TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

Viện Công nghệ thông tin và Truyền thông

Báo cáo bài tập lớn phân tán

Bách Khoa Xanh

Môn: Phát triển phần mềm phân tán

**Giảng viên hướng dẫn: Ts.Vũ Thị Hương Giang**

Nhóm DSD-26: Nguyễn Văn Công MSSV: 20140514

Dương Thế Cường MSSV: 20140557

Nguyễn Phúc Đạc MSSV: 20140918

Vũ Đức Việt MSSV: 20145254

*Hà Nội, ngày 23 tháng 03 năm* *2018*

**Mục Lục**

[I. Giới thiệu 3](#_Toc511857850)

[1. Giới thiệu đề tài, công nghệ sử dụng. 3](#_Toc511857851)

[2. Cơ sở toán học, mô hình ứng dụng. 3](#_Toc511857852)

[3. Phân công công việc. 7](#_Toc511857853)

[4. Trừu tượng. 3](#_Toc511857854)

[5. Cụ thể. 4](#_Toc511857855)

[6. Chuyên biệt. 6](#_Toc511857856)

[II. Mô tả hệ thống 7](#_Toc511857857)

[1. Tổng quan. 8](#_Toc511857858)

[2. Yêu cầu chức năng. 9](#_Toc511857859)

[3. Yêu cầu phi chức năng. 9](#_Toc511857860)

[4. Yêu cầu giao diện. 10](#_Toc511857861)

[III. Kiến trúc hệ thống 16](#_Toc511857862)

[1. Kiến trúc vật lý. 16](#_Toc511857863)

[2. Kiến trúc logic. 16](#_Toc511857864)

[3. Mô hình, giao thức và tương tác 18](#_Toc511857865)

[IV. Xây dựng hệ thống 18](#_Toc511857866)

[1. Công nghệ nền tảng. 18](#_Toc511857867)

[2. Các công nghệ nào sử dụng cho thành nào ? 18](#_Toc511857868)

[V. Kết luận và phát triển 18](#_Toc511857869)

# Giới thiệu

## Giới thiệu đề tài, công nghệ sử dụng.

* Trường Đại học Bách Khoa Hà Nội có diện tích rộng với số lượng cây trồng tương đối lớn và mật độ phân bố cây không đều. Để có thể chăm sóc tốt cũng như tướitiêu đầy đủ ta cần có kế hoạch phân công công việc hợp lý, áp dụng công nghệ thông tin đảm bảo việc thực hiện là chính xác. Để có thể tối ưu hóa quá trình tưới cây, ta cần một phần mềm quản lý hoạt động tưới cây đó để có thể giảm tối đa thời gian di chuyển của người tưới nước cũng như cây sẽ được đáp ứng lượng nước vừa đủ để không bị khô hạn hay úng nước giúp cây phát triển tốt hơn.
* Xây dựng ứng dụng phân tán “Bách Khoa Xanh”, dùng để hỗ trợ việc tưới nước cho các cây trong khuôn viên trường Đại học Bách Khoa Hà Nội.
* Công nghệ sử dụng: java , apache, java script, android, mysql.

Server chính viết bằng Java vì hỗ trợ nhiều như thế nào: ổn định, free, cộng đồng lớn nên dễ học hỏi.

Dùng Server NodeJs vì muốn nó phân tán, đồng bộ và khá nhanh.

## Cơ sở toán học, mô hình ứng dụng.

+ Bài toán đặt ra vấn đề:

* tìm đường đi tối ưu để tiết kiệm thời gian, công sức của nhân viên
* cần một ứng dụng dùng chung giữa các nhân viên cho phép chia sẻ, gửi nhận, cập nhật dữ liệu trong thời gian thực.

+ Giải thuật Dịjkstra tìm đường đi ngắn nhất từ x đến y trong đồ thị G áp dụng cho bài toán tìm cây, tìm nguồn nước và tưới cây.

+ Mô hình ứng dụng là Client – Server với các Client ở đây sử dụng web và app mobie.

## Trừu tượng.

|  |  |  |
| --- | --- | --- |
| Giải pháp | Thành phần | Tài nguyên |
| Sử dụng giải thuật Dijkstra tìm đường đi tối ưu | Server | * Cơ sở dữ liệu về cây, bản đồ * Bộ nhớ server để lưu trữ * Bộ xử lý tính toán |
| Chia sẻ cập nhật dữ liệu thời gian thực | Server | * Cơ sở dữ liệu người dùng * Cơ sở dữ liệu cây, bản đồ * Bộ nhớ server * Mạng internet |
| Client | * Bộ nhớ trên thiết bị device * Mạng internet |

## Cụ thể.

|  |  |  |
| --- | --- | --- |
| Giải pháp | Thành phần | Nhiệm vụ |
| Sử dụng giải thuật Dijkstra tìm đường đi tối ưu | Server | * Tìm đường đi ngắn nhất theo vị trí người dùng, lượng nước có, vị trí cây |
| Chia sẻ cập nhật dữ liệu thời gian thực | Server | * Gửi thông báo, dữ liệu đến client. * Tiếp nhận, xử lý yêu cầu từ phía client |
| Client | * Kết nối server, * Gửi yêu cầu lấy dữ liệu từ phía server * Sử lý dữ liệu nhận được * Hiện thị cho người dùng |

+Danh sách các API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Api | Chức năng | input | output |
| 1 | getCay | Lấy thông tin cây | idCay | Dữ liệu cây |
| 2 | getListCay | Lấy thông tin tất cả các cây |  | Dữ liệu tất cả các cây |
| 3 | capNhatNuocChoCay | Đi tưới nước cho cây | idThanhVien,idCay, idCay, luongNuoc | Thông báo |
| 4 | baoCaoTinhTrangCay | Báo cáo tình trạng cây | idThanhVien,idCay, idCay, noiDung | Thông báo |
| 5 | getTrace1 | Tìm đường đi từ thành viên đến cây chọn trước | idThanhVien,toaDoXtoaDoY, idCay | Array tọa độ X, Y các điểm cần đi qua |
| 6 | getTrace2 | Tìm đường đi từ thành viên đến cây được chọn ngẫu nhiên | idThanhVien,toaDoXtoaDoY, | Array tọa độ X, Y các điểm cần đi qua |
| 7 | getDiemCapNuoc | Lấy thông tin điểm cấp nước | idDiemCapNuoc | Dữ liệu điểm cấp nước |
| 8 | getListDiemCapNuoc | Lấy thông tin tất cả các điểm cấp nước |  | Dữ liệu tất cả điểm cấp nước |
| 9 | baoCaoTinhTrangDCN | Báo cáo tình trạng điểm cấp nước | idThanhVien,idCay, idCay, noiDung | Thông báo |
| 10 | getTrace1 | Tìm đường đi từ thành viên đến điểm cấp nước chọn trước | idThanhVien,toaDoXtoaDoY, idDiemCapNuoc | Array tọa độ X, Y các điểm cần đi qua |
| 11 | getTrace2 | Tìm đường đi từ thành viên đến điểm cấp nước chọn ngẫu nhiên | idThanhVien,toaDoXtoaDoY, | Array tọa độ X, Y các điểm cần đi qua |
| 12 | getLichSuTuoiCay | Lấy lịch sử tưới của cây đã chọn | idCay | Dữ liệu Lịch sử tưới cây đó |
| 13 | getLichSuTuoiThanhVien | Lấy lịch sử tưới của thành viên đó | idThanhVien | Dữ liệu lịch sử tưới thành viên đó |
| 14 | getThanhVien | Lấy thông tin thành viên | idThanhVien | Dữ liệu thành viên |
| 15 | getListThanhVien | Lấy thông tin tất cả thành viên |  | Dữ liệu tất cả thành viên |
| 16 | login | Đăng nhập | tenTaiKhoan, matKhau | Dữ liệu thành viên |
| 17 | logout | Đăng xuất | idThanhVien | Thông báo |

## Chuyên biệt.

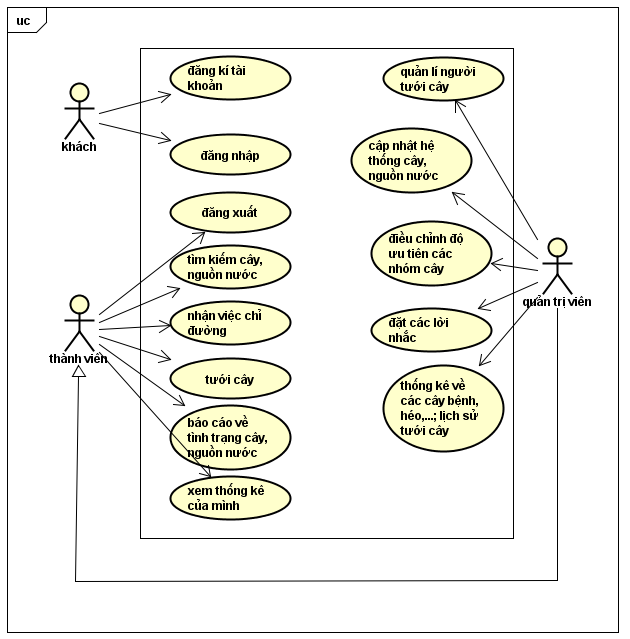
* Server Java theo mô hình Java Spring MVC, Server NodeJs và App Android.
* Thêm sửa xóa dữ liệu cây, loại cây, điểm cấp nước, thành viên, loại thành viên.
* Các phương thức get lấy thông tin như: cây, danh sách cây(điểm cấp nước, thành viên).
* Các phương thức post có như: báo cáo tình trạng cây(điểm cấp nước), , tìm đường đi ngắn nhất từ thành viên đến cây(điểm cấp nước), login logout cho thành viên.

## Phân công công việc.

File Excel Job\_List kèm theo

# Mô tả hệ thống

## Tổng quan.



|  |  |  |  |
| --- | --- | --- | --- |
|  | Số lượng chức năng | Số lượng yêu cầu | Định lượng độ hoàn thiện |
| Kết quả | 11 của thành viên  12 của quản trị viên | Mỗi chức năng có các yêu cầu: yêu cầu chung, riêng, phi chức năng | 80% |

## Yêu cầu chức năng.

+ Phần mềm có 2 tác nhận: Quản lý và nhân viên

* Người dùng được chia làm hai loại chính là: quản lý và nhân viên. Người quản lý và nhân viên đều được cung cấp một tài khoản riêng được tạo bởi nhân viên quản trị hệ để đăng nhập vào ứng dụng.
* Chức năng của người quản lý: quản lý nhân viên, quản lý cây trồng, thông kê.
* Chức năng của nhân viên: xem bản đồ, đánh dấu cây, tưới cây, xem thống kê làm việc.

+ Mỗi vai trò người dùng được thiết lập để có thể sử dụng một số chức năng nào đó. Mỗi chức năng có thể được nhiều vai trò người dùng sử dụng. Mỗi khi có thêm một chức năng mới, quản trị viên cần đưa các thông tin về chức năng này vào phần mềm để quản lý. Sau khi một người dùng đăng nhập thành công, tuỳ thuộc vào các vai trò người dùng mà người đó có, phần mềm sẽ tự động tạo menu chứa các chức năng mà các vai trò người dùng đó được phép sử dụng. Mỗi khi người dùng chọn một chức năng trên menu, giao diện tương ứng với chức năng sẽ được đưa ra.

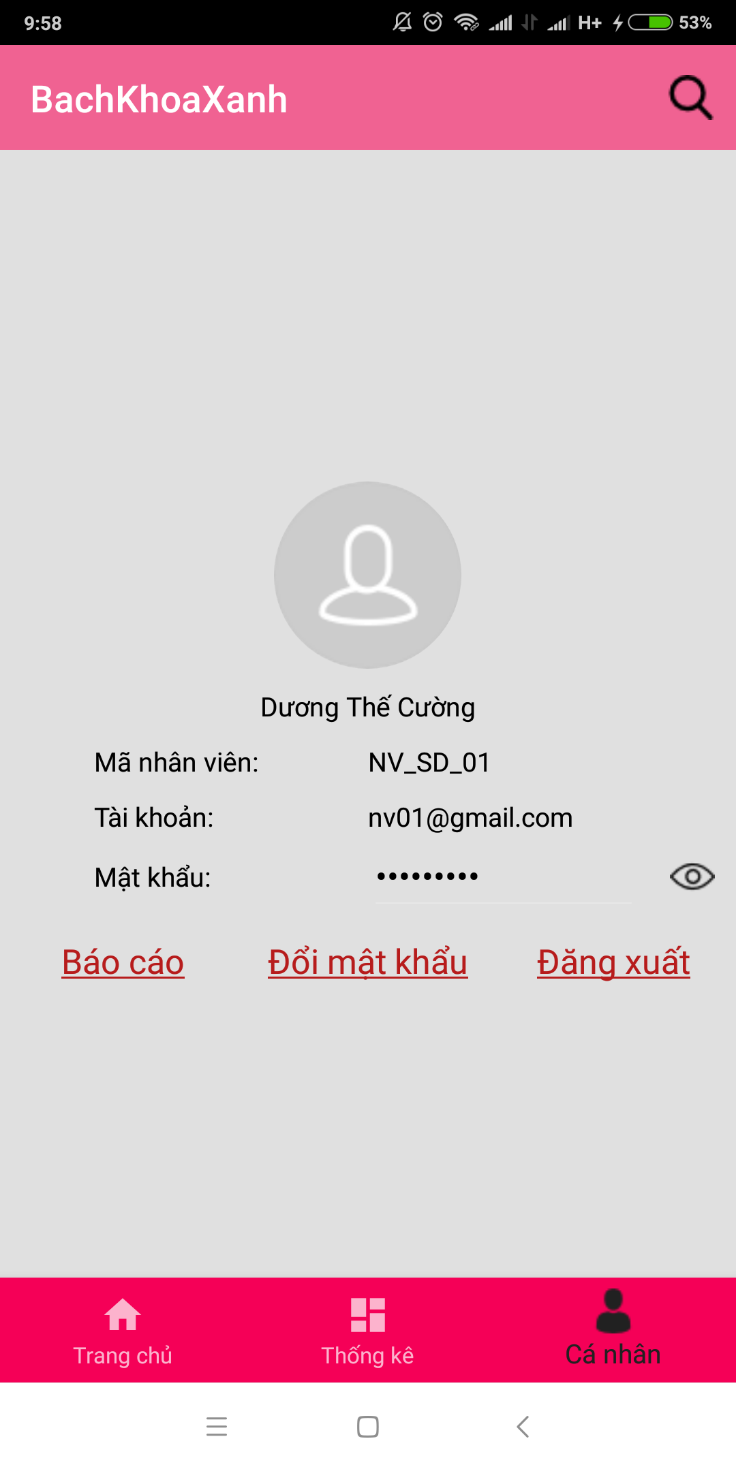
## Yêu cầu phi chức năng.

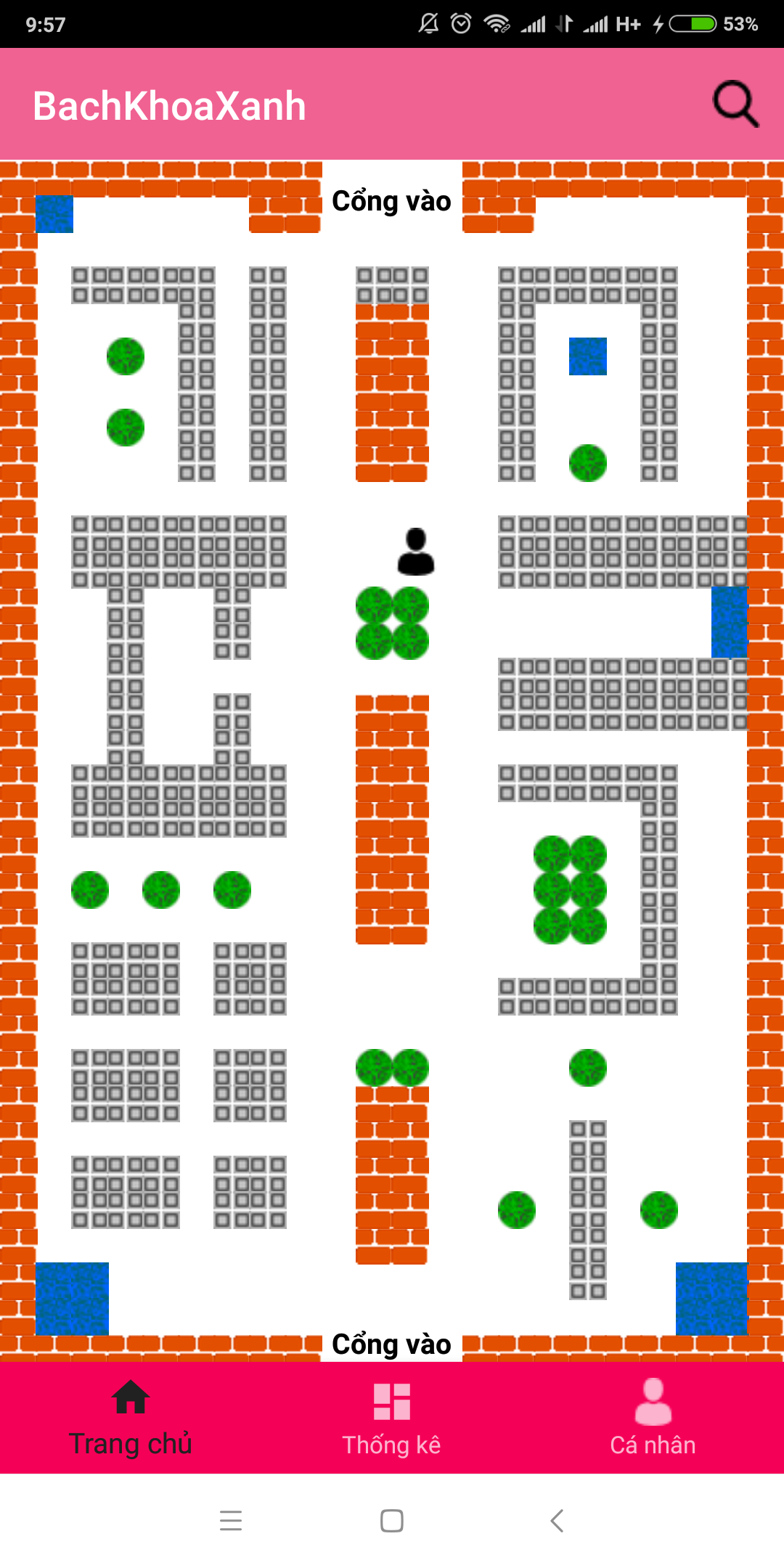
* Hiệu năng, tương thích, khả dụng, tin cậy,…
* Ứng dụng có giao diện đơn giản, thân thiện với người dùng.
* Khả năng truyền, nhận dữ liệu nhanh.

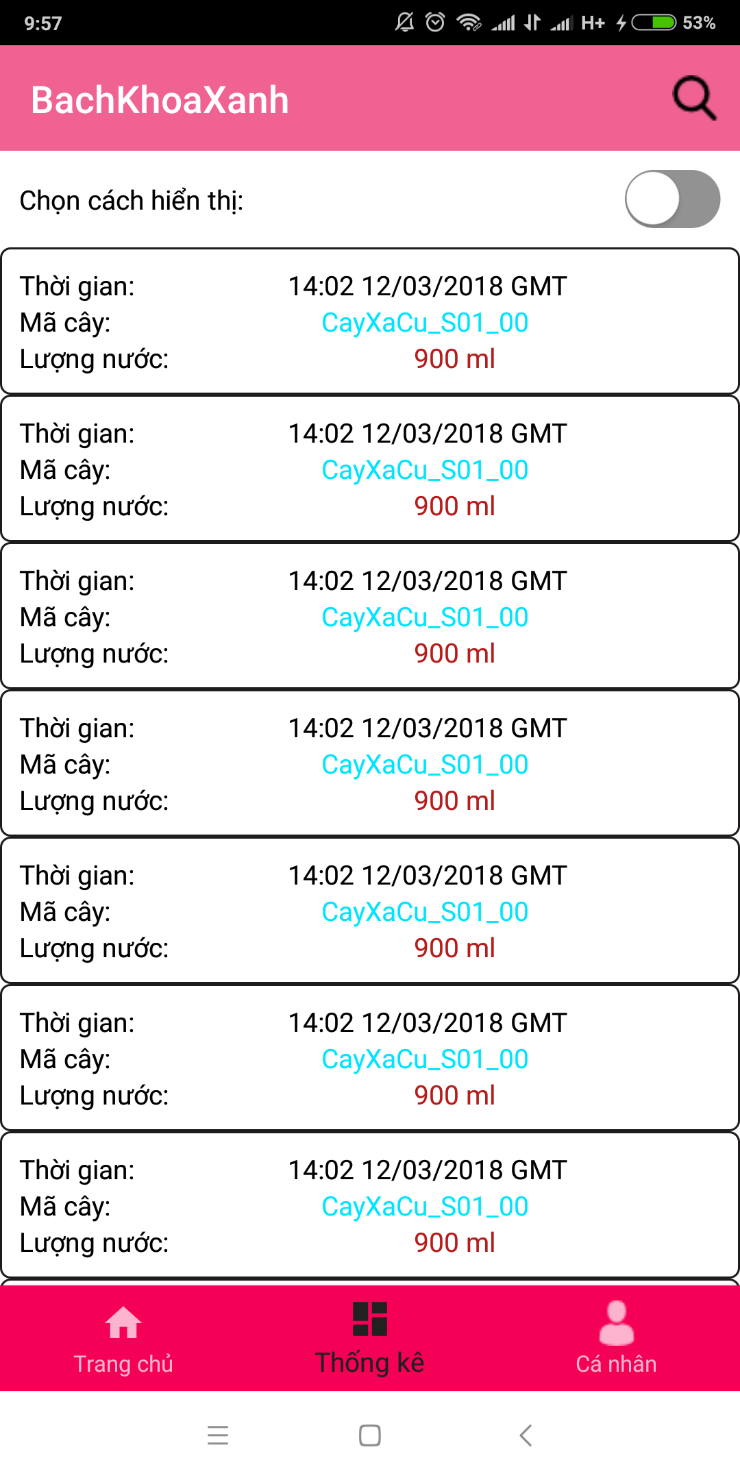
## Giao diện.

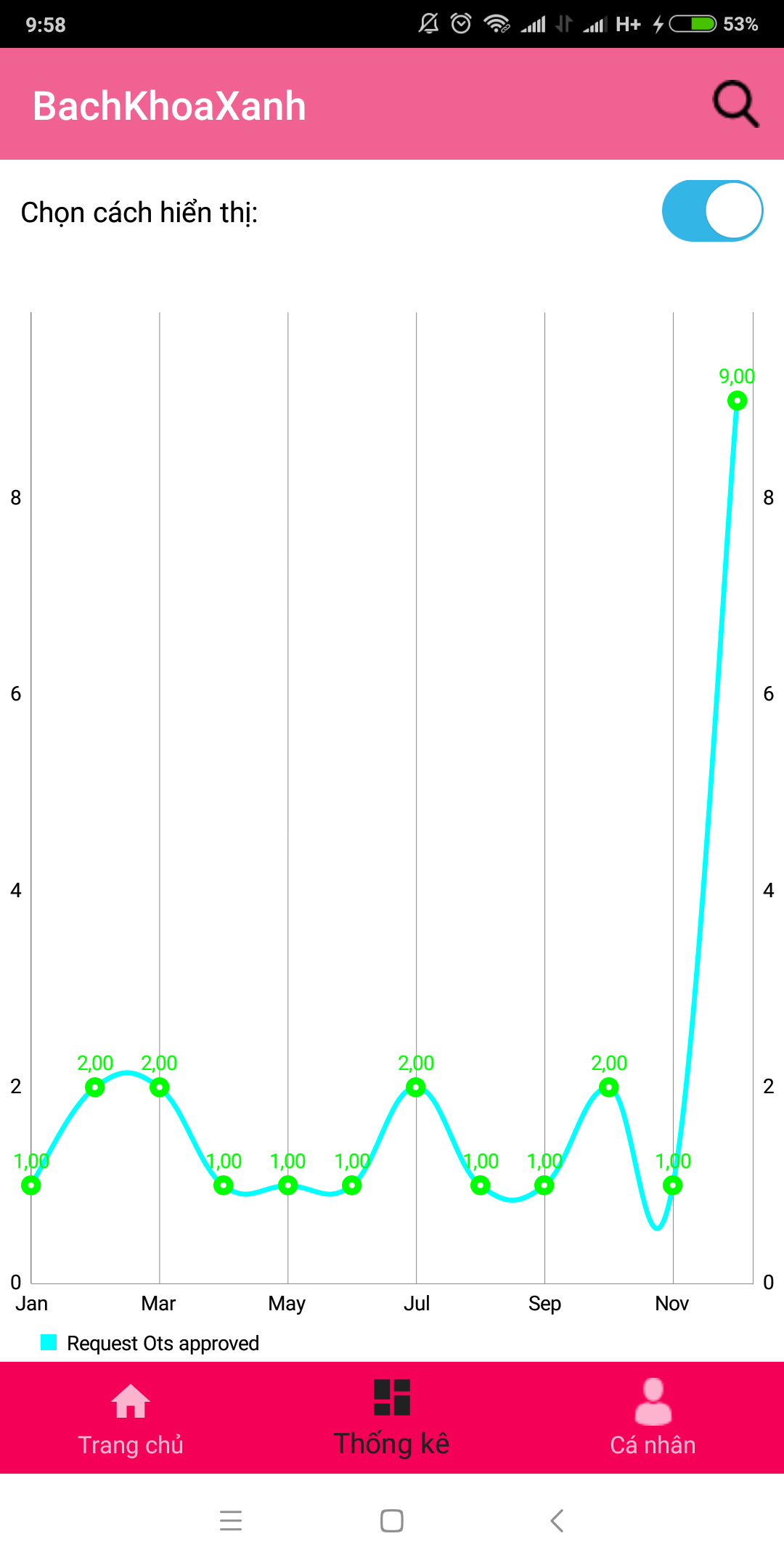
* Dễ sử dụng, tương tác.
* Giao diện web cho quản lý .
* Giao diện app android .

+ Giao diện App điện thoại Android.

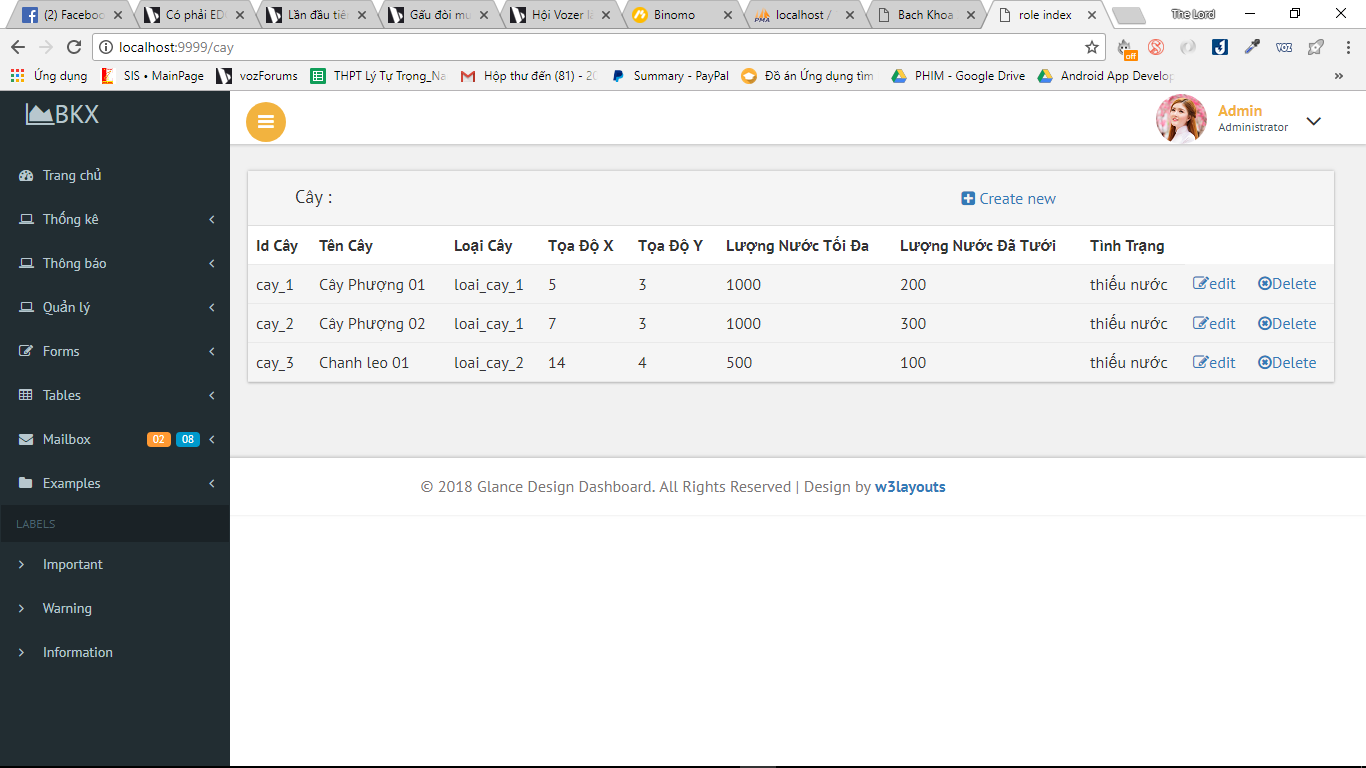


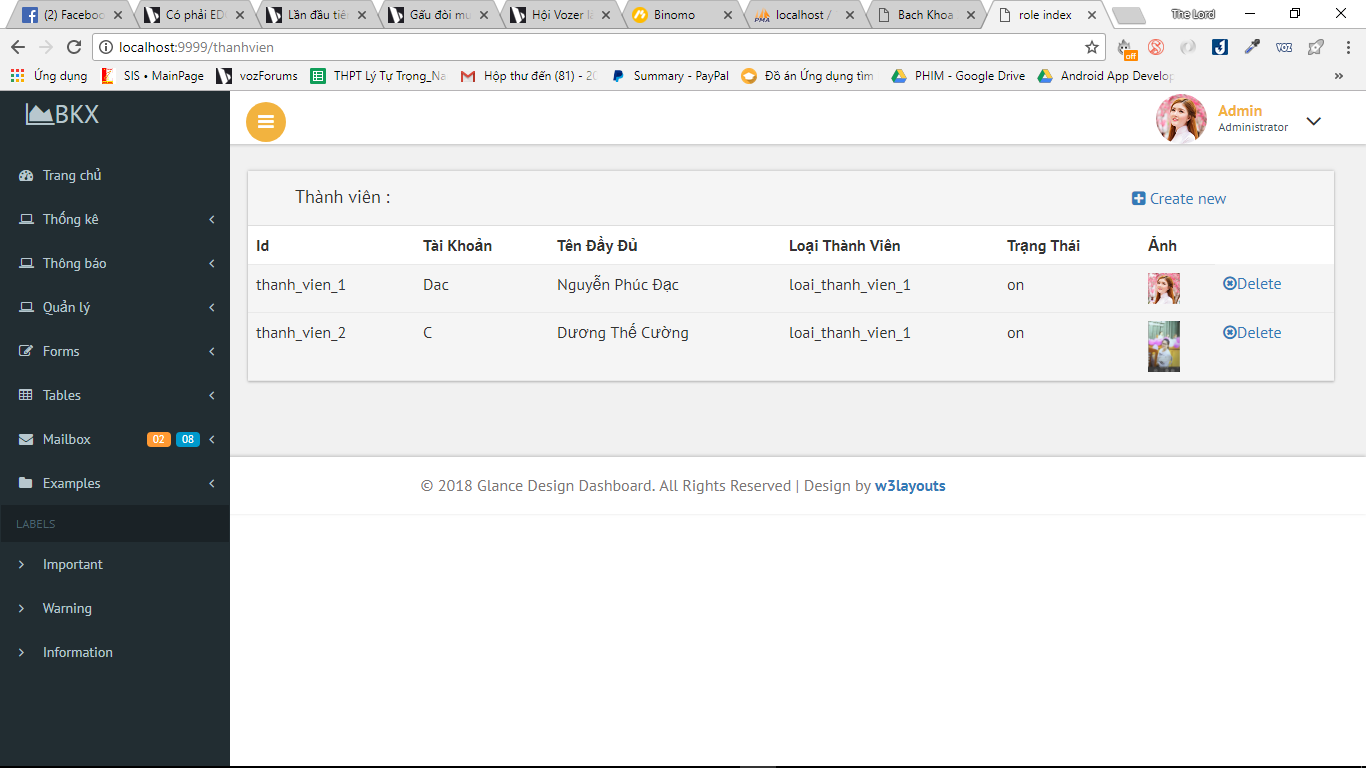


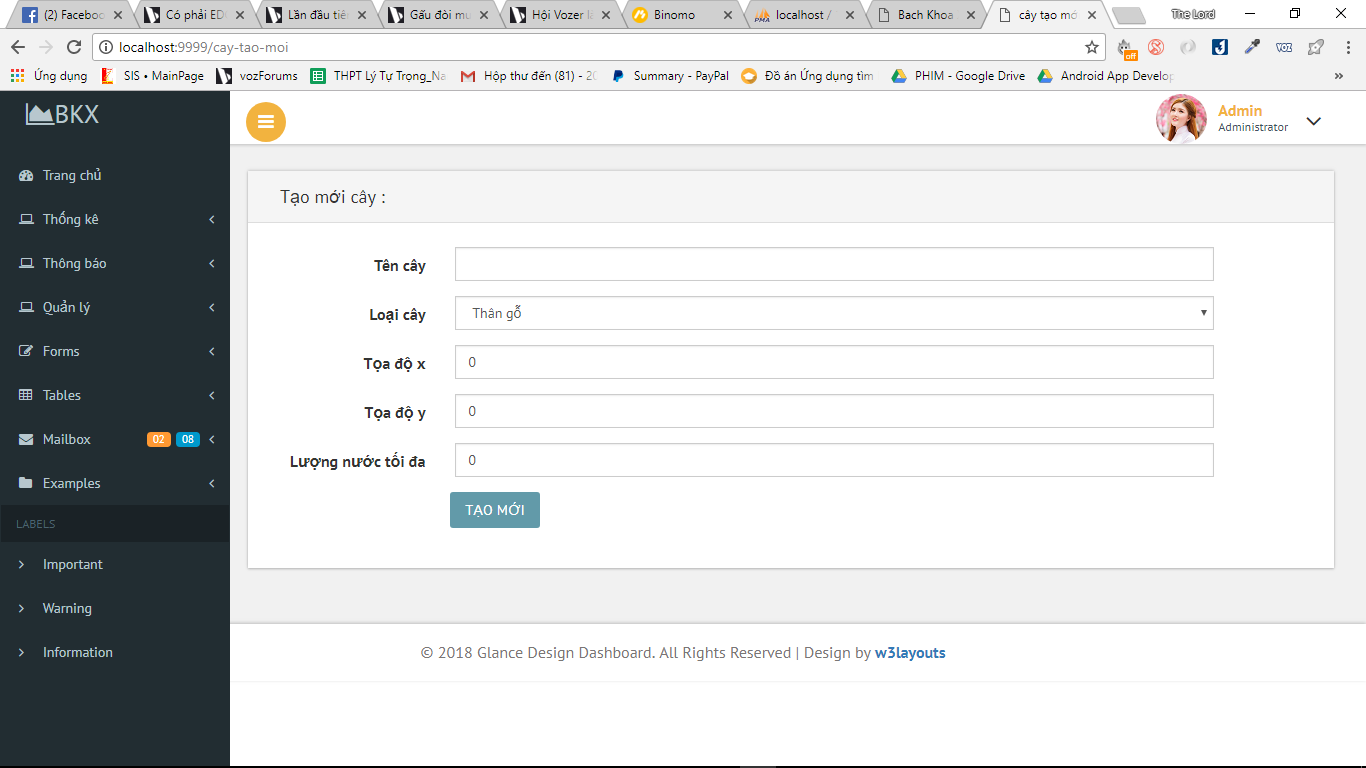


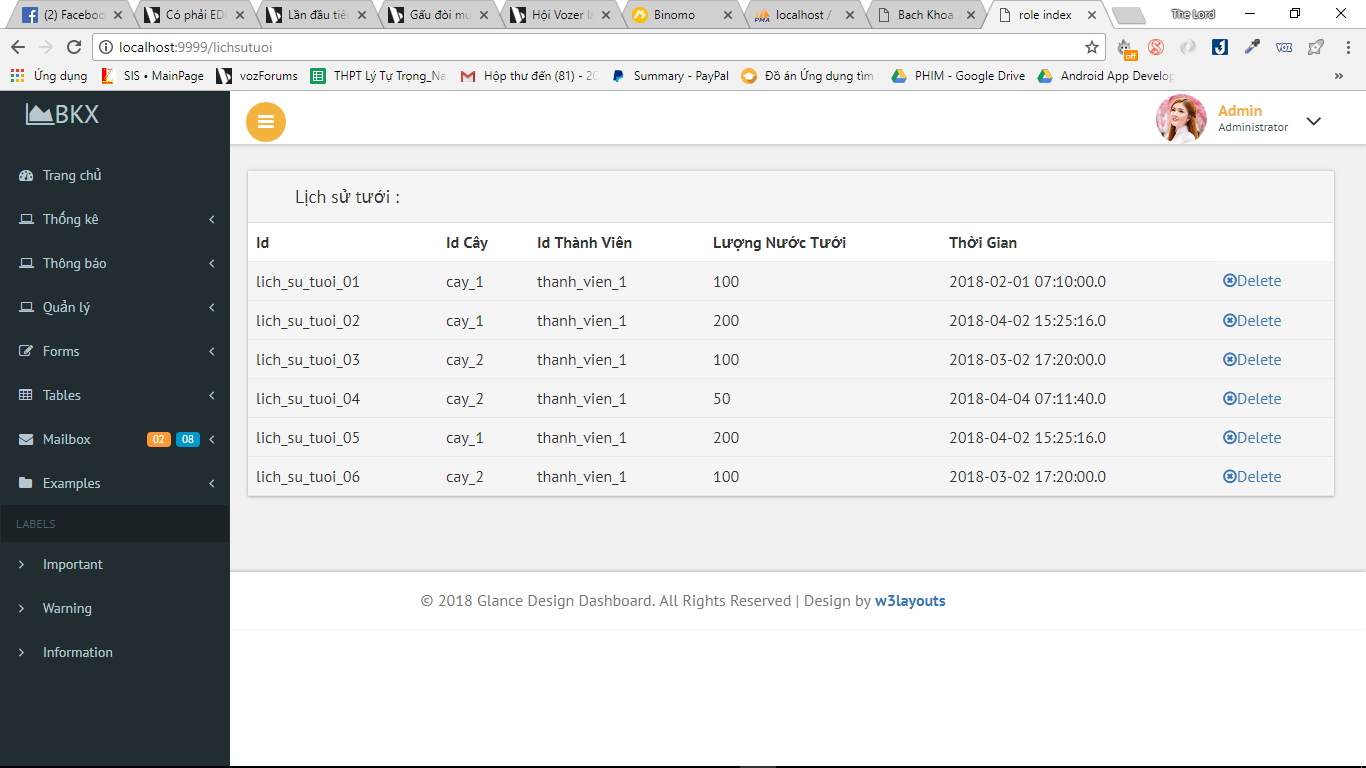


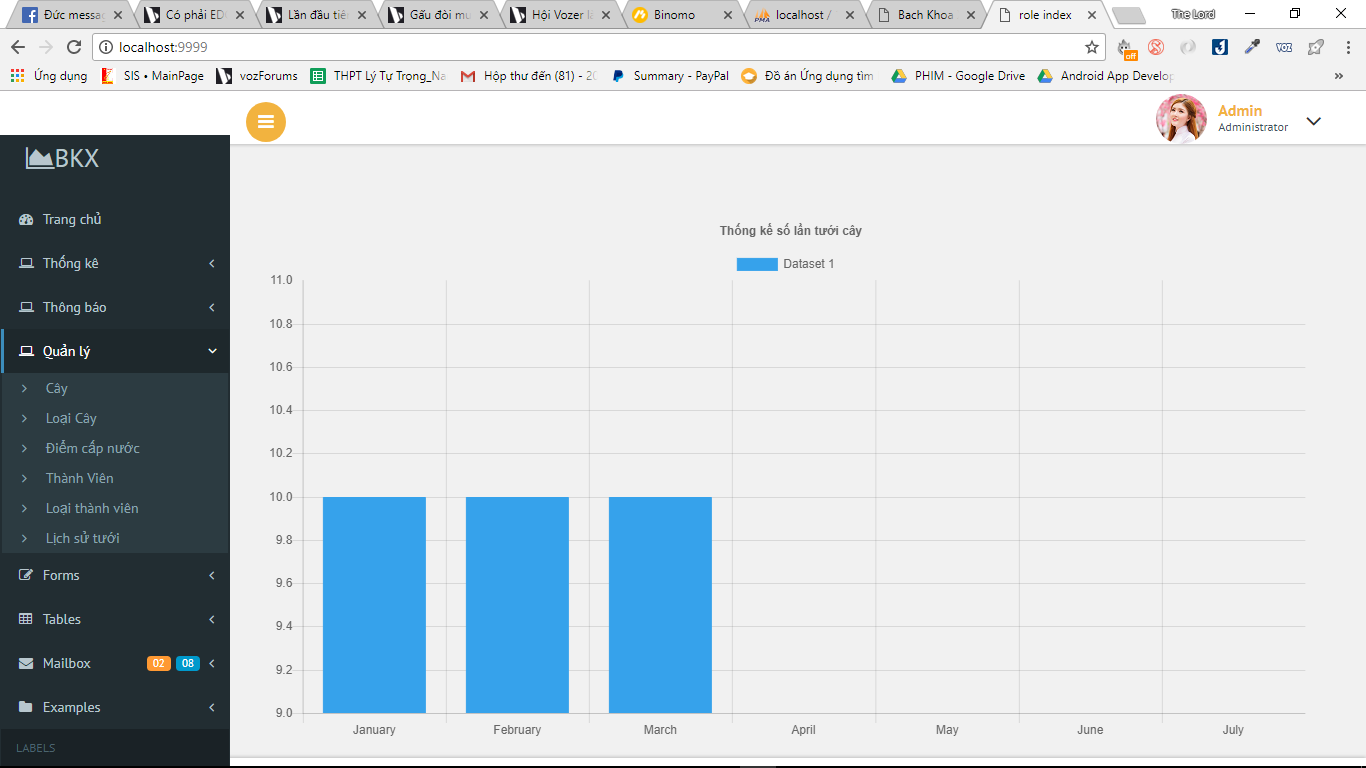
+ Giao diện trên Web của người quản lí.











# Kiến trúc hệ thống

## Kiến trúc vật lý.

+ Bao nhiêu máy, kết nối thế nào, máy nào triển khai thành phần nào ?

* 1 máy chạy server Java và làm admin.
* 1 máy chạy server NodeJs.
* 1 điện thoại chạy App Android với vai trò thành viên tưới cây.
* Kết nối qua mạng Lan Wifi.

## Kiến trúc logic.

+ Thành phần nào đảm nhiệm chức năng nào ?

+ Mô hình phần tán như nào ?

+ Hỗ trợ đồng bộ.

Cơ sở dữ liệu

Server Java

Client

Client

Server NodeJs

+Hệ thống gồm các thành phần:

* Server Java lấy CSDL, cung cấp các API cho các Client kết nối đến;
* Khi có sự thay đổi CSDL thì Server Java gửi Request có thay đổi CSDL đến Server NodeJS.
* NodeJs thông báo đến các Client, các Client cập nhật lại dữ liệu.

+ Server NodeJs: Server NodeJs đóng vai trò là một Server trung gian. Nó có tác dụng đồng bộ dữ liệu thời gian thực cho Server chính Java và các Client. Khi có các thay đổi dữ liệu trên Server Java thì Server Java sẽ gửi thông điệp request đến Server NodeJs, từ đó Server NodeJs sẽ sử dụng socket để gửi thông điệp emit đến các Client.

+ Server Java: là Server chính, đảm nhận CRUD dữ liệu thông qua kết nối với CSDL Mysql, cung cấp các Controller để người quản lý quản lý dữ liệu hệ thồng. Cung cấp các API cho Client, cũng như gửi Request khi có thay đổi CSDL đến cho Server NodeJs để đồng bộ dữ liệu.

+ Phía Client: Giao diện trên Web và trên App Android.

## Mô hình, giao thức và tương tác

+ Mô hình: Client – Server

+ Giao thức: TCP/IP

+ Tương tác: có sự đồng bộ giữa hiển thị trên app Android và trên Web về bản đồ, thành viên, sự di chuyển của thành viên. Đồng bộ, nhất quán về dữ liệu trên các Client.

Bonus: các thuật toán đảm bảo chất lượng (đồng bộ, sao lưu, nhất quán, an toàn)

# Xây dựng hệ thống

## Công nghệ nền tảng.

* Java, NodeJs, Android, Mysql, Apache.
* Kết nối : Socket.Io và TCP.

## Các công nghệ nào sử dụng cho thành nào ?

* Java Web Spring cho Server Java.
* NodeJs cho Server NodeJS.
* Android cho App Android.
* MySql cho quản trị CSDL.
* Apache tomcat cho kết nối Server Java với CSDL.
* Socket.Io và TCP cho kết nối App Android đến các Server.

# Kết luận và phát triển

Đánh giá sản phẩm:

+ Mức độ hoàn thiện: %

* Có các chức năng: thêm sửa xóa CSDL, thống kê, gửi cảnh báo cho người quản trị; có tìm cây, nguồn nước, chỉ đường và tưới nước, thống kê cho thành viên
* Giao diện: có trên app mobie android và trên web
* Độ tương tác:
* Phân tán: hệ thống phân tán trên nhiều thiết bị Client, có Server và Client.

+ Phát triển thêm:

* Có app cho thiết bị IOS
* Dễ dàng thay đổi bổ sung bản đồ
* Các server chạy ổn định, nhanh hơn