Khai Thác Dữ Liệu Trong Siêu Thị

Version 1.2

Sinh viên thực hiện:

1642050 – Nguyễn Thành Phong

1642067 – Nguyễn Lê Quang Thắng

1642076 – Huỳnh Duy Thức

1642084 – Trần Lê Tuấn

**Bảng ghi nhận thay đổi tài liệu**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ngày** | **Phiên bản** | **Mô tả** | **Người thay đổi** |
| 08/07/2017 | 1.0 | Hoàn thành phần 1: level 1, level 2 và level 3   * Đọc dữ liệu và thể hiện lên màn hình * Thao tác trên màn hình mức cơ bản * hao tác trên màn hình mức chi tiết | Nguyễn Thành Phong  Nguyễn Lê Quang Thắng  Huỳnh Duy Thức  Trần Lê Tuấn |
| 05/08/2017 | 1.1 | Hoàn thành phần 2: level 5   * Đồng bộ hóa dữ liệu | Nguyễn Thành Phong  Nguyễn Lê Quang Thắng  Huỳnh Duy Thức  Trần Lê Tuấn |
| 30/08/2017 | 1.2 | Hoàn thành phần 3: level 4   * Khai thác dữ liệu trong siêu thị | Nguyễn Thành Phong  Nguyễn Lê Quang Thắng  Huỳnh Duy Thức  Trần Lê Tuấn |

**BẢNG KẾ HOẠCH CÔNG VIỆC**

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Ghi chú** |
| 1 | Tìm hiểu yêu cầu của phần mềm |  |
| 2 | Phân tích và thiết kế phần mềm |  |
| 3 | Tìm hiểu công cụ thiết kế giao diện và ngôn ngữ lập trình |  |
| 4 | Thiết kế cơ sở dữ liệu |  |
| 5 | Hiện thực level 1,2,3 – Đọc dữ liệu và thể hiện lên màn hình |  |
| 6 | Hiện thực level 5 – Thao tác trên màn hình mức cơ bản |  |
| 7 | Hiện thực level 4 – Khai thác dữ liệu |  |
| 8 | Kiểm định phần mềm & Đóng gói |  |
| 9 | Lập báo cáo |  |

**BẢNG PHÂN CÔNG**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Level** | **Công việc** | **Phân công** | | | | **Ghi chú** |
| **Phong** | **Thắng** | **Thức** | **Tuấn** |
| 1 | 1,2,3 | Phân tích yêu cầu, đặc tả đồ án | 25% | 25% | 25% | 25% |  |
| 2 | Thiết kế cơ sở dữ liệu | 0% | 0% | 50% | 50% |  |
| 3 | Thiết kế giao diện | 50% | 50% | 0% | 0% |  |
| 4 | Tìm công cụ lập trình giao diện | 25% | 25% | 25% | 25% |  |
| 5 | Lập trình giao diện | 50% | 50% | 0% | 0% |  |
| 6 | Viết báo cáo | 0% | 0% | 50% | 50% |  |
| 7 | 5 | Phân tích yêu cầu | 25% | 25% | 25% | 25% |  |
| 8 | Tìm hiểu đồng bộ hóa dữ liệu theo thời gian thực | 25% | 25% | 25% | 25% |  |
| 9 | Tìm hiểu về web socket | 25% | 25% | 25% | 25% |  |
| 10 | Ứng dụng socket-client | 50% | 50% | 0% | 0% |  |
| 11 | Ứng dụng socket-server | 0% | 0% | 50% | 50% |  |
| 12 | Lập trình giao diện | 40% | 40% | 10% | 10% |  |
| 13 | Viết báo cáo lần 2 | 10% | 10% | 40% | 40% |  |
| 14 | 4 | Phân tích yêu cầu | 25% | 25% | 25% | 25% |  |
| 15 | Tìm hiểu tuật toán FP Growth để tìm mẫu phổ biến và luật kết hợp | 25% | 25% | 25% | 25% |  |
| 16 | Chuẩn bị dữ liệu để áp dụng thuật toán FP Growth. Sau đó thực hiện khai thác dữ liệu (Client side) | 50% | 50% | 0% | 0% |  |
| 17 | Áp dụng thuật toán FP Growth (Server side) | 0% | 0% | 50% | 50% |  |
| 18 | Kiểm thử phần mềm | 25% | 25% | 25% | 25% |  |
| 19 | Đóng gói phần mềm | 40% | 10% | 40% | 10% |  |
| 20 | Viết báo cáo lần 3 | 10% | 40% | 10% | 40% |  |

# Giới thiệu

## Mục tiêu

* + *Để dễ dàng quản lý siêu thị cũng như thông tin của các món hàng, cách thức tổ chức các kệ hàng , siêu thị cần có một hệ thống để sử dụng.*
  + *Ngày nay thời đại công nghệ thông tin phát triển, Hệ thống quản lí siêu thị theo cách cổ điển đã không đáp ứng được các nhu cầu về thời gian, kinh tế cũng như nguồn nhân lực. Ví dụ: Lưu thông tin các món hàng vào sổ giấy sẽ mất nhiều thời gian khi ta muốn tìm kiếm và khai thác dữ liệu các món hàng mà khách hàng đã mua.*
  + *Sự ra đời của phần mềm Quản lí siêu thị của nhóm chúng tôi là câu trả lời cho nhu cầu thực tế ngày nay. Phần mềm sẽ lưu những thông tin các món hàng, các kệ hàng, cách bố trí không gian trong siêu thị,…Từ đó sẽ giúp người quản lý siêu thị thực hiện công việc của mình một cách dễ dàng, giảm được thời gian và công sức.*
  + *Bên cạnh đó, phần mềm được thiết kế theo giao diện đồ họa 2D mang đến trải nghiệm cho người dùng một cách chân thật và thực tế nhất.*
  + *Ngoài ra, để phục vụ cho chiến lược kinh doanh, phần mềm sẽ tính toán và cho người quản lý biết được những món hàng được mua phổ biến, dự đoán các món hàng mà khách hàng có thể mua tiếp theo sau khi họ đã chọn một số món hàng cũng như dự đoán sở thích của người mua hàng.*

## Giới hạn

* Phần mềm chưa hỗ trợ cho các thiết bị di động.
* Giao diện đồ họa 2D nên còn hạn chế so với đồ họa 3D.

## Thuật ngữ và từ viết tắt

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuật ngữ** | **Mô tả** |
| 1 | Stakeholder | Những nhân tố ảnh hưởng đến phát triển phần mềm |
| 2 | Workflow | Luồng làm việc |
| 3 | Keyword | Từ khóa |
| 4 | Use case | Được xem như một chức năng hệ thống từ quan điểm người dùng, mô tả yêu cầu của hệ thống về mặt chức năng, mỗi chức năng được biểu diễn bằng 1 hay nhiều use case |
| 5 | CSDL | Cơ sở dữ liệu |
| 6 | API | Application Programming Interface: Giao diện lập trình ứng dụng |
| 7 | IDE | Integrated Development Environment: môi trường phát triển tích hợp |

# Mô tả chi tiết chức năng

## Chức năng

* Các chức năng của hệ thống :
  1. Tìm kiếm món hàng
  2. Thêm món hàng vào kệ
  3. Di chuyển món hàng sang kệ khác
  4. Xóa món hàng khỏi kệ
  5. Cập nhật thông tin món hàng
  6. Xem thông tin món hàng
  7. Xem lịch sử đặt hàng
  8. Thêm kệ hàng
  9. Xóa kệ hàng
  10. Tìm nhóm món hàng được mua phổ biến
  11. Dự đoán sở thích của người mua hàng
  12. Lập hóa đơn cho khách hàng

## Ccác loại người dùng

|  |  |  |
| --- | --- | --- |
| **STT** | **Lớp người dùng** | **Mô tả** |
| 1 | Người quản trị | Đối tượng người dùng thực hiện các chức năng nhằm mục đích quản trị website |

## Môi trường hoạt động

* Client : trình duyệt Web ( Chrome, Firefox, … )
* Server : NodeJS phiên bản – 7.0.0

## Các ràng buộc thiết kế và triển khai

* Ngôn ngữ lập trình:
  + Front-end: AngularJS framework, Bootstrap, Jquery, HTML5, CSS3.
  + Back-end: ExpressJS framework
* Cơ sở dữ liệu: MySQL
* Môi trường hoạt động : Internet, Web browser
* IDE : Visual Studio Code, Notepad++, Netbeans,…

## Đặc tả yêu cầu phi chức năng

* Sử dụng hệ quản trị cơ sở dữ liệu MySQL.
* Xứ lý và phản hồi nhanh (dưới 15 giây).
* Dữ liệu phải chính xác, không xuất hiện giá trị *null* hay *NaN.*
* Ứng dụng phải tương thích nhiều hệ điều hành.

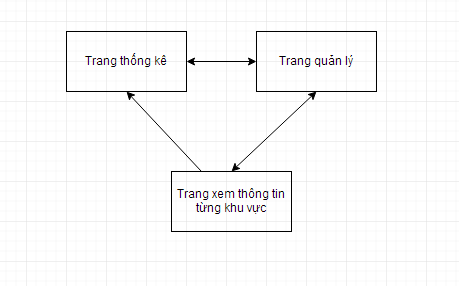
# Thiết kế dữ liệu & giao diện

## Mô hình hoá dữ liệu

# https://scontent.fsgn5-3.fna.fbcdn.net/v/t34.0-12/20643956_1109143162551051_1611952602_n.png?oh=926a845949cbfce7e904b264f469f5cc&oe=5987D5E8

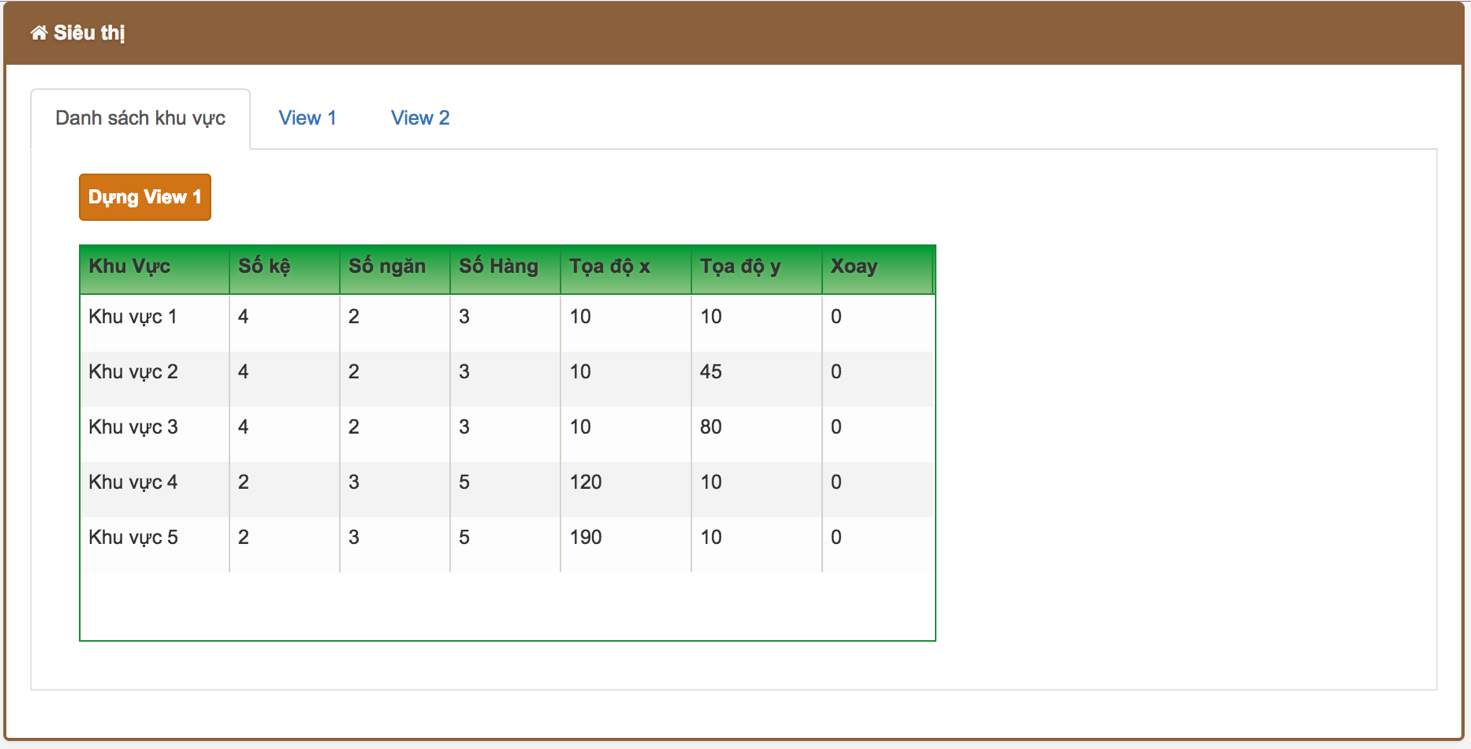
## Thiết kế giao diện

### Sơ đồ và danh sách màn hình



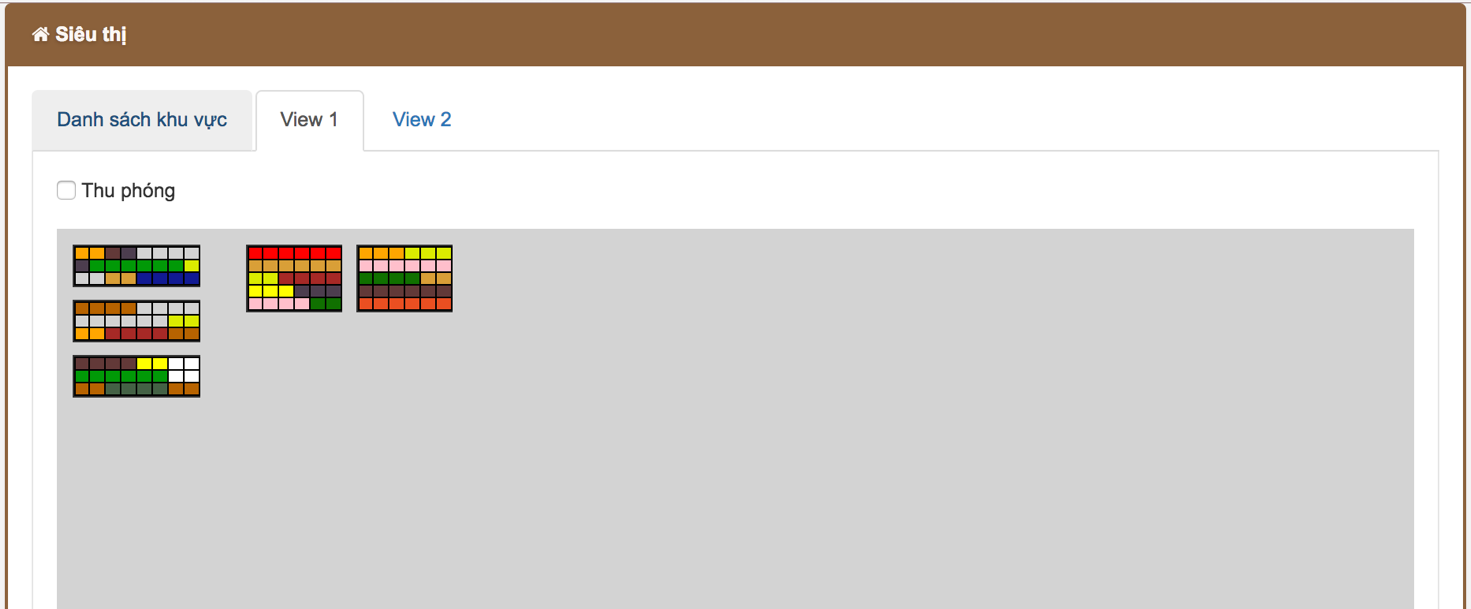
|  |  |  |
| --- | --- | --- |
| **STT** | **Tên màn hình** | **Ý nghĩa/Ghi chú** |
| 1 | Quản lý khu vực | Quản lý các khu vực có trong siêu thị theo thông tin: tên khu vực, số lượng kệ, số lượng ngăn, số lượng hàng và tọa độ. |
| 2 | Sơ đồ quầy kệ | Thực hiện sắp xếp và di chuyển vị trí quầy kệ Khi thực hiện thao tác này, hệ thống sẽ đồng bộ dữ liệu trên tất cả các máy client khác |
| 3 | Chi tiết khu vực | Thực hiện sắp xếp, xóa, cho hàng hóa vào đơn hàng. Khi thực hiện các thao tác này, hệ thống sẽ đồng bộ dữ liệu trên tất cả các máy client khác |
| 4 | Quản lý kiện hàng | Quản lý và tạo mới cũng như chỉnh sửa thông tin kiện hàng có trong siêu thị theo thông tin người dùng nhập vào: tên kiện hàng, giá, màu sắc, kích thước, hạn sử dụng. |
| 5 | Thống kê và Khai thác dữ liệu | Xem thống kê:   * Tổng kiện hàng * Tổng đơn hàng * Số ngày giao dịch   Top 10 sản phẩm bán chạy Thực hiện các bài toán khai thác dữ liệu cơ bản:   * Tìm nhóm món hàng được mua phổ biến. * Tìm luật kết hợp giữa các món hàng. |

### Đặc tả các màn hình giao diện

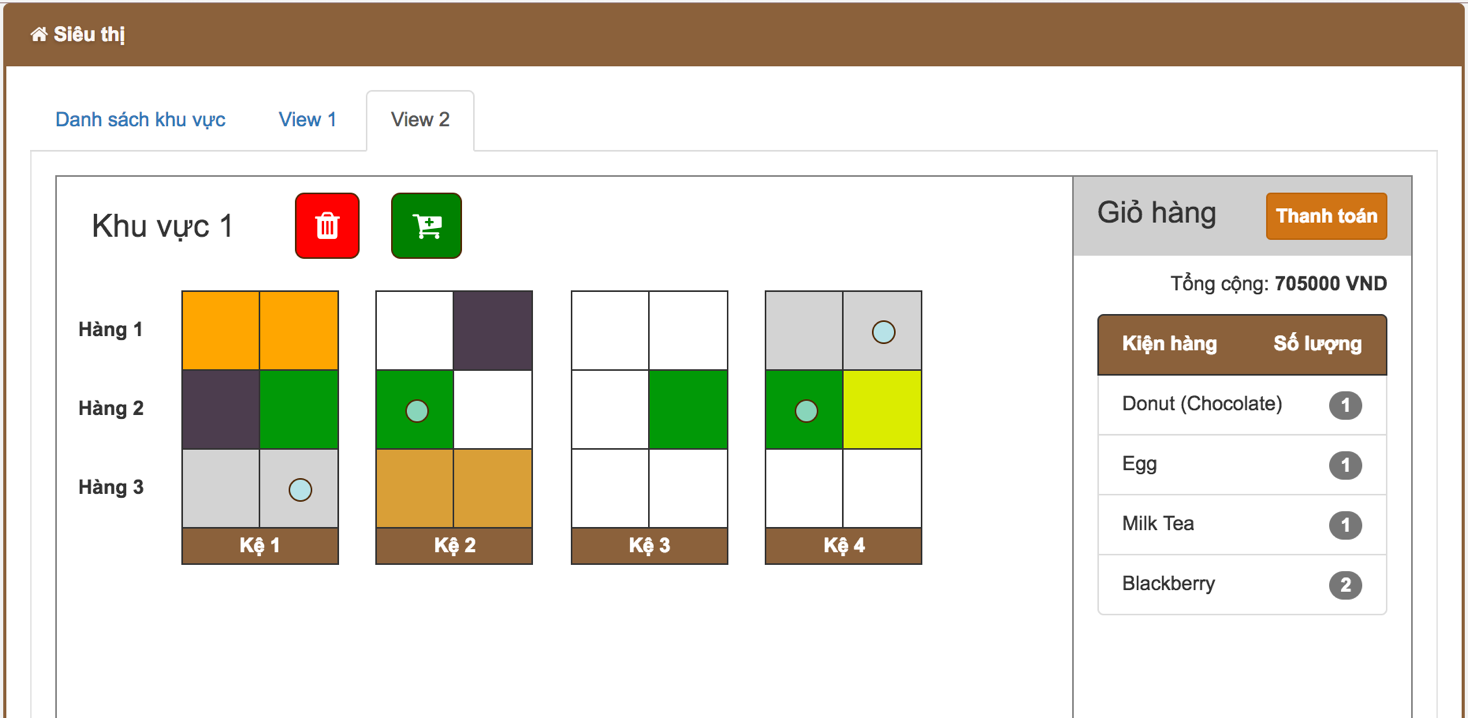


**1.Màn hình quản lý danh sách quầy kệ**

|  |  |
| --- | --- |
| **CLIENT** | **SERVER** |
| **Lấy danh sách Khu vực lên** | |
| **Url data:** api/info – GET method | //tương tác bảng nào, lấy những gì lên |
| **File:** market-ctrl.js  **Func:** getZoneList() |  |
| **Hàm được gọi khi:** - Vừa khởi tạo màn hình quản lí danh sách quầy kệ |  |

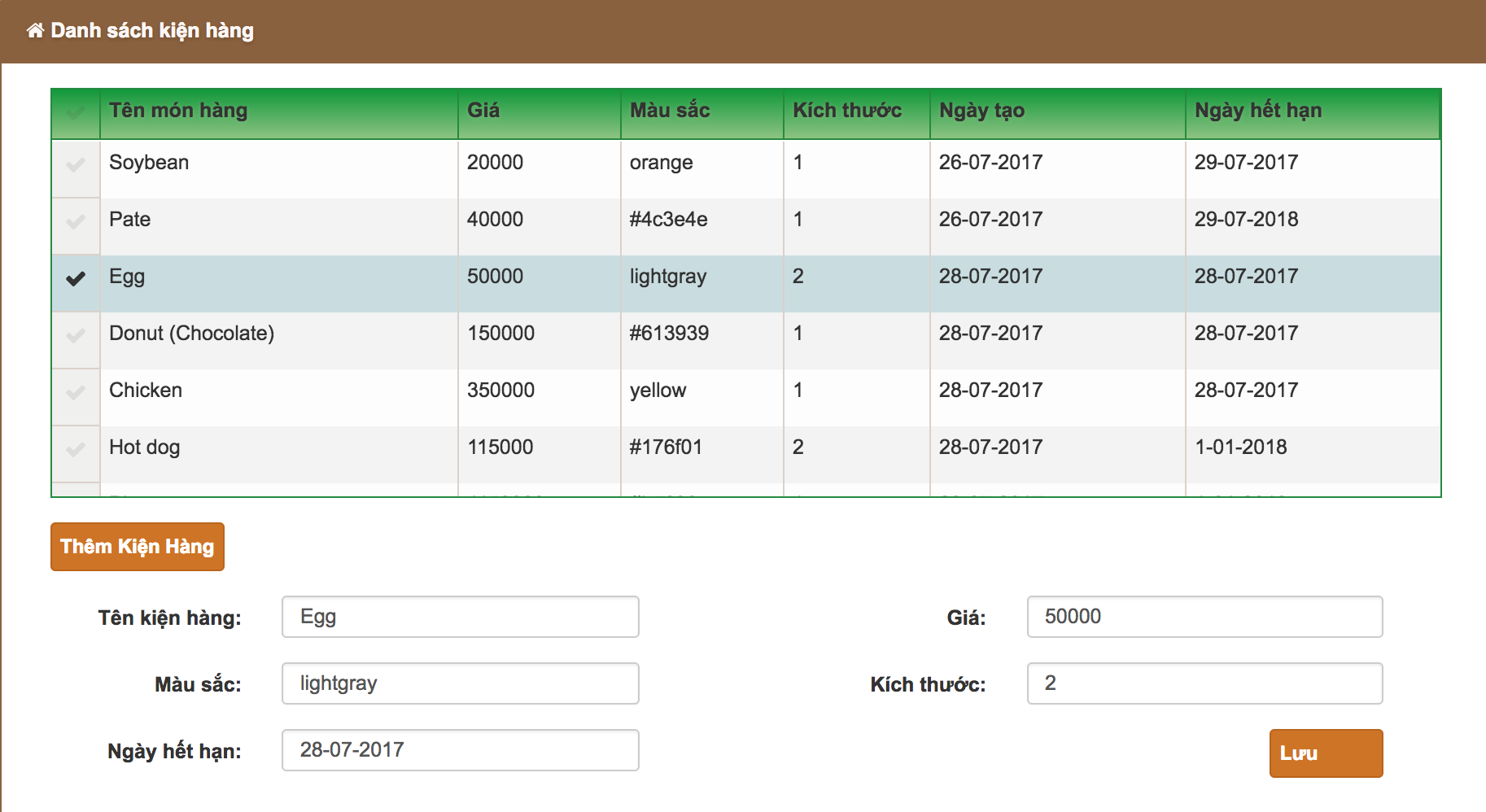
** **2.Màn hình sơ đồ quầy kệ (View 1)**

|  |  |
| --- | --- |
| **CLIENT** | **SERVER** |
| **Dựng màn hình View 1** | |
| **Url data:** dùng bộ dữ liệu của api/info trên |  |
| **File:** market-ctrl.js  **Func:** $scope.renderView1() |  |
| **Hàm được gọi khi:**   * Vừa khởi tạo màn hình quản lí danh sách quầy kệ * Khi nhận được thông báo có sự thay đổi View 1 (di chuyển quầy kệ, di chuyển kiện hàng trong quầy kệ) |  |
| **Kéo dãy kệ, dịch chuyển vị trí dãy kệ** | |
| Bắt sự kiện kéo dãy kệ  $(".khuvuc-container").draggable |  |
| **Socket thay đổi View 1** | |
| Chỉnh sửa vị trí quầy kệ: Gửi 1 thông điệp và bộ dữ liệu cho Server  Thay đổi vị trí của các item trong View 2 cũng làm cho View 1 thay đổi  Công thức: \_socket.emit("change\_zone\_view1", zone); //zone là bộ dữ liệu mới |  |
|  | Nhận thông điệp này và gửi bộ dữ liệu thay đổi cho tất cả các Client khác  Công thức: socket.broadcast.emit('change\_zone\_view1',\_data); |
| Các Client khác (trừ client gửi Socket) sẽ cập nhật lại màn hình View 1 |  |

****

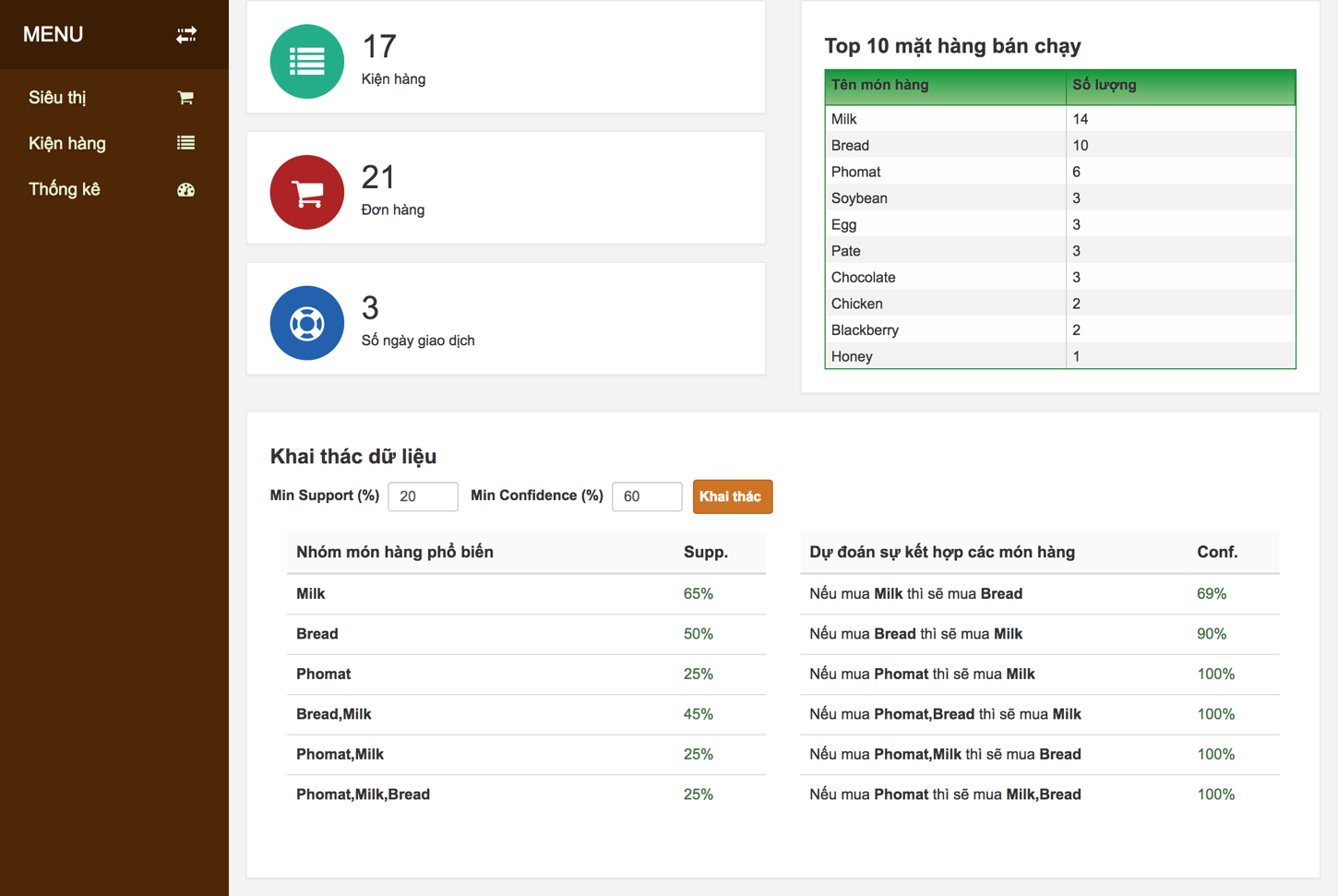
**3.Màn hình chi tiết khu vực (View 2)**

|  |  |  |
| --- | --- | --- |
| **CLIENT** | **SERVER** | |
| **Dựng màn hình View 2** | | |
| **Url data:** dùng bộ dữ liệu của api/info trên |  | |
| **File:** market-ctrl.js  **Func:** $scope.renderView2() |  | |
| **Hàm được gọi khi:**   * Vừa khởi tạo màn hình quản lí danh sách quầy kệ * Khi nhận được thông báo có sự thay đổi View 2 (di chuyển kiện hàng trong quầy kệ) |  | |
| **Thêm vào giỏ hàng** | | |
| **Url data:** Không có | Không có | |
| **File:** market-ctrl.js  **Func:** bắt sự kiện khi Nút giỏ hàng có món hàng kéo đến (Nút màu xanh có hình xe đẩy)  $("#detail\_area #control #addtocart").droppable  //#addtocart là tên của nút Thêm vào giỏ hàng | Không có | |
| **Xóa kiện hàng khỏi kệ** | | |
| **Url data:** Không có | Không có | |
| **File:** market-ctrl.js  **Func:** bắt sự kiện khi Nút xóa kiện hàng có món hàng kéo đến nó (nút màu đỏ hình thùng rác)  $("#detail\_area #control #remove").droppable  //#remove là tên của nút Xóa kiện hàng | Không có | |
| **Đổi vị trí, Xóa, Thêm kiện hàng 🡺 đều làm thay đổi View 2** | | |
| Bắt sự kiện này  $("#detail\_area table.ke td.ngan div.item-head").draggable | |  |
| **Socket thay đổi View 2** | | |
| Thay đổi vị trí của các item trong View 2 cũng làm cho View 1 thay đổi  Công thức: \_socket.emit("change\_item\_view2", request);  //request là 1 json gồm: Action là tên của hành động thay đổi (hiện tại là ‘ChangePos’), Zone: là bộ dữ liệu mới để gửi lên Server |  | |
|  | Nhận thông điệp này và gửi bộ dữ liệu thay đổi cho tất cả các Client khác  Công thức: socket.broadcast.emit('change\_item\_view2',\_data["Zone"]);  //\_data["Zone"] là bộ dữ liệu yêu cầu các View 2 xây dựng theo | |
| Các Client khác (trừ client gửi Socket) sẽ cập nhật lại màn hình View 2 |  | |
| **Thanh toán đơn hàng** | | |
| **Url data:** api/invoice – POST method | //tương tác với bảng nào, lưu vào đâu | |
| **File:** market-ctrl.js  **Func:** $scope.entryBill() |  | |
| **Hàm được gọi khi:**   * Click nút Thanh toán trên màn hình View 2   **Sau khi gọi hàm:**   * Dữ liệu được lưu vào DB, đây là bộ dữ liệu mẫu dùng để dùng chức năng khai thác dữ liệu |  | |

****

**4.Màn hình quản lí kiện hàng**

|  |  |
| --- | --- |
| **CLIENT** | **SERVER** |
| **Lấy danh sách Kiện hàng** | |
| **Url data:** api/item – GET method | //tương tác bảng nào, lấy những gì lên |
| **File:** item-ctrl.js  **Func:** getItemList() |  |
| **Hàm được gọi khi:** - Vừa khởi tạo màn hình Danh sách kiện hàng |  |
| **Hàm Thêm mới kiện hàng**  ***Nhập tên, màu sắc, ngày hết hạn, giá, kích thước của nó trên bản đồ***  ***Lưu ý: Bấm nút “Thêm Kiện Hàng” để làm trống các giá trị rồi mới thêm được*** | |
| **Url data:** api/item – POST method | //tương tác bảng nào, lấy những gì lên |
| **File:** item-ctrl.js  **Func:** save() //lưu với ý nghĩa là thêm mới |  |
| **Hàm được gọi khi:** - Khi bấm nút Lưu |  |
| **Hàm Update kiện hàng**  ***Lưu ý: Chọn dữ liệu trên Danh sách*, *dữ liệu sẽ hiện lên các field vào cho người dùng chỉnh sửa thông tin (đổi tên, màu sắc, ngày hết hạn, giá, kích thước của nó trên bản đồ)*** | |
| **Url data:** api/item – PUT method | //tương tác bảng nào, lấy những gì lên |
| **File:** item-ctrl.js  **Func:** save() //lưu với ý nghĩa là cập nhật |  |
| **Hàm được gọi khi:** - Khi bấm nút Lưu |  |

****

**5.Màn hình Thống kê và Khai thác dữ liệu**

|  |  |
| --- | --- |
| **CLIENT** | **SERVER** |
| **Lấy thống kê dữ liệu cơ bản**  ***Dữ liệu trả về là Số kiện hàng, Số đơn hàng, Số ngày giao dịch*** | |
| **Url data:** api/statistic – GET method | //tương tác bảng nào, lấy những gì lên |
| **File:** statistic.js  **Func:** getStatistic() |  |
| **Hàm được gọi khi:** - Vừa khởi tạo màn hình Thống kê |  |
| **Lấy danh sách kiện hàng bán chạy**  ***Dữ liệu trả về là danh sách top 10 sản phẩm bán chạy*** | |
| **Url data:** api/item/bestseller – GET method | //tương tác bảng nào, lấy những gì lên |
| **File:** statistic.js  **Func:** getBestseller() |  |
| **Hàm được gọi khi:** - Vừa khởi tạo màn hình Thống kê |  |
| **Lấy Mẫu phổ biến và Luật kết hợp** | |
| **Url data:** api/statistic/datamining – GET method  **Param:** Minsupp và MinConf | //tương tác bảng nào, lấy những gì lên |
| **File:** statistic.js  **Func:** getDatamining() | //chạy giải thuật ? |
| **Hàm được gọi khi:**   * Vừa khởi tạo màn hình Thống kê * Giá trị mặc định ban đầu minSupp=20 và minConf=40 |  |

**KẾT THÚC BÁO CÁO**