo tối đa hoạt động cho các luồng dữ liệu khác.  Giả sử ta xây dựng 1 ứng dụng chat, feed … Facebook, Twitter là điển hình cho Web thực. NodeJS làm khá tốt điều đó! | Nếu ta cần xử lý các ứng dụng tốn tài nguyên CPU như encoding video, convert file, decoding encryption… hoặc các ứng dụng tương tự như vậy thì không nên dùng NodeJS (Lý do: NodeJS được viết bằng C++ & Javascript, nên phải thông qua thêm 1 trình biên dịch của NodeJS sẽ lâu hơn 1 chút). Trường hợp này ta hãy viết 1 Addon C++ để tích hợp với NodeJS để tăng hiệu suất tối đa !(Việc tích hợp rất thân thiện và nhanh chóng).  NodeJS, PHP, Ruby, Python .NET …thì việc cuối cùng là phát triển các App Web. NodeJS mới sơ khai như các ngôn ngữ lập trình khác. Vậy nên ta đừng hi vọng NodeJS sẽ hơn PHP,Ruby,Python… ở thời điểm này. Nhưng với NodeJS ta có thể có 1 ứng dụng như mong đợi, điều đó là chắc chắn (perfect).  Với những gì các ngôn ngữ tiền bối đang có(cộng đồng lâu năm, framework, cms, opensource…) Nếu ta chưa biết về NodeJS thì việc cần xây dựng dự án quan trọng, kinh doanh phát triển trên NodeJS sẽ không phải lựa chọn bây giờ.  NoSQL + Nodejs không dành cho người mới bắt đầu lập trình. |

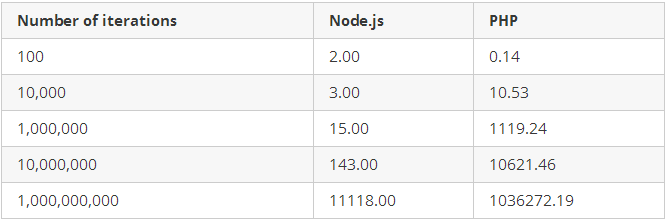
**3.1.2 Tại sao nên dùng NodeJS**

Tất cả các web developer đều biết JavaScript, do đó, việc cả phía client và phía server đều sử dụng JavaScript cũng mang nhiều ý nghĩa. Hơn nữa, việc cả hai phía Client/Server cùng sử dụng một ngôn ngữ chung đem tới khả năng hấp dẫn là cả hai phía có thể sử dụng chung một đoạn code, một module, làm giảm thời gian phát triển, ít lỗi hơn, và thời gian test, fix bug cũng ngắn hơn.

Đặc biệt trong ứng dụng Cộng đồng cười, mục tiêu của ứng dụng đưa ra là xây dựng 1 mạng xã hội ảo, với rất nhiều hoạt động tương tác liên tục, qua lại lẫn nhau của người dùng, dẫn đến việc server phải xử lý hàng ngàn request liên tục, dẫn tới việc server có thể bị quá tải nếu viết bằng ngôn ngữ lập trình server thông thường như PHP hay Python, ta có thể thấy được điều này qua bảng so sánh:

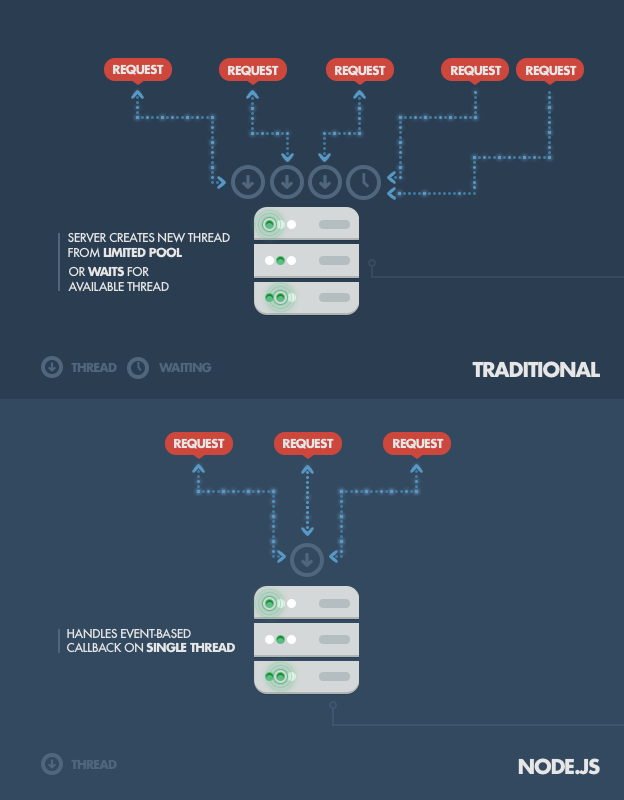
|  |  |
| --- | --- |
| **NodeJS** | **PHP** |
| **var** i, a, b, c, max;  max = 1000000000;  **var** d = Date.now();  **for** (i = 0; i < max; i++)  {      a = 1234 + 5678 + i;      b = 1234 \* 5678 + i;      c = 1234 / 2 + i;  }  console.log(Date.now() - d); | $a = null;  $b = null;  $c = null;  $i = null;  $max = 1000000000;  $start = microtime(true);  for ($i = 0; $i < $max; $i++) {      $a = 1234 + 5678 + $i;      $b = 1234 \* 5678 + $i;      $c = 1234 / 2 + $i;  }  var\_dump(microtime(true) - $start); |

Cả 2 đoạn code đều cùng thực thi 1 công việc, nhưng tốc độ thực thi thì khác nhau, ta có thể thấy được điều này qua bảng kết quả:



Kết quả trung bình cho thấy 10 lần thử nghiệm. PHP đáng chú ý là chạy nhanh hơn với số lượng đầu vào lặp lại là nhỏ, nhưng ưu thế này bị mất đi nhanh chóng nếu ta tăng số lượng xử lý đầu vào. Gần như là PHP chậm hơn 93% so với Node.js ! Điều này có được là do Node.js sử dụng kiến trúc mô-đun để đơn giản hóa việc tạo ra các ứng dụng phức tạp, cho nên tốc độ xử lý là rất nhanh.

Ngoài ra, NodeJS còn hoạt động theo cơ chế non-blocking, event driven ( bất đồng bộ, song song ) để tạo ra các ứng dụng web thời gian thực nhẹ tải, hiệu suất cao đồng thời có thể chạy trên rất nhiều thiết bị khác nhau, phù hợp với những yêu cầu mà ứng dụng Cộng đồng cười cần có.



Các nền tảng web-server cũ (apache,nginx...) sẽ sinh ra một thread mới khi có request và cấp bộ nhớ cho thread đó đến khi nào hết RAM thì thôi, nghĩa là max-connection sẽ phụ thuộc khá nhiều vào RAM. Nodejs thì ngược lại, dùng một thread duy nhất cùng với các câu lệnh I/O non-blocking cho phép nó phục vụ hàng chục ngàn connection cùng một lúc.

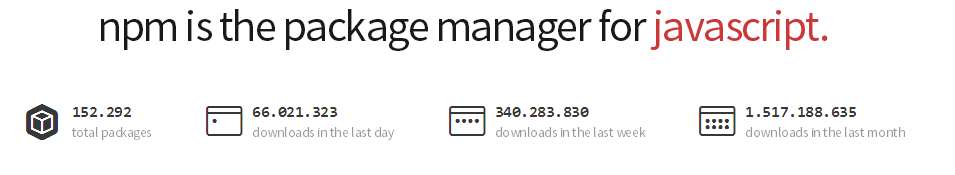
Giả sử mỗi request được cấp 2MB với một cục RAM 8GB ta chỉ có thể phục vụ 4000 connection một lúc . Đó là còn chưa tính bộ nhớ cho các process khác hay phục vụ cho việc chuyển đổi của đống thread đó. Tuy  nhiên cũng với ngần ấy tài nguyên Nodejs có thể đảm đương một lúc cả triệu người dùng sử dụng ứng dụng.

Và một điều cuối cùng là NodeJS hoàn toàn viết bằng Javacripts, đó là 1 điều thuận lợi khi ta xây dựng nên ứng dụng này, thật tuyệt vời khi mà cả server lẫn client đều sử dụng cùng 1 ngôn ngữ lập trình, giúp tiết kiệm rất nhiều thời gian và công sức thay vì phải lập trình 1 lúc 2 hoặc 3 ngôn ngữ khác nhau.

**3.1.3 NPM**

NPM là viết tắt của từ Node Package Modules, là 1 package manager của Node. Ta có thể sử dụng nó để cài cắm các module package khác của tất cả mọi lập trình viên trên thế giới mà đã được publish trên **NPM**. Hoặc ta cũng có thể publish những module package mà chính ta viết cho cộng đồng sử dụng.

NPM có nhiệm vụ là quản lý và sử dụng các module được tải về và sử dụng trong dự án của ta dưới dạng gói thông qua command hoặc các file nén từ github. Khi ta tham gia vào 1 dự án nào đó, thì những module này sẽ không được đẩy lên github, nếu ta muốn dự án được chạy tốt trên localhost thì sau khi ta clone mã nguồn từ github về, ta phải tiến hành cái đặt các gói npm tương ứng.



Tính đến thời điểm hiện tại, đã có hơn 15.000 package được viết, trung bình 1 ngày có hơn 60 triệu lượt tải các package, và con số đó trong 1 tháng là hơn 1.5 tỷ lượt tải ! Điều này chứng tỏ nền tảng web với nodejs hứa hẹn 1 vùng đất màu mỡ cho các lập trình viên phát triển ứng dụng của mình trong tương lai gần.

Với hàng ngàn module như thế, việc xây dựng ứng dụng của ta rõ ràng là sẽ dễ dàng hơn là thay vì phải code từ đầu đến cuối. Đồng thời, qua việc sử dụng các module này, ta cũng sẽ tiếp cận và học hỏi được nhiều hơn nữa về NodeJS nhờ vào sự hỗ trợ của cộng đồng, từ đó sẽ phát triển và nâng cấp ứng dụng ngày một tốt hơn.

Một số module cần thiết khi khởi tạo và được sử dụng trong ứng dụng Cộng đồng cười:

* express : chính là nodejs nhưng được viết gộp lại và giúp lập trình viên phát triển dễ dàng: đây được là là web framework của nodejs
* mongoose : quản lý mongdb của ta, giống như Doctrine trong PHP, giúp ta tiếp cận và xử lý dữ liệu với mongodb
* jade : 1 dạng template mới dùng trong nodejs view, nó giúp cho phần viên ngắn gọn và dễ nhìn hơn so với html
* connect-mongodb : giúp kết nối với mongodb
* socket.io : dùng để quản lý socket, chuyên dùng tạo những ứng dụng chat thời gian thực.
* request : quản lý yêu cầu và dữ liệu được truyền giữa client và server nodejs
* passport-facebook : kết nối và chứng thực với facebook.
  1. **Giới thiệu về NoSQL**

Khái niệm NoSQL được tạo ra vào năm 1998 bởi Carlo Strozzi. Nhiều người nghĩ rằng khái niệm này dùng để hạ thấp SQL nhưng thực ra nó có nghĩa là Not Only SQL. Về mặt ý tưởng thì cả hai (NoSQL và RDMS) sẽ sống chung và bổ sung cho nhau.

NoSQL là 1 dạng CSDL mã nguồn mở không sử dụng T-SQL để truy vấn thông tin. NoSQL viết tắt bởi: None-Relational SQL, hay có nơi thường gọi là Not-Only SQL.

NoSQL được phát triển trên Javascript Framework với kiểu dữ liệu là JSON và dạng dữ liệu theo kiểu key và value (1 đặc trưng về dữ liệu trong JSON). NoSQL ra đời như là 1 mảnh vá cho những khuyết điểm và thiếu sót cũng như hạn chế của mô hình dữ liệu quan hệ RDBMS về tốc độ, tính năng, khả năng mở rộng, memory cache,…

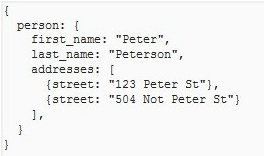
Với NoSQL ta có thể mở rộng dữ liệu mà không lo tới những việc như tạo khóa ngoại, khóa chính, kiểm tra ràng buộc, .v.v…Vì NoSQL không hạn chế việc mở rộng dữ liệu nên tồn tại nhiều nhược điểm như: sự phục thuộc của từng bản ghi, tính nhất quán, toàn vẹn dữ liệu,….nhưng chúng ta có thể chấp nhận những nhược điểm đó để khiến ứng dụng cải thiện hiệu suất cao hơn khi giải quyết những bài toán lớn về hệ thống thông tin, phân tán hay lưu trữ dữ liệu.

NoSQL được sử dụng ở đâu? NoSQL được sử dụng ở rất nhiều công ty, tập đoàn lớn, ví dụ như FaceBook sử dụng Cassandra do FaceBook phát triển, Google phát triển và sử dụng BigTable…

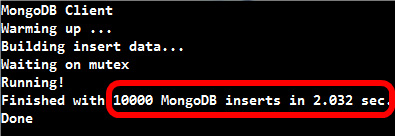
Trong ứng dụng này, nhóm em đã quyết định sử dụng MongoDB có cấu trúc NoSQL vì những ưu điểm và khả năng lưu trữ, mở rộng, truy vấn tuyệt vời của nó.

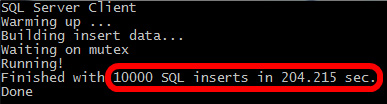
**3.2.1 Ưu điểm của MongoDB**

MongoDB là 1 hệ thống CSDL mã nguồn mở được phát triển và hỗ trợ bởi 10gen, viết bằng ngôn ngữ C++ là CSDL NoSQL hàng đầu được hàng triệu người sử dụng. MongoDB tự nhận mình là nền tảng NoSQL database hàng đầu dựa trên các số liệu thống kê. Và không thể phủ định độ phổ biến của nó.MongoDB lưu dữ liệu dạng Json mỗi bản ghi được coi như 1 document, chính vì vậy mà nó rất dễ tiếp cận.



Rõ ràng, trong bất kì trang web hay ứng dụng nào, thì việc tổ chức và quản lý CSDL bao giờ cũng là 1 vấn đề nhức nhối. Trong ứng dụng này, ta sẽ sử dụng MongoDB như là một giải pháp tối ưu cho vấn đề CSDL. Với mỗi người dùng sử dụng ứng dụng, trung bình có hơn 10 trường giá trị khác nhau, ví dụ như họ tên, số điện thoại, email ..v..v..và lấy con số đó nhân lên cho hàng ngàn người dùng, thì đó thực sự là con số khổng lồ, và nếu sử dụng CSDL SQL để thao tác hoặc truy vấn dữ liệu thì quả thực đó là 1 cơn ác mộng, nhưng với MongoDB, mọi chuyện sẽ được giải quyết.





Ảnh trên cho thấy tốc độ thêm CSDL vượt trội của MongoDB so với SQL.

Chưa dừng lại ở đó, MongoDB rất dễ học, dễ nắm bắt nhanh, ko có khái niệm khóa chính, khóa ngoại. Ta cũng không cần phải định nghĩa cấu trúc dữ liệu trước khi lưu trữ, khả năng mở rộng gần như là vô hạn, do đó trong ứng dụng mọi người dùng có thể thêm bạn thoải mái hay chat liên tục, mọi thứ đều được lưu trữ lại. Và ưu điểm cuối cùng khi sử dụng MongoDB là tốc độ truy vấn dữ liệu rất nhanh và chính xác, ta có thể thấy được điều đó qua biểu đồ sau:



Nhìn vào biểu đồ trên, rõ ràng MongoDB vẫn tỏ ra vượt trội hơn hẳn so với SQL về hiệu suất (khoảng 3 lần).

Với những biểu đồ so sánh trên, ta hoàn toàn có lý do khi sử dụng MongoDB để lưu trữ CSDL của ứng dụng. Bởi lẽ khi ứng dụng hoạt động, mỗi giây sẽ có hàng ngàn lượt truy vấn dữ liệu liên tục giữa người dùng và server, cùng với đó là việc ghi dữ liệu liên tục từ các hoạt động ví du như chat, kết bạn, đăng bài..v.v..của người dùng, thì việc sử dụng MongoDB là phương án tốt nhất mà ta nên chọn. Đơn giản, dễ học, miễn phí, tốc độ đọc, ghi và truy vấn dữ liệu nhanh nhất hiện nay, mọi thứ đều nằm trong MongoDB.

**3.2.2 Cách thức lưu trữ dữ liệu của MongoDB**

MongoDB được rút ra từ “humongous”, có nghĩa là rất lớn. Nó được phát triển từ C++ bởi công ty 10gen.

Về cơ bản, mọi thứ trong MongoDb được lưu trữ với dạng document, cụ thể là json-style document (BSON). Và như vậy là nó sẽ không có schema, không có lưu trữ dữ liệu dạng bảng như cơ sở dữ liệu quan hệ, và cũng không có joins.

Dữ liệu được lưu trữ sẽ trông giống như thế này:

*{*

*\_id: 0,*

*title: ”,*

*body: ”,*

*comments: [{*

*person: ”,*

*comment: ”,*

*created\_at: new Date()*

*}],*

*created\_at: new Date()*

*}*

Câu hỏi khiến nhiều người thắc mắc là làm thế nào để truy vấn NoSQL nếu như nó không hỗ trợ ngôn ngữ SQL. Mặc dù không phải là RDBMS nhưng MongoDB vẫn cung cấp các cơ chế để truy vấn dữ liệu. Cụ thể như sau:

*db.collection.find(<query>,)*

Phương thức find() tương đương với câu SELECT trong SQL, và tương đương với mệnh đề WHERE, còn tương ứng với danh sách các fields ta cần truy vấn.

Ví dụ ta có một collection chứa các bộ phim, mỗi phần tử gồm có title và rating của khán giả.

*> db.movies.insert({title:"SHERLOCK BBC", rating:9.2})*

*> db.movies.insert({title:"Elementary TV Series", rating:7.6})*

*> db.movies.insert({title:"Lie to Me (2009-2011)", rating:7.8})*

***/\*SELECT\*FROM movies WHERE title= ‘Elementary TV Series’\*/***

*> db.movies.find({title:"Elementaryeries"})*

*{*

*"\_id" : ObjectId("51556737c79ef506ca8fc47d"),*

*"title" : "Elementary TV Series",*

*"rating" : 7.6*

*}*

***/\*SELECT \* FROM movies WHERE rating>9\*/***

*> db.movies.find({rating:{$gt:9}})*

*{*

*"\_id" : ObjectId("515567f1c79ef506ca8fc47f"),*

*"title" : "SHERLOCK BBC",*

*"rating" : 9.2 }*

*}*

* 1. **VPS**

Server riêng ảo (Virtual Private Server - VPS) là phương pháp phân chia một server vật lý thành nhiều server riêng. Mỗi server là một hệ thống hoàn toàn riêng biệt, chạy hệ điều hành riêng có toàn quyền quản lý và khởi động lại hệ thống.

VPS dành cho các doanh nghiệp vừa và nhỏ cần có server riêng có thể quản trị từ xa, có thể cài đặt phần mềm theo nhu cầu mà không bị hạn chế, có thể xây dựng server web, email, backup dữ liệu hoặc dùng để truyền tải file dữ liệu giữa các chi nhánh một cách dễ dàng, nhanh chóng, bảo mật.

VPS hoạt động hoàn toàn như một server riêng với quyền quản trị cao nhất, đảm bảo tính bảo mật cao, giúp tết kiệm được chi phí đầu tư server ban đầu.

Có thể dùng VPS để thiết lập Web Server, Mail Server cũng như các server ứng dụng khác và có thể cài đặt riêng theo nhu cầu cũng nhữ dễ dàng chia sẽ dữ liệu, truyền dữ liệu giữa các chi nhánh.

Có tính chất quản lý nội bộ riêng của doanh nghiệp.

Không tốn chi phí mua thiết bị, chi phí bảo dưỡng.

Dễ dàng nâng cấp tài nguyên RAM, HDD, Băng thông khi cần.

Có thể cài lại hệ điều hành từ 5-10 phút (cần thiết trong lúc khẩn cấp giảm tối đa thời gian sập Mạng của hệ thống, các trường hợp lỗi, quá tải hay tấn công mạng).

Có thể quản trị từ xa, cài đặt các ứng dụng quản trị doanh nghiệp, CRM, quản lý khách hàng, bán hàng trực tuyến...

**3.3.1 Tổng quan về DigitalOcean**

DigitalOcean là một nhà cung cấp dịch vụ máy chủ riêng ảo (VPS – Virtual Private Server) của Mỹ có trụ sở tại thành phố New York. Hiện nay [DigitalOcean](http://blog-xtraffic.pep.vn/digitalocean-nha-cung-cap-dich-vu-vps-gia-re/) là nhà cung cấp [VPS](http://blog-xtraffic.pep.vn/vultr-nha-cung-cap-vps-gia-re/) giá rất cạnh tranh, [hỗ trợ](http://blog-xtraffic.pep.vn/cong-cu-ho-tro-khach-hang-truc-tuyen-website/) ổ cứng hiệu năng cao SSD và có mức giá thấp nhất từ 5 USD/Tháng đến mức giá 640 USD/Tháng.

DigitalOcean được thành lập vào năm 2011 bởi Ben Uretsky. Hiện nay DigitalOcean có các trung tâm dữ liệu tại New York (USA), San Francisco (USA), Amsterdam (Hà Lan) và Singapore.

DigitalOcean có hỗ trợ ổ cứng SSD, là dạng ổ cứng hiệu năng cao so với loại ổ cứng thông thường. SSD rất tốt khi ta dùng làm nơi lưu trữ dữ liệu đòi hỏi phải truy vấn liên tục, đặc biệt là database. Dù ta dùng SSD trong trường hợp nào đi nữa thì nó cũng là một lựa chọn tốt, tuy nhiên chi phí dành cho nó cũng không phải thấp so với loại ổ cứng thông thường hiện nay.

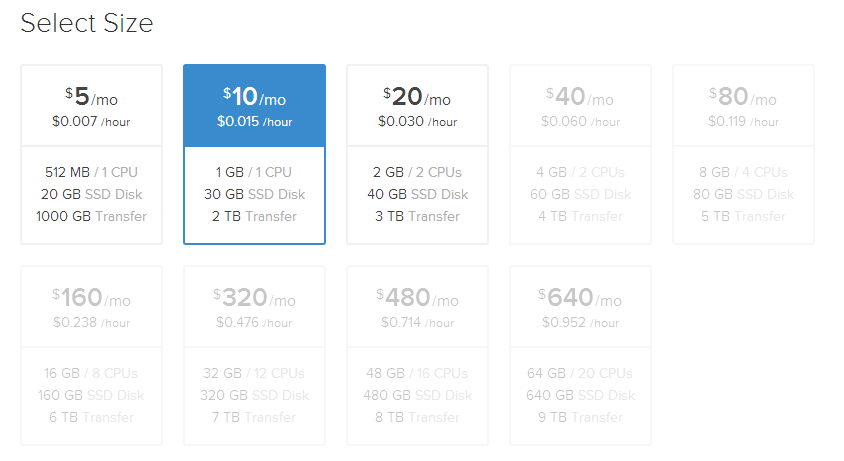
DigitalOcean có mức giá rẻ và linh hoạt, phù hợp cho mọi đối tượng từ sinh viên hay những công ty lớn thì DigitalOcean đều có những gói dịch vụ cung cấp đa dạng. Từ mức phí rẻ nhất 5USD/tháng với cấu hình 512MB RAM, 1CPU, 20GB không gian lưu trữ SSD và 1TB bandwidth đủ để ta chạy hầu hết những ứng dụng không đòi hỏi hiệu suất quá cao như blog, forum,… Nhưng nếu ta là một doanh nghiệp lớn cần một dịch vụ cao cấp thì DigitalOcean vẫn có thể đáp ứng tốt yêu cầu với cấu hình khủng : 64GB RAM , 20 CPU , 640 GB SSD Disk , 9 TB bandwidth.

Thời gian khởi tạo VPS trên DigitalOcean rất nhanh, chỉ cần 55 giây là ta đã có 1 VPS để chạy các dịch vụ của mình.

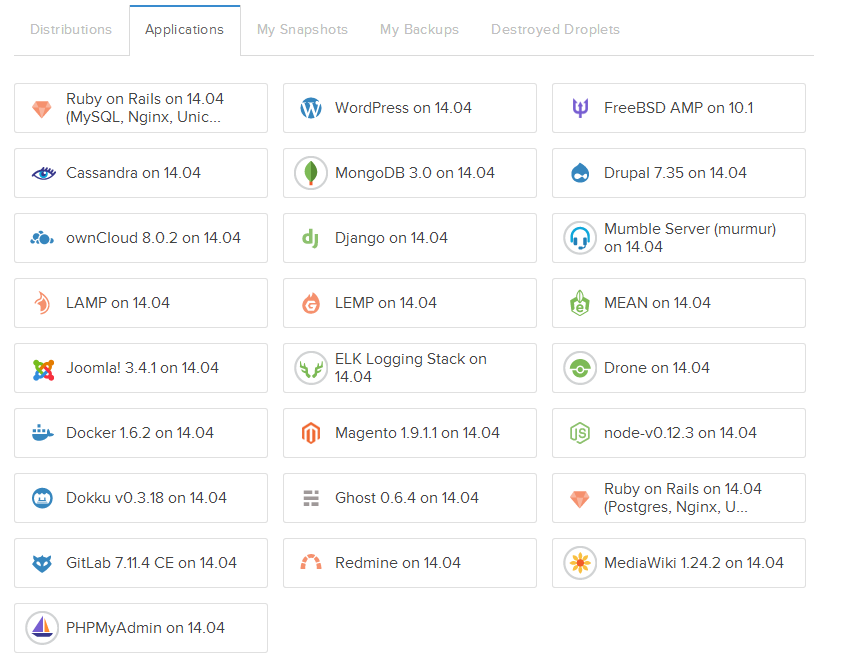
DigitalOcean có nhiều trung tâm dữ liệu trên thế giới, đó là một trong những tính năng rất hay của DigitalOcean đó là ta có thể chọn vùng (thành phố, quốc gia) để khởi tạo VPS của ta. Hiện nay DigitalOcean có các trung tâm dữ liệu tại New York (USA), San Francisco (USA), Amsterdam (Hà Lan) và Singapore giúp ta lựa chọn tối ưu nhất cho dịch vụ của mình.

**3.3.2 Server của ứng dụng sử dụng dịch vụ của DigitalOcean**

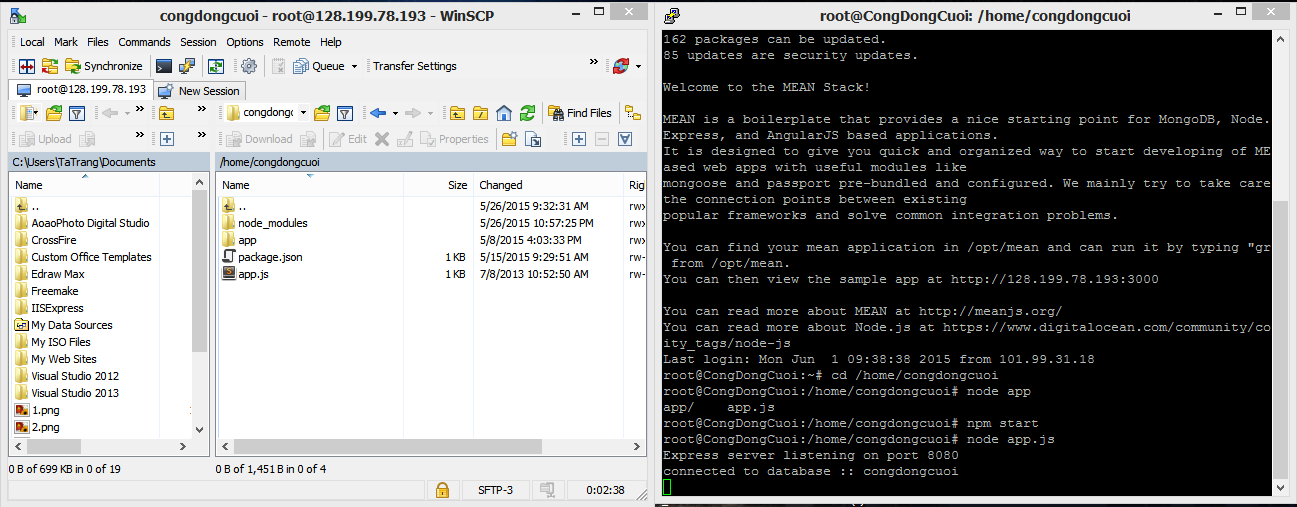
DigitalOcean có khả năng hoạt động rất tốt, do được trang bị ổ cứng SSD cùng với cấu hình có thể tùy chỉnh, mở rộng, hoặc nâng cấp RAM, CPU tùy thuộc vào lượng người dùng sử dụng ứng dụng.



Nhóm sử dụng hệ điều hành Ubuntu nhân Linux được cung cấp miễn phí. Bên cạnh đó DigitalOcean còn hỗ trợ thêm 1 số gói phần mềm kèm theo VPS.

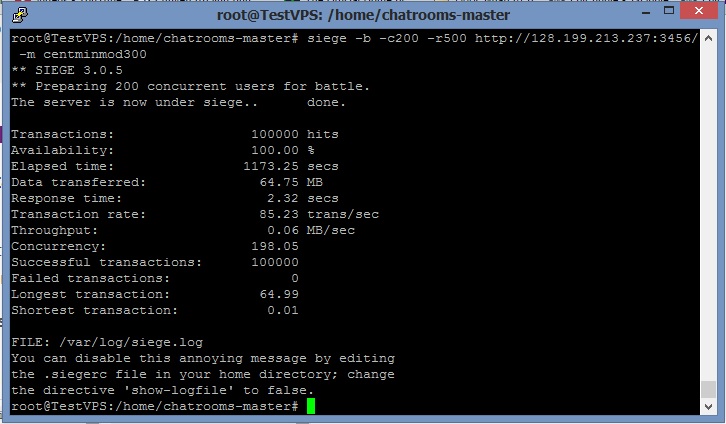


Do DigitalOcean không hỗ trợ giao diện đồ họa nên khi sử dụng nhóm phải sử dụng lệnh Linux để thao tác trên VPS. Rất may, hiện tại thì có một số phần mêm làm rất tốt nhiệm vụ này, ví dụ như Putty và WinSCP, cho phép người dùng có thể làm việc thuận tiện hơn.



Giao diện làm việc của Putty và WinSCP.

Với sức mạnh phần cứng vượt trội, sử dụng hệ điều hành nhân Linux, cũng như được cài đặt một số phần mềm thích hợp, nên khả năng chịu tải của DigitalOcean hoàn toàn ko thua kém bất kì VPS nào. Với 1 server sử dụng RAM 512MB, 1 CPU và 20GB SSD, được cài sẵn NodeJS, server này có thể chịu tải 1 lúc 100 000 thao tác trong vòng gần 20 phút mà ko hề gặp sự cố nào.



* 1. **Lập trình Windows phone với WinJS**

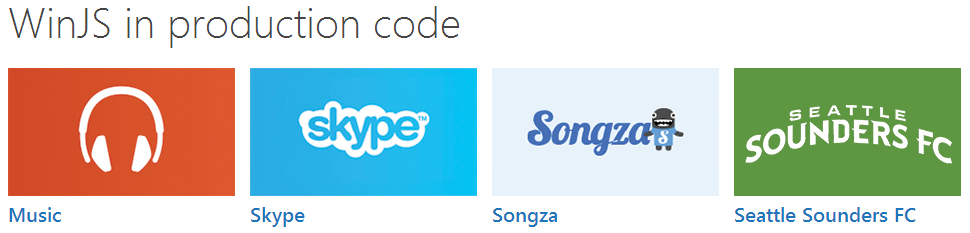
Microsfot đã thông báo thư viện Javascript WinJS của hãng đã được phát hành như là phần mềm mã nguồn mở nhằm giúp trợ giúp các nhà phát triển trong việc phát triển các ứng dụng đa nền tảng theo phong cách Windows.

**3.4.1 WinJS là gì ?**

WinJS là tập hợp các công cụ Javascript trợ giúp các nhà phát triển trong việc xây dựng các ứng dụng HTML/CSS/JS đặc biệt là cho các ứng dụng Windows trên Web, Android và iOS. Và bây giờ nó đã trở thành mã nguồn mở, các nhà phát triển ứng dụng có thể đóng góp trên [Gihub](https://github.com/winjs/winjs). Giống như Angular, Backbone và EmbedJS, WinJS là framework mà các nhà phát triển chọn để sử dụng trên mỗi dự án tùy thuộc liệu nó có đáp ứng nhu cầu của họ hay không. Nó không trường hợp nào mà các nhà phát triển bắt buộc phải sử dụng WinJS để phân phối ứng dụng Window.

Microsoft đã phát hành thư viện này lần đầu tiên vào năm 2011 do vậy các nhà phát triển có thể xây dựng ứng dụng Windows cho cả Windows Phone và giao diện Windows 8 Modern sử dụng Javascript cùng với HTML, CSS và các công cụ phát triển Web khác. Nó là một trong hai cách xây dựng giao diện cho nền tảng Windows 8, cái còn lại là XAML (Extensible Application Markup Language).

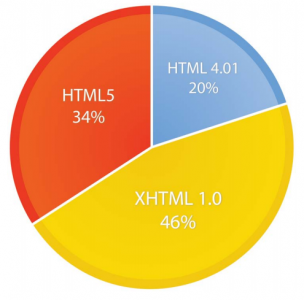
Một số ứng dụng Windows phone nổi tiếng viết bằng WinJS.



**3.4.2 Xây dựng ứng dụng Windows phone với HTML5**

Native App là một ứng dụng được thiết kế đặc biệt chỉ chạy trên một hệ điều hành của một thiết bị nào đó và thường phải điều chỉnh để chạy được trên các thiết bị khác nhau.

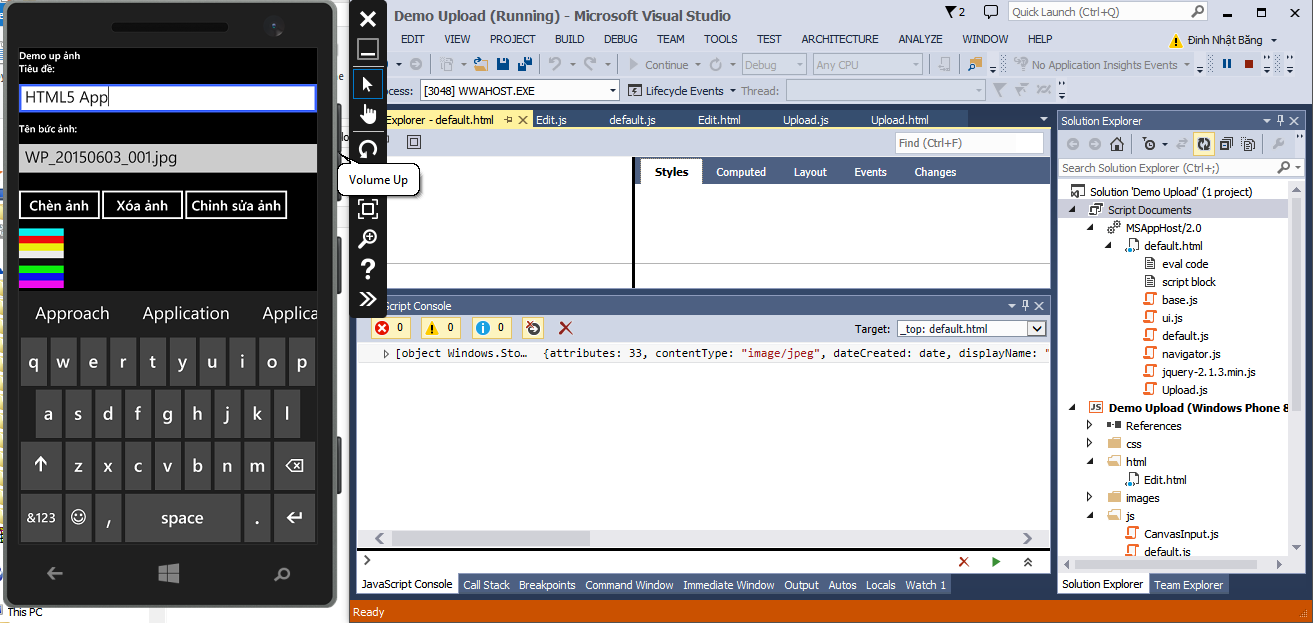
Vì vậy, dường như HTML5 là một lựa chọn tốt hơn so với Native App.



Thị phần các ứng dụng HTML5 hiện nay

HTML5 đặc biệt hữu dụng khi được dùng để phát triển các ứng dụng media và ứng dụng truy cập mạng (cho phép người dùng truy cập nhanh vào một tài khoản hiện có thông qua thiết bị di động, chẳng hạn như tài khoản ngân hàng). Bởi vì các ứng dụng này có thể hiển thị văn bản, hình ảnh, video và quảng cáo với giá rẻ và hiệu quả hơn nhờ HTML5.

HTML5 chính là một công nghệ mang tính đột phá kinh điển. Mặc dù, hiện nay HTML5 còn nhiều mặt hạn chế hơn Native App do thời gian phát triển chưa lâu. Tuy nhiên, công nghệ này đang được cải thiện từng ngày. Bên cạnh đó, kinh phí để sản xuất các ứng dụng HTML5 rẻ hơn các Native Ap. Trong tương lai, tận dụng những ưu điểm của mình, HTML5 sẽ dần thay thế các Native App trong kho ứng dụng của các nhà cung cấp hệ điều hành.



Hình trên là một ứng dụng được xây dựng từ WinJS, HTML5 và CSS.

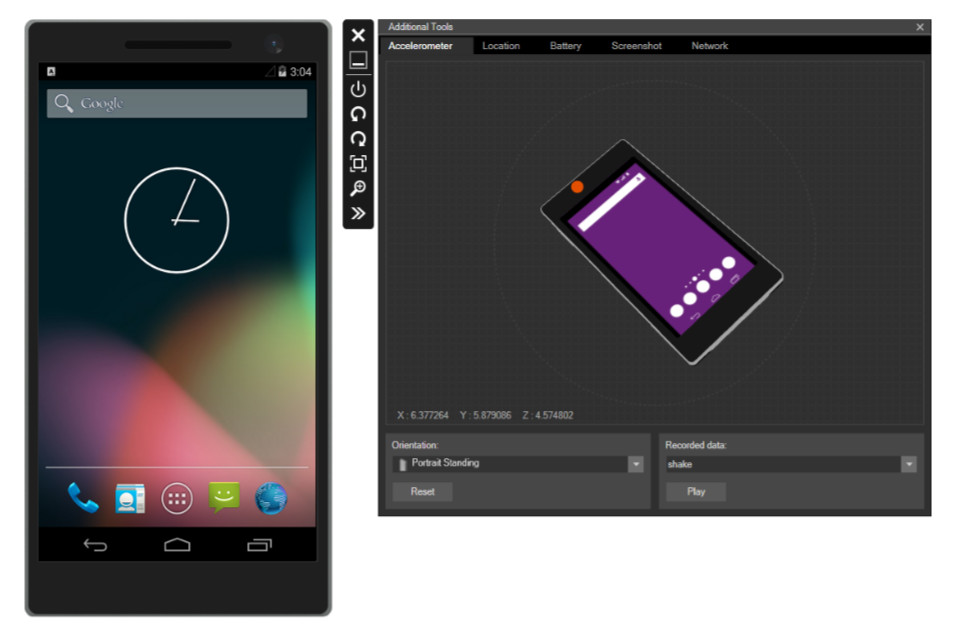
Với việc kết hợp giữa WinJS, HTML5 Và CSS, nhóm em dự định tạo nên 1 “mini browser” chỉ dùng sử dụng riêng biệt cho ứng dụng, với đầy đủ các chức năng của 1 trang web trên di động. Như thế sẽ tận dụng khả năng hiển thị, đồ họa, hiệu ứng mà bộ đôi HTML5/CSS đem lại, tạo nên 1 ứng dụng độc đáo và chỉ duy nhất trên Windows phone.

Mặc dù được xác định là một công nghệ tiên tiến, nhưng việc phổ cập công nghệ mới HTML5 không thể diễn ra trong một sớm một chiều. Bước đầu, HTML5 sẽ mới chỉ hỗ trợ cho các ứng dụng media và truy cập mạng, còn bước tiếp theo nền tảng này sẽ hướng tới ngành công nghiệp sản xuất game di động đầy tiềm năng.

**3.4.3 Phát triển ứng dụng di động đa nền tảng (Cross Platform)**

Với HTML5, ta vẫn nên dùng công nghệ này khi ứng dụng của ta sử dụng nội bộ, có mức độ tính toán logic tin cao, muốn đồng nhất và hạn chế sai sót số liệu, hoặc nếu có giao diện đơn giản tiện dụng, không có nhiều hiệu ứng và dữ liệu phức tạp thì chúng vẫn sẽ rất tốt khi phân phối trên chợ ứng dụng. Ta sẽ tiết kiệm được rất nhiều thời gian và chi phí nếu muốn hỗ trợ nhiều nền tảng. Đây thực sự là một bước tiến lớn so với HTML5.

Song song với sự phát triển của HTML5 là sự phát triển và ra mắt của Visual Studio 2015 cho phép lập trình viên tạo Website trên ASP.NET 5 có thể chạy trên đa nền tảng, bao gồm Windows, Linux và Mac và tích hợp hỗ trợ xây dựng ứng dụng chạy trên đa thiết bị với Visual Studio Tools for Apache Cordova và công cụ Visual C++ mới với thư viện phát triển đa nền tảng.



Máy ảo chạy Android được build từ Visual Studio 2015

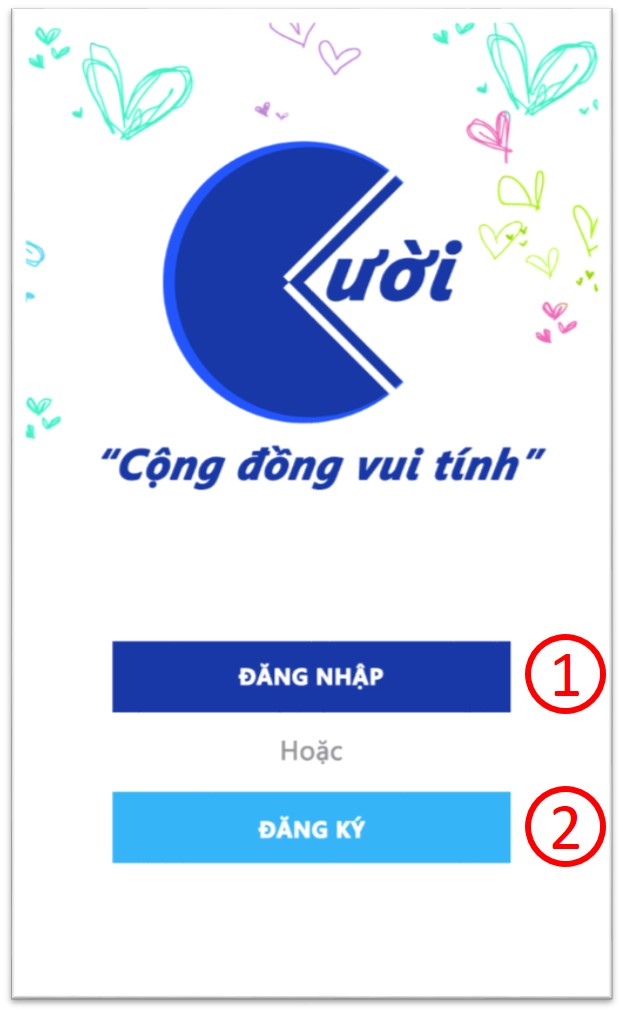
**CHƯƠNG IV: ỨNG DỤNG CỘNG ĐỒNG CƯỜI**

**4.1 Giới thiệu về ứng dụng “Cộng dồng cười”**

Ứng dụng được viết dựa trên ngôn ngữ lập trình Javascript là chủ yếu. Server của ứng dụng sử dụng dịch vụ của Digital Ocean để đặt máy chủ, và dùng NodeJs kết hợp với các gói Npm có sẵn để xây dựng nên. Về phần Client, sử dụng Visual Studio của Microsoft để phát triển. Phần giao diện tương tác với người dùng của ứng dụng ( front-end ) được viết bởi HTML5 kết hợp với CSS, phần xử lí thông tin từ Front-end ( Back-end ) sử dụng WinJS được tích hợp sẵn trong Visual Studio. Cuối cùng là về hệ thống lưu trữ CSDL của ứng dụng thì sử dụng MongoDB để lưu trữ và truy suất dữ liệu.

**4.2 Thiết kế giao diện, các chức năng của ứng dụng**

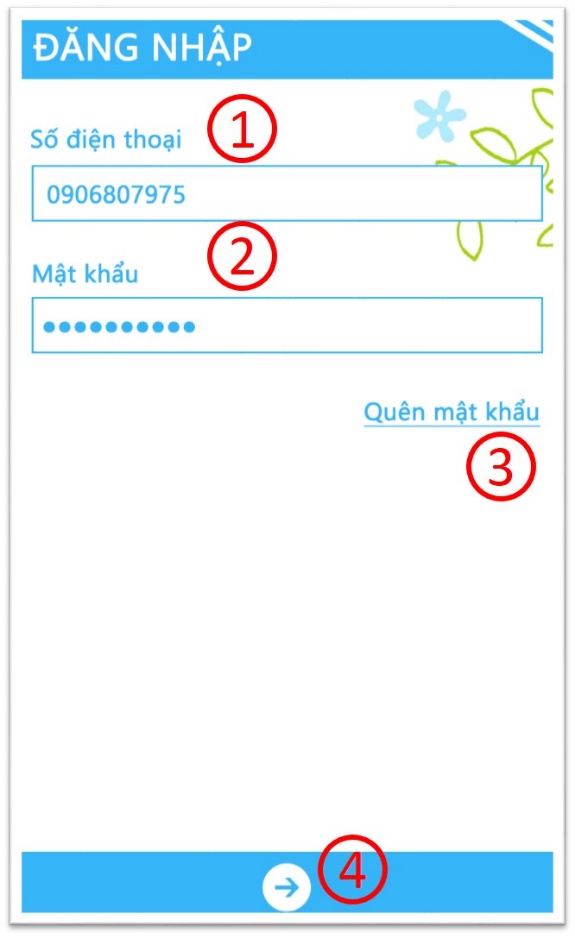
Giao diện mở đầu:



Chức năng:

1. Đăng nhập: nếu đã có tài khoản, người dùng nhấp vào đây để đăng nhập.
2. Đăng ký: nếu chưa có tài khoàn, người dùng phải nhấp vào đây để tạo tài khoản mới.

Giao diện đăng nhập:



Chức năng:

1. Tên tài khoản: khi đăng kí, người dùng sẽ được server cấp 1 id dựa trên số điện thoại của mình, id này là duy nhất, không trùng với bất kì id nào, và server sẽ kiểm tra thông tin đăng nhập dựa trên id này.
2. Mật khẩu đăng nhập: mỗi id sẽ đi kèm với 1 mật khẩu đã được đăng kí, và mật khẩu này sẽ được server mã hóa để bảo mật an toàn cho người dùng.
3. Quên mật khẩu: khi người dùng quên mật khẩu, nhấp vào đây ứng dụng sẽ chuyển sang mục Quên mật khẩu.
4. Login: khi nhấp vào đây, ứng dụng sẽ gửi id và mật khẩu của người dùng lên server để kiểm tra và hiển thị từng thông báo tương ứng với mỗi trường hợp của server phản hồi.

Giao diện đăng ký:



Chức năng:

1. Thông tin đăng ký: nhưng thông tin cơ bản mà ứng dụng yêu cầu người dùng phải nhập.
2. Điều khoản sử dụng: những quy định, yêu cầu và nội quy mà ứng dụng bắt buộc người dùng phải chấp hành, nếu không sẽ có từng biện pháp xử lý tương ứng.
3. Đăng ký: khi nhấp vào đây có nghĩa là mọi thông tin người dùng cung cấp sẽ được gửi lên server để lưu vào CSDL, đồng thơi người dùng cũng đã đồng ý với tất cả điều khoản sử dụng của ứng dụng.

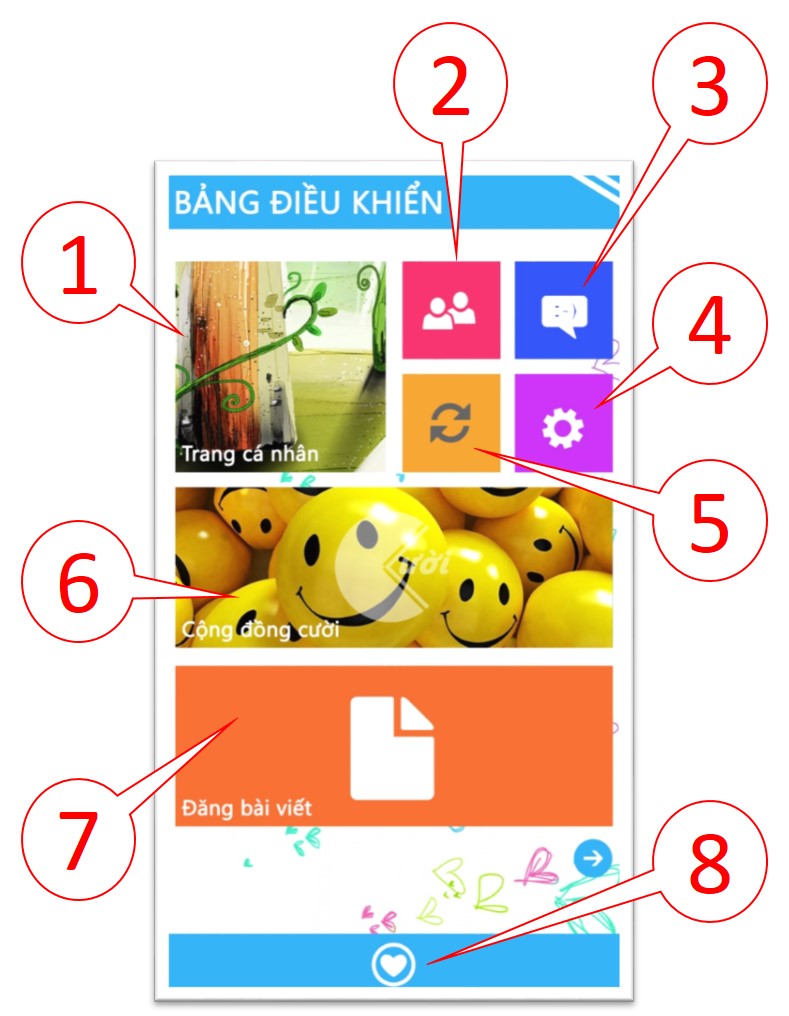
Giao diện quên mật khẩu:



Chức năng:

1. Số điện thoại: khi quên mật khẩu, người dùng phải nhập số điện thoại đã đăng ký, đồng thời phải nhắn tin theo đúng cú pháp đến tổng đài được cung cấp bởi server.
2. Mã xác nhận: sau khi nhận mã xác nhận từ server gửi về, người dùng phải nhập vào trong vòng 5 phút, nếu không, mã xác nhận sẽ tự động hủy, người dùng phải nhắn tin lại để lấy mã mới.
3. Lấy lại mật khẩu: khi nhấp vào, ứng dụng gửi số điện thoại kèm theo mã xác nhận của người dùng lên server để kiểm tra, nếu dúng, ứng dụng sẽ chuyển sang mục phục hồi mật khẩu.

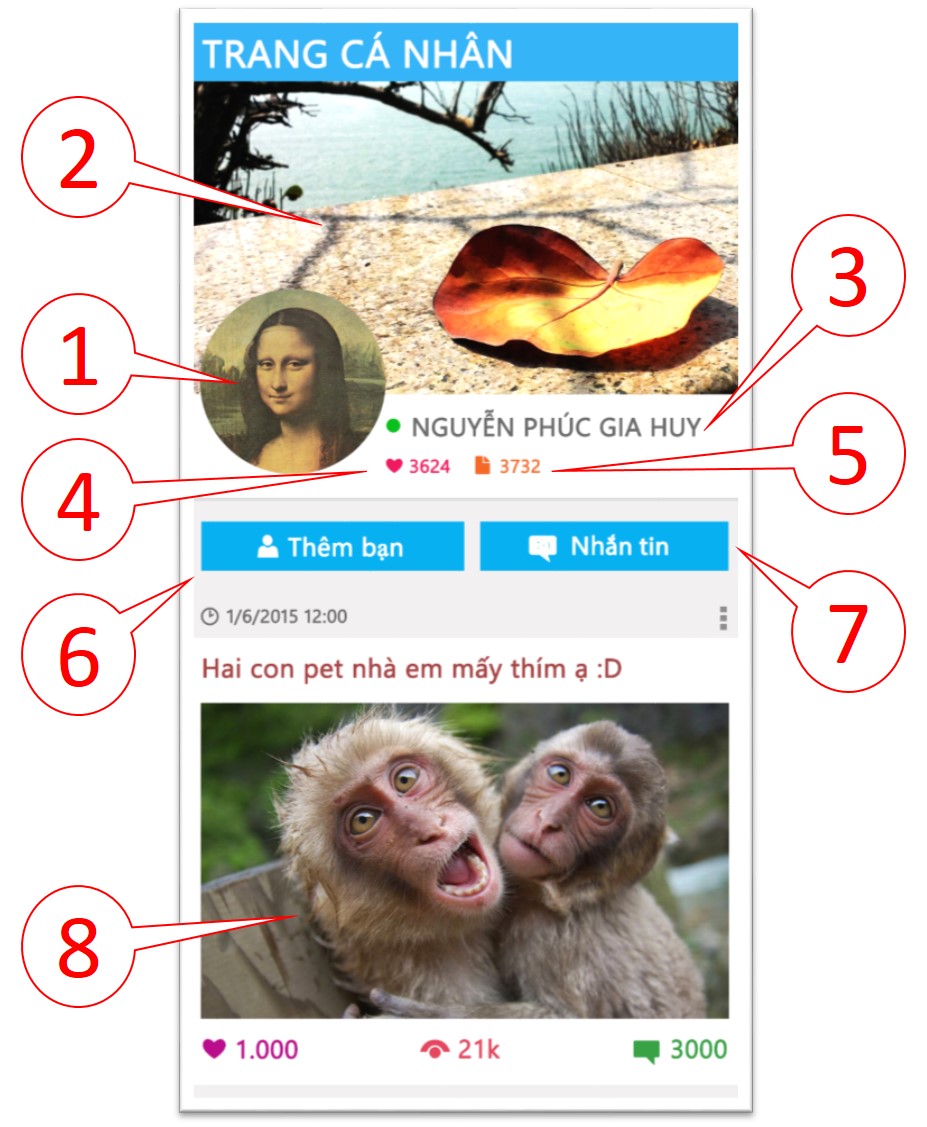
Giao diện bảng điều khiển:



Chức năng:

1. Trang cá nhân: hiển thị tất cả thông tin của người dùng, bao gồm số lượt like, report, tất cả bài viết đã đăng ( bao gồm đã kiểm duyệt và chưa kiểm duyệt ) ..v.v..
2. Danh sách bạn bè: hiển thị tất cả những người dùng đã kết bạn.
3. Tin nhắn: nơi lưu trữ tất cả tin nhắn của người dùng.
4. Cài đặt: tại đây người dùng có thể chỉnh sửa 1 số thông tin cá nhân của bản thân, cũng như thay đổi màu nền của ứng dụng hoặc 1 chức năng khác.
5. Gọi thoại: người dùng sẽ thực hiện cuộc gọi thoại miễn phí, tận dụng camera trước của điện thoại để trò chuyện với những người dùng khác trong danh sách bạn.
6. Cộng đồng cười: ứng dụng chuyển sang mục cộng đồng chung, hay có thể gọi là đại sảnh, nơi mà tất cả mọi bài đăng ( đã được kiểm duyệt ) của người dùng khác được đưa lên.
7. Đăng bài: người dùng có thể tự do đăng bài viết theo ý của mình, ứng dụng sẽ hiển thị ra 3 lựa chọn cho người dùng tùy chọn, gồm: đăng ảnh, đăng video hoặc tạo ảnh.

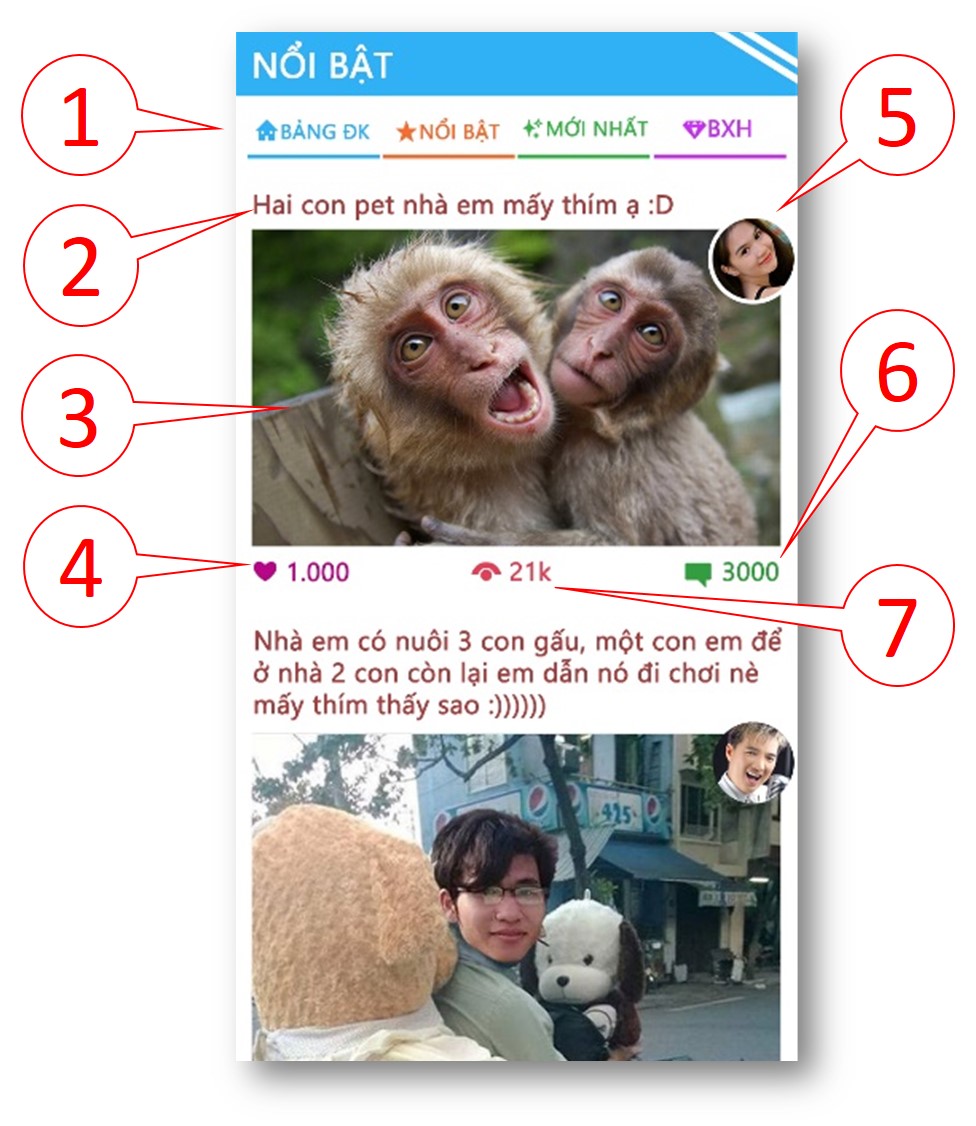
Giao diện trang cá nhân:



Chức năng:

1. Ảnh đại diện: hiển thị ảnh đại diện của người dùng, nếu người dùng chưa chọn ảnh nào, ứng dụng tự động hiển thị ảnh đại diện mặc định cho người dùng.
2. Ảnh bìa: tương tự như ảnh đại diện.
3. Tên người dùng: tên hiển thị của người dùng, người dùng có thể tự do thay đổi tên này trong 1 khoảng thời gian nhất định.
4. Số lượt like: tổng số lượt like của người dùng.
5. Số bài đăng: tổng số bài đăng của người dùng.
6. Thêm bạn: khi nhấp vào đây, ứng dụng gửi yêu cầu kết bạn lên server, và server sẽ tìm và phản hồi lại chính xác tên người kết bạn kèm yêu cầu đến người dùng còn lại.
7. Nhắn tin: ứng dụng sẽ chuyển người dùng sang mục tin nhắn, người dùng có thể thoải mái nhắn tin, mọi tin nhắn đều được gửi lên server và lưu lại vào trong CSDL tương ứng.
8. Tất cả những bài đăng cảu người dùng sẽ được hiển thị tại đây dưới dạng danh sách trượt.

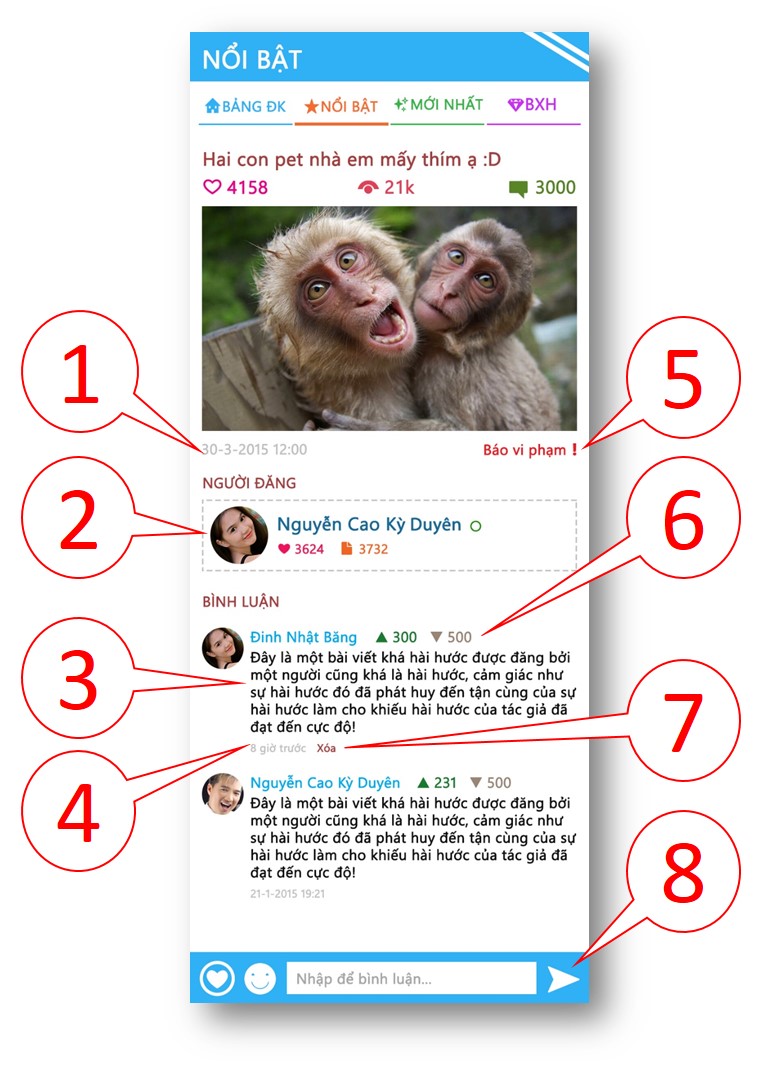
Giao diện cộng đồng cười ( hoặc đại sảnh )



Chức năng:

1. Đề mục: hiển thị tất cả những mục mà người dùng có thể truy cập, người dùng chỉ cần vuốt sang trái hoặc phải để thay đổi mục.
2. Tiêu đề bài viết
3. Ảnh bài đăng: nếu bài viết có nhiều ảnh, ứng dụng sẽ lựa chọn ảnh đầu tiên trong bài, nếu bài viết có 1 ảnh thì chọn ảnh đó.
4. Số lượt like bài viết: người dùng có thể nhấp vào đây để tăng lượt like cho bài viết lẫn chủ bài viết mà không cần phải truy cập vào bài viết.
5. Ảnh đại diện của chủ bài viết: nhấp vào đây ứng dụng sẽ chuyển sang trang cá nhân của người dúng đó.
6. Lượt bình luận: hiển thị số lượt bình luận của bài viết.
7. Lượt xem: hiển thị tất cả lượt xem của bài viết.

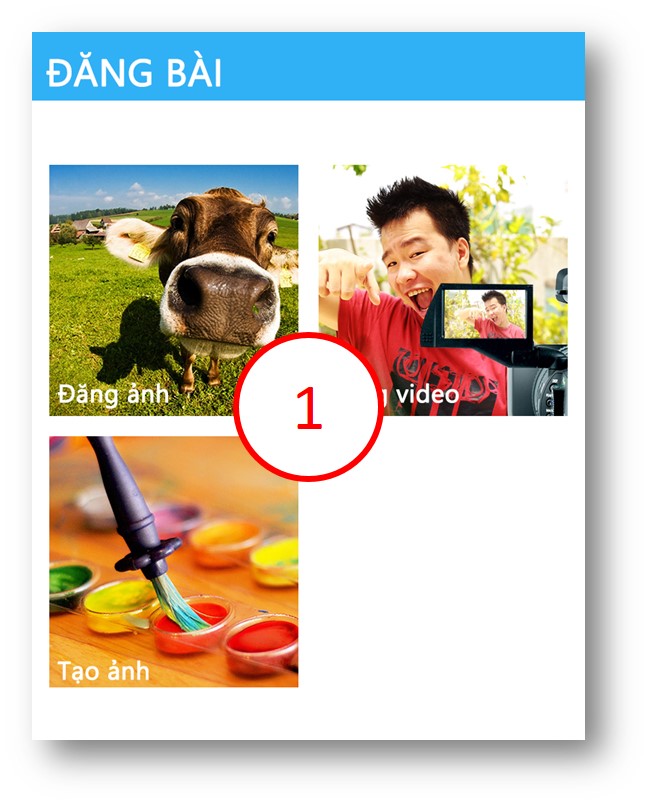
Giao diện bài viết đơn:



Chức năng:

1. Ngày giờ của bài viết
2. Thông tin chủ bài viết: hiển thị ảnh đại diện, trạng thái online/offline, lượt like, bình luận. Khi người dùng khác nhấp vào đây sẽ chuyển đến trang cá nhân của chủ bài viết.
3. Thông tin bình luận: hiển thị nội dung bình luận, ảnh đại diện. Tương tự thông tin chủ bài viết, khi nhấp vào sẽ chuyến đến trang cá nhân của người dùng đó.
4. Ngày giờ bình luận
5. Report: khi người dùng cảm thấy bài viết vi phạm, thì sẽ nhấp vào đây, ứng dụng ngay lập tức gửi thông tin bài viết kèm chủ bài viết lên server chờ admin vào giải quyết, lúc này, bài viết vẫn còn tồn tại.
6. Xếp hạng lượt bình luận: bình luận nào có số lượt xếp hạng cao hơn sẽ được đưa lên đầu tiên, và ngược lại với bình luận có lượt xếp hạng thấp.
7. Xóa bình luận: chỉ có chủ của bình luận đó mới thấy nút này và được quyền xóa bình luận, những người dùng khác sẽ không thấy nút này.
8. Thanh bình luận: nơi mà người dùng nhập bình luận của mình, những thông tin sẽ được gửi lên server để cập nhật chính xác cho bài viết đó.

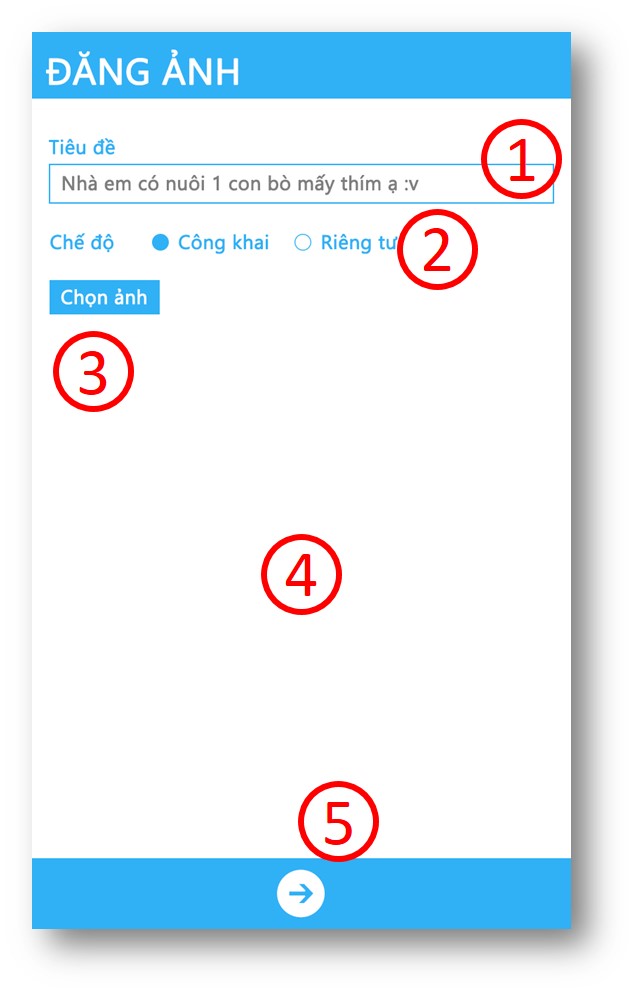
Giao diện đăng bài:



Chức năng:

1. Cách đăng bài: sẽ có 3 sự lựa chọn để đăng bài viết tùy thuộc vào nhu cầu và ý tưởng của người dùng.

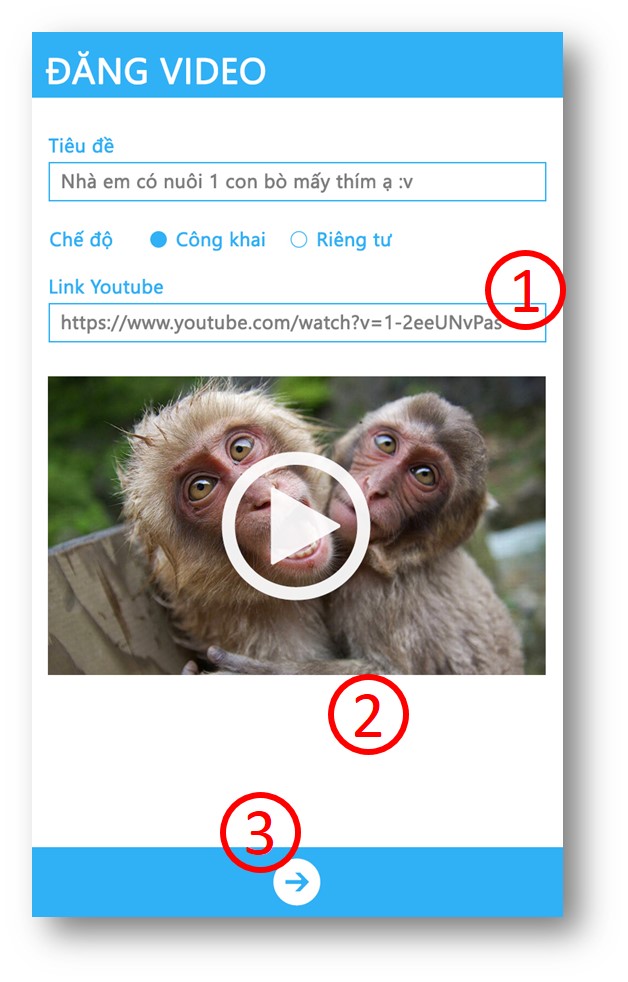
Giao diện đăng ảnh:



Chức năng:

1. Tiêu đề bài đăng: người đăng có thể ghi bất kì nội dung gì, nhưng không được vượt quá 256 ký tự.
2. Chế độ đăng: Bài viết khi đăng thành công vẫn sẽ nằm trong hàng đợi, và đợi admin kiểm duyệt, dù cho đang ở chế độ nào. Người dùng có thể thay đổi thay đổi chế độ bài viết bất cứ khi nào muốn.
3. Chọn ảnh: khi nhấn nút này, ứng dụng tự động chuyển đến kho ảnh trong điện thoại của người dùng và yêu cầu chọn ảnh.
4. Hiển thị: sau khi chọn ảnh thành công, ảnh sẽ được hiển thị tại khung này.
5. Đăng bài viết: ảnh được tải và lưu trữ trên server. Sau khi quá trình tải lên hoàn tất, ứng dụng sẽ lấy được link ảnh và gửi kèm với tiêu đề bài đăng lên server lưu trữ và xử lý.

Giao diện đăng video:



Chức năng:

1. Link youtube: người dùng nếu muốn đăng video thì bắt buộc phải dán link youtube vào đây.
2. Hiển thị: Link youtube sau khi được sử lý sẽ hiển thị tại đây.
3. Đăng video: khi nhấn nút này, link youtube sẽ được ứng dụng xử lý theo đúng định dạng để hiển thị, sau đó link kèm thông tin bài viết sẽ được gửi lẹn server lưu trữ và chờ đợi admin kiểm duyệt.

**4.3 Cấu trúc CSDL**

Do ứng dụng sử dụng MongoDB để lưu trữ, cho nên các trường, thuộc tính của dữ liệu không bị ràng buộc, có thể mở rộng tùy ý. Sau đây là 1 bản ghi ( record ) dữ liệu mẫu của 1 user:

*{*

*"\_id" : ObjectId("5581270444d7573a01000003"),*

*"name" : "Ga9xvn",*

*"phone" : "09123456987",*

*"pass" : "SsoC0l7VGp3525a62cca9830f176c98a5c8936e407",*

*"id" : 30,*

*"avatar" : "http://128.199.78.193:3000/uploaded/files/avatar.jpg",*

*"cover" : "http://128.199.78.193:3000/uploaded/files/cover.jpg",*

*"online" : 0,*

*"like" : 0,*

*"friends" : [ ],*

*"blacklist" : [ ],*

*"level" : 0,*

*"date" : "2015-06-17T14:28:59+07:00"*

*}*

Với mỗi người dùng khi đăng ký tài khoản, sẽ được server cấp cho 1 “**\_*id”*** duy nhất và không trùng với các record khác, và “***pass”*** mà hóa trước khi lưu vào, đảm bảo tính toàn vẹn của dữ liệu. Đồng thời, các trường dữ liệu ( field ) như ***“name”***, ***“phone”*** ..v.v.. đều được lưu dưới dạng chuỗi, không phân biệt kiểu dữ liệu số hay mẫu, sẽ hỗ trợ cho việc quản lý và truy xuất nhanh hơn hệ CSDL MySQL thông thường.

Ở trường ***“friend”*** hay ***“blacklist”*** tương ứng với 2 thông tin cần lưu trữ của người dùng là bạn bè và danh sách người dùng chặn, kiểu dữ liệu được lưu trữ ở đây vẫn là chuỗi, nhưng dưới dạng 1 mảng các chuỗi kí tự. Các mảng này có thể lưu trữ hàng trăm, thậm chí hàng ngàn phần tử chuỗi khác nhau mà không hề ảnh hưởng đến CSDL chung của bản ghi, nhưng vẫn đảm bảo được tốc độ truy vấn của MongoDB.

Tương tự, dưới đây là 1 record mẫu của 1 bài viết bất kì:

*{*

*"\_id" : ObjectId("5582d3c7f7f2277f22000002"),*

*"phone" : "0906807975",*

*"title" : "Anh đã làm gì mà không có gì là thế mà ",*

*"type" : "image",*

*"url": "http://128.199.78.193:3000/uploaded/files/WP\_20150618\_004.jpg",*

*"id" : 19,*

*"like" :*

*[*

*"0906807975"*

*],*

*"view" : 0,*

*"category" :*

*[*

*"0"*

*],*

*"comments" :*

*[*

*{*

***"phone"*** *: "0906807975",*

***"name"*** *: "Dinh Nhat Bang",*

***"content"*** *: " Nhập để bình luận...",*

***"like"*** *: [ ],*

***"dislike"*** *: [ ],*

***"date"*** *: "2015-06-18T21:08:41+07:00"*

*}*

*],*

*"date" : "2015-06-18T21:08:41+07:00"*

*}*

Về cơ bản, thông tin lưu trữ giữa 2 record thông tin người dùng và bài viết có thể không liên quan nhau lắm, nhưng thật ra giữa 2 record này đều có chung 1 thông tin, đó là số điện thoại hay trường ***“phone”*** , đây chính là cầu nối giữa 2 record, tuy thông tin của người dùng bị tách ra, nhưng chỉ cần 2 lệnh truy vấn thông thường, là ta có thể lấy được tất cả thông tin của người dùng có số điện thoại đó, đây có thể xem như là 1 khóa mà ta hay dùng trong MySQL thông thường.

Về tổng quan, mỗi người dùng sẽ có những thông tin lưu trữ sau đây:

|  |  |
| --- | --- |
| **Accounts collection ( thông tin cá nhân )** | |
| **Tên trường** | **Ý nghĩa** |
| Name | Tên hiển thị ở trang cá nhân |
| Phone | Số điện thoại, dùng làm id đăng nhập |
| Pass | Mật khẩu đăng nhập |
| Id | Số thứ tự thành viên |
| Avatar | Link ảnh chứa ảnh đại diện |
| Cover | Linh ảnh chứa ảnh bìa |
| Online | Trạng thái online/offline |
| Like | Tổng số lượt like |
| **Friends** | Danh sách bạn bè |
| Blacklist | Dah sách người dùng chặn |
| Level | Cấp độ thành viên |
| Date | Ngày đăng ký |
| **Friends field ( danh sách bạn bè )** | |
| Phone | Id của người kết bạn |
| Status | Trạng thái của yêu cầu kết bạn |

|  |  |
| --- | --- |
| **Posts collection ( thông tin bài viết )** | |
| **Tên trường** | **Ý nghĩa** |
| Name | Tên hiển thị bài viết |
| Phone | Số điện thoại, dùng làm id đăng nhập |
| Title | Tiêu đề bài viết |
| Type | Dạng bài viết |
| URL | Link ảnh chứa ảnh bài viết |
| Id | Số thứ tự của bài viết |
| Like | Danh sách những id thích bài viết |
| View | Tổng số lượt xem |
| Category | Trạng thái bài viết ( nổi bật/mới nhất ) |
| **Comments** | Dah sách những bình luận |
| Date | Ngày đăng bài |
| **Comments field ( danh sách những bình luận )** | |
| Phone | Id của người bình luận |
| Name | Tên cùa người bình luận |
| Content | Nội dung bình luận |
| Like | Số lượt thích bình luận |
| Dislike | Số lượt không thích bình luận |
| Date | Ngày đăng |

**TÀI LIỆU THAM KHẢO**

1. Chương II – Phân tích yêu cầu, hệ thống

* <http://voer.edu.vn/m/phan-tich-thiet-ke-he-thong-thong-tin/84987530>

1. Chương II – Use Case

* <http://iviettech.vn/blog/543-ban-ve-use-case-use-case-diagram.html>
* <http://agilemodeling.com/artifacts/useCaseDiagram.htm>
* <http://www.victoria.ac.nz/lals/research/usecase/working%20files/files/template.html>

1. Chương III – NodeJS

* <http://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js>
* <http://techmaster.vn/posts/33428/nodejs-la-gi-va-tai-sao-toi-nen-hoc-lap-trinh-nodejs>
* <http://fsd14.com/post/20-nhap-mon-nodejs>
* <http://techmaster.vn/posts/33448/nodejs-cho-nguoi-moi-bat-dau>

1. Chương III – MongoDB

* <http://docs.mongodb.org/manual/core/document/>
* <http://www.quantrimang.com.vn/print/84584.aspx>
* <http://iviettech.vn/blog/543-ban-ve-use-case-use-case-diagram.html>

1. ***MEAN Machine - A beginner's practical guide to the JavaScript stack*** by Chris Sevilleja and Holly Lloyd
2. ***The Node Beginner Book - A comprehensive Node.js tutorial*** by Manuel Kiessling
3. ***Programming Windows Store Apps with HTML, CSS, and JavaScript (2nd Edition)*** by Kraig Brockschmidt
4. ***MongoDB: The Definitive Guide*** by Kristina Chodorow.
5. Cùng với sự giúp đỡ từ các anh em lập trình viên trên khắp thế giới ở diễn đàn <http://stackoverflow.com/>

**KẾT LUẬN**

Với việc mạnh dạn áp dụng những công nghệ mới nhất hiện tại trên thế giới để xây dựng nên 1 ứng dụng mạng xã hội kết hợp với trang giải trí, nhóm em hoàn toàn tự tin vào khả năng cũng như kiến thức tích lũy được trong quá trình tìm hiều, vừa học vừa làm của mình để tạo nên 1 ứng dụng hoàn chỉnh. Mục tiêu của ứng dụng này là cố gắng cạnh tranh và giành lấy 1 phần người dùng của các ứng dụng mạng xã hội lớn hiện nay như Facebook, Zalô hay 9gag trên điện thoại Windows phone, sau đó sẽ tiến tới việc đưa ứngd ụng sang các hệ điều hành khác như Android, iOS hay thậm chí là trên PC.

Tuy nhiên, trong phạm vi đồ án này, vì các chức năng của ứng dụng quá nhiều, trong quá trình thực hiện phát sinh thêm, vượt quá sự tưởng tượng của nhóm em, cùng với đó là trong quá trình làm còn gặp nhiều lỗi, dẫn tới việc mất thời gian tìm hiểu để sửa lỗi, hậu quả là đến thời hạn báo cáo ứng dụng vẫn chưa hoàn chỉnh, nhóm em thành thật xin lỗi thầy vì sai sót này.

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**