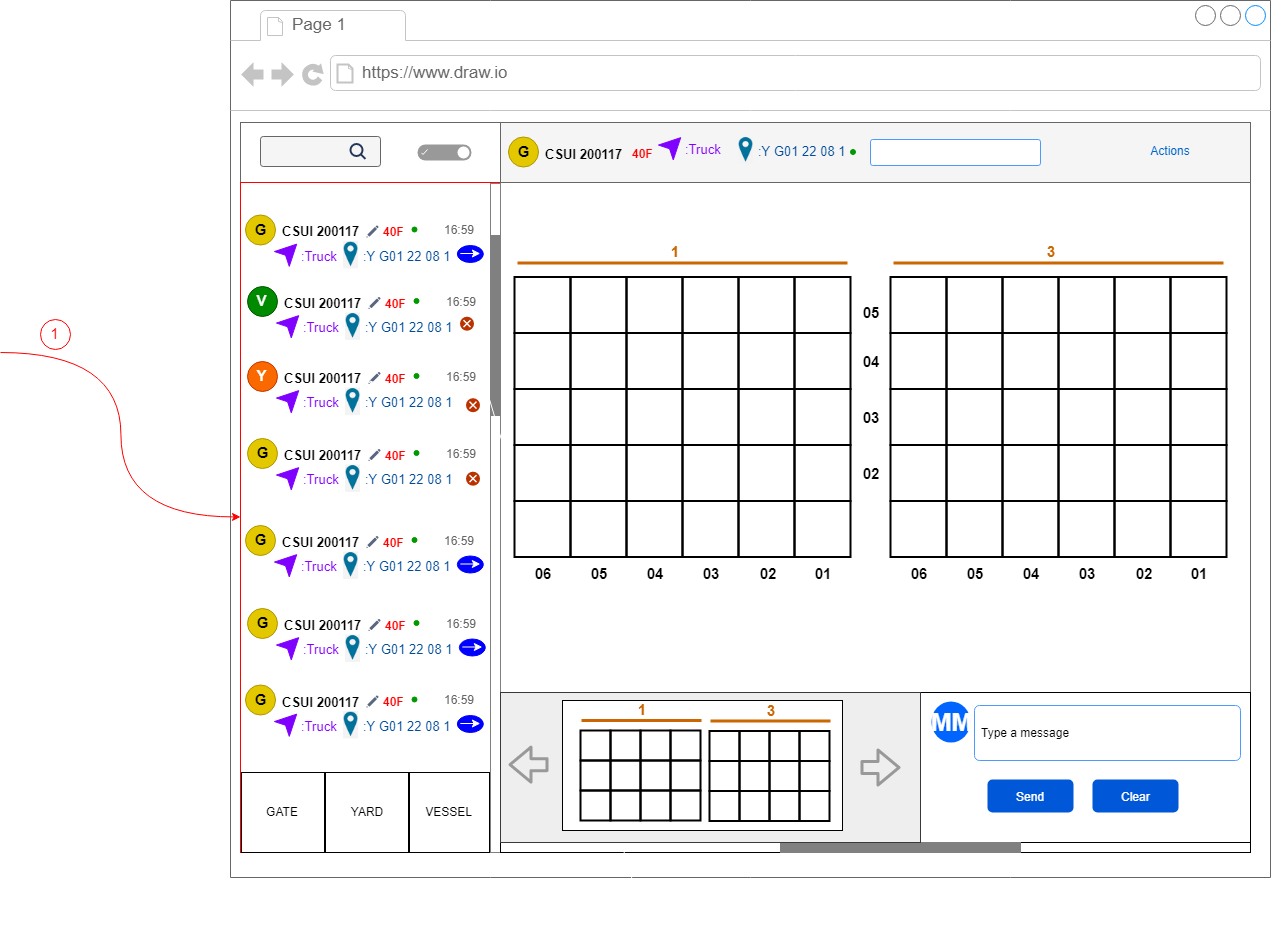
ĐẶC TẢ MÀN HÌNH HANDHELD (VERSION 1)

# Màn hình Danh sách container làm hàng:

## Mockup:



Hình : Đặc tả màn hình danh sách container làm hàng

## Đặc tả:

* Trên Gridview Y10-Truck Gate WQ-Receival bao gồm các columns:

### Component input:

### Component output:

## Flowchart & Diagrams

**Bao gồm 03 luồng chính:**

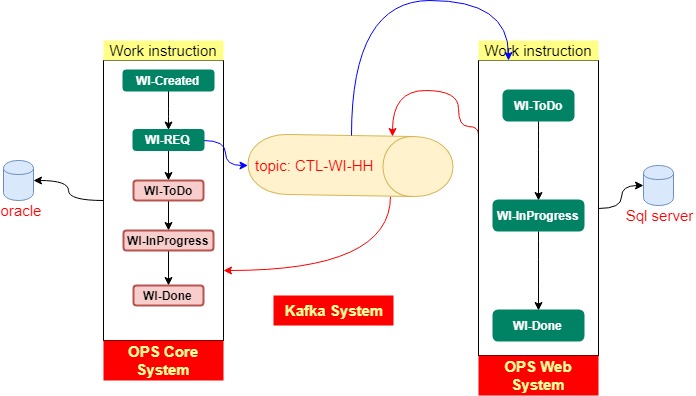
* Luồng hiển thị danh sách các container cần làm hàng (Nội bộ OPS Web)
* Luồng cập nhật n container cần làm hàng (OPS Web – OPS Core)
* Luồng xóa n container khỏi hàng đợi cần làm hàng (OPS Web – OPS Core)
* Luồng cập nhật vị trí container thực tế trên bãi (OPS Web – OPS Core)

### **Luồng hiển thị danh sách các container cần làm hàng (Nội bộ OPS Web)/ Luồng cập nhật n container cần làm hàng (OPS Web – OPS Core)**

Mỗi 1 lệnh làm hàng (mỗi đối tượng Work Instuction – WI) sẽ trải qua các trạng thái:

* Yêu Cầu Làm Hàng (**WI-REQ**) → nhận yêu cầu làm hàng ( **WI-ToDo**) → Đang làm hàng (**WI-InProgress**) → làm hàng thành công (**WI-Done**).
* Trạng thái Yêu Cầu Làm Hàng (WI­-REQ): Khi OPS Core cho phép 1 container nào đó vào bãi chuẩn bị làm hàng thì sẽ cập nhật trạng thái cho container đó là **WI-REQ** (Work instruction request). OPS Core sẽ gửi yêu cầu làm hàng này đến hệ thống Kafka (topic: <SiteId>-WI.)
* Nếu hệ thống OPS Web nhận được, một mặt sẽ cập nhật vào hệ thống của mình. Mặt khác, sẽ gửi thông báo đến OPS Core báo rằng OPS Web đã nhận được. Lúc này OPS Core sẽ cập nhật container đó ở trạng thái **WI-ToDo** (Work instruction todo).
* Khi OPS Web gửi được lệnh làm hàng đó lên thiết bị hanhheld thì nó sẽ cập nhật và gửi lại trạng thái cho OPS Core là **WI-InProgress**
* Khi OPS Web trên thiết bị handheld đã hoàn thành việc làm hàng sẽ gửi lại vị trí và cập nhật lại trạng thái **WI-Done** cho cả hai hệ thống OPS Web + OPS Core.

*Sơ đồ trạng thái sẽ như sau:*



Hình : Sơ đồ trạng thái của WI

Đối tượng giao tiếp sẽ theo format sau:

{

"Sender":"<sender>",

"Object":"<object\_name>",

"Data": "<data>",

"links": [

{

"href": "<href>",

"rel": "<rel>",

"type": "<method>"

}

]

}

* *<sender>*: Tên Server gửi message.
* *<object\_name>*: Tên đối tượng được quy ước trước.
* *<data>*: Dữ liệu được truền qua kafka theo kiểu dynamic/ hoặc quy ước giữa các bên.
* *<href>*: Link api
* *<rel>*: Key work api
* *<method>*: Phương thức gọi: POST, GET, PUT, DELETE

Ứng với mỗi trạng thái, đối tượng giao tiếp sẽ có nội dung như sau:

**WI-REQ**: OPS Core → OPS Web :

{

"Sender":"OPSCore"

"Object":"WI",

"Data": [{

"WIId":<wi\_id>,

"Status":<status>,

},…],

"links": [

{

"href": "<domain-ops-core>:<port>/api/v2/{siteId}/req-wis",

"rel": "req-wi",

"type": "GET"

}

]

}

**WI-ToDo;** **WI-InProgress;**

{

"Sender":"OPSWeb"

"Object":"WI",

"Data": [{

"WIId":<wi\_id>,

"Status":<status>,

},…],

"links": [

{

"href": "<domain-ops-web>:<port>/api/v2/{siteId}/res-wis/{fromDate}/{toDate}",

"rel": "res-wi",

"type": "GET"

}

]

}

**WI-Done**:

{

"Sender":"OPSWeb"

"Object":"WI",

"Data": [{

"WIId":<wi\_id>,

"Status":<status>,

"UserUpdated":<user\_updated>,

"DateUpdated":<date\_updated>,

"CheId":<che\_id>,

"Location": {

"BlockId":<blockId>,

"Bay":<bay>,

"Row":<row>,

"Tier":<tier>,

},

},…],

"links": [

{

"href": "<domain-ops-web>:<port>/api/v2/{siteId}/done-wis",

"rel": "done-wi",

"type": "GET"

}

]

}

1. GET api/v2/{siteId}/che/{cheId}/workinstructions

* Desc: Lấy danh sách các container cần làm hàng theo phương tiện này. API này được cài đặt phía OPS Web
* Input:

|  |  |
| --- | --- |
| **Paramatter** | **Desc** |
| siteId | * Mã site, Vd: CTL, HP,... |
| cheId | * Mã phương tiện. |

* Output:

|  |  |
| --- | --- |
| **Data** | **Desc** |
| [{  "WIId": "<wi\_id>",  "WIIdRef": "<wi\_id\_ref>",  "IsRemove": "<is\_remove>",  "CurrentLocName": "<current\_loc\_name>",  "DateCreated": "<date-created>",  "ContainerInfor": {  "Fel": "<fel>",  "ISO":"<iso>",  "ItemNo":"<item\_no>",  "ArrCar":"<arr\_car>",  "GrWt":"<grwt>",  "Tare":"<tare>",  "DepCar":"<dep\_car>",  "Cat":"<cat>",  "Commod":"<commod>",  "OrgPort":"<org\_port>",  "LOP":"<lop>",  "LP":"<lp>",  "SealNo":"<seal\_no>",  "DP":"<dp>",  "SpecHdlCd":"<spec\_hdl\_cd>",  "LLDP":"<lldp>",  "AvailTime":"<avail\_time>",  "FDP":"<fdp>",  "ArrivalTime":"<arrival\_time>",  "Temp":"<temp>",  "EstMvTime":"<est\_mv\_time>",  "Grade":"<grade>",  "EstMvTime2":"<est\_mv\_time2>",  "LoadInstr":"<load\_instr>",  "CurrentLoc":"<current\_loc>",  "PlanLoc":"<plan\_loc>",  "HistoricalLoc":"<historical\_loc>",  }  },…] | <wi\_id>: Work instruction Id: Mã làm hàng.  <wi\_id\_ref>: Work instruction Id references: Mã làm hàng của hệ thống OPS Core.  <is\_remove>: trạng thái có bị remove hay không?  <current\_loc\_name>: Vessel/ Yard/ Truck  <status>: trạng thái WI: **WI-Created → WI-REQ → WI-ToDo → WI-InProgress → WI-Done**.  <fel>: F: full/ E: empty.  <iso>: chuẩn container (loại 20/40 feet).  <item\_no>: Số container.  <current\_loc>: Vị trí hiện tại của container.  <plan\_loc>: Vị trí dự kiến sẽ đặt container trên bãi. |

Nhóm API hỗ trợ cho hoạt động đồng bô dữ liệu thông qua Kafka.

1. GET api/v2/{siteId}/req-wis

* Desc: Lấy danh sách các WI cần làm hàng cho tới thời điểm hiện tại (lấy danh sách các WI ở trạng thái **WI-REQ**). API này được cài đặt phía OPS Core.
* Input:

|  |  |
| --- | --- |
| **Paramatter** | **Desc** |
| siteId | * Mã site, Vd: CTL, HP,... |

* Output:

|  |  |
| --- | --- |
| **Data** | **Desc** |
| [{  "WIId": "<wi\_id>",  "Status": "<status>",  "DateUpdated": "<date\_updated>",  "UserUpdated": "<user\_updated>",  "CHEIds": [  "<che\_id1>",  "<che\_id2>",  ….  ],  "ContainerInfor": {  "Fel": "<fel>",  "ISO":"<iso>",  "ItemNo":"<item\_no>",  "ArrCar":"<arr\_car>",  "GrWt":"<grwt>",  "Tare":"<tare>",  "DepCar":"<dep\_car>",  "Cat":"<cat>",  "Commod":"<commod>",  "OrgPort":"<org\_port>",  "LOP":"<lop>",  "LP":"<lp>",  "SealNo":"<seal\_no>",  "DP":"<dp>",  "SpecHdlCd":"<spec\_hdl\_cd>",  "LLDP":"<lldp>",  "AvailTime":"<avail\_time>",  "FDP":"<fdp>",  "ArrivalTime":"<arrival\_time>",  "Temp":"<temp>",  "EstMvTime":"<est\_mv\_time>",  "Grade":"<grade>",  "EstMvTime2":"<est\_mv\_time2>",  "LoadInstr":"<load\_instr>",  "CurrentLoc": {  "LocName":"<loc\_name>",  "BlockId":"<block\_id>",  "Bay":"<bay>",  "Row":"<row>",  "Tier":"<tier>",  },  "PlanLoc": {  "LocName":"<loc\_name>",  "BlockId":"<block\_id>",  "Bay":"<bay>",  "Row":"<row>",  "Tier":"<tier>",  },  }  },…] | <wi\_id>: Work instruction Id: Mã làm hàng.  <status>: trạng thái WI: **WI-Created → WI-REQ → WI-ToDo → WI-InProgress → WI-Done**.  <online>: True: online/False: offline (optional)  <fel>: F: full/ E: empty.  <iso>: chuẩn container (loại 20/40 feet).  <item\_no>: Số container.  <current\_loc>: Vị trí hiện tại của container.  <plan\_loc>: Vị trí dự kiến sẽ đặt container trên bãi. |

1. GET api/v2/{siteId}/res-wis/{fromDate}/{toDate}

* Desc: Lấy danh sách các WI có trạng thái thay đổi trong khoản thời gian từ . API này được cài đặt phía OPS Web.
* Input:

|  |  |
| --- | --- |
| **Paramatter** | **Desc** |
| siteId | * Mã site, Vd: CTL, HP,... |
| fromDate | * Thời gian bắt đầu, tuân theo format dd/MM/yyyy hh:mm:ss |
| toDate | * Thời gian kết thúc, tuân theo format dd/MM/yyyy hh:mm:ss |

* Output:

|  |  |
| --- | --- |
| **Data** | **Desc** |
| [{  "WIId": "<wi\_id>",  "Status": "<status>",  "DateUpdated": "<date\_updated>",  "UserUpdated": "<user\_updated>",  },…] | <wi\_id>: Work instruction Id: Mã làm hàng.  <status>: trạng thái WI: **WI-Created → WI-REQ → WI-ToDo → WI-InProgress → WI-Done**  <date\_updated>: Thời gian OPS Web cập nhật trạng thái. |

1. GET api/v2/{siteId}/done-wis/{fromDate}/{toDate}

* Desc: Lấy danh sách các WI ở trạng thái **WI-Done** trong khoản thời gian từ . API này được cài đặt phía OPS Web.
* Input:

|  |  |
| --- | --- |
| **Paramatter** | **Desc** |
| siteId | * Mã site, Vd: CTL, HP,... |
| cheId | * Mã phương tiện. |

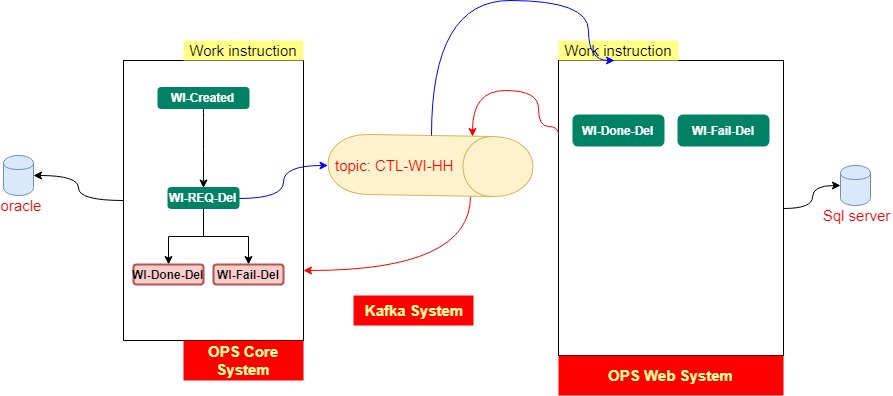
* Output:

|  |  |
| --- | --- |
| **Data** | **Desc** |
| [{  "WIId": "<wi\_id>",  "Status": "<status>",  "DateUpdated": "<date\_updated>",  "UserUpdated": "<user\_updated>",  "CHEId": "<che\_id>",  "Location": {  "BlockId": "<block\_id>",  "Bay": "<bay>",  "Row": "<row>",  "Tier": "<tier>",  }  },…] | <wi\_id>: Work instruction Id: Mã làm hàng.  <status>: trạng thái WI: **WI-Created → WI-REQ → WI-ToDo → WI-InProgress → WI-Done** |

### **Luồng xóa n container khỏi hàng đợi cần làm hàng (OPS Web – OPS Core)**

* Việc xóa 1 lệnh làm hàng nào đó (Hoặc thay đổi nội dung) sẽ do OPS Core quyết định.
* Việc xóa nội dung WI sẽ phụ thuộc vào trạng thái của lệnh làm hàng đó tại OPS Core. Trạng thái Lệnh làm hàng được xếp theo thứ tự: (1) **WI-Created →** (2)**WI-REQ →** (3)**WI-ToDo →** (4)**WI-InProgress →** (5)**WI-Done**. Nếu lệnh làm hàng ở Trạng thái (5)**WI-Done** sẽ không được Xóa. Nếu muốn thay đổi vị trí container đó trên bãi thì phải tạo 1 WI mới. Ngược các trạng thái còn lại, OPS Core có quyền yêu cầu: Xóa nội dung lệnh làm hàng.

*Sơ đồ trạng thái cho luồng xóa sẽ như sau:*



Hình : Sơ đồ trạng thái xóa

P/s: Ta có thể đặt lại tên các trạng thái liên quan đến Delete cho hay hơn, nhưng để mang tính nhất quáng trong xuyên suốt các hệ thống ta nên dùng lại các từ như Created, Req, ToDo, InProgress, Done.

Các trạng thái liên quan đến Xóa, đối tượng giao tiếp sẽ có nội dung như sau:

**WI-REQ-Del**:

{

"Sender":"OPSCore"

"Object":"WI",

"Data": [{

"WIId":<wi\_id>,

"Status":<status>,

},…],

"links": [

{

"href": "<domain-ops-core>:<port>/api/v2/{siteId}/del-req-wis",

"rel": "del-req-wi",

"type": "GET"

}

]

}

**WI-Done-Del**:

{

"Sender":"OPSWeb"

"Object":"WI",

"Data": [{

"WIId":<wi\_id>,

"Status":<status>,

},…],

"links": [

{

"href": "<domain-ops-web>:<port>/api/v2/{siteId}/del-res-wis/{fromData}/{toDate}",

"rel": "del-res-wi",

"type": "GET"

}

]

}

1. GET api/v2/{siteId}/del-req-wis

* Desc: Lấy danh sách các WI cần xóa khỏi hàng đợi làm hàng (Lấy danh sách các WI ở trạng thái **WI-REQ-Del**. API được cài đặt phía OPS Core.
* Input:

|  |  |
| --- | --- |
| **Paramatter** | **Desc** |
| siteId | * Mã site, Vd: CTL, HP,... |

* Output:

|  |  |
| --- | --- |
| **Data** | **Desc** |
| [{  "WIId": "<wi\_id>",  "Status": "<status>",  "DateUpdated": "<date\_updated>",  "UserUpdated": "<user\_updated>",  },…] |  |

1. GET api/v2/{siteId}/del-res-wis/{fromData}/{toDate}

* Desc: Lấy danh sách các WI ở trạng thái **WI-Done-Del**; **WI-Fail-Del**. API được cài đặt phía OPS Web.
* Input:

|  |  |
| --- | --- |
| **Paramatter** | **Desc** |
| siteId | * Mã site, Vd: CTL, HP,... |
| fromDate | * Thời gian bắt đầu, tuân theo format dd/MM/yyyy hh:mm:ss |
| toDate | * Thời gian kết thúc, tuân theo format dd/MM/yyyy hh:mm:ss |

* Output:

|  |  |
| --- | --- |
| **Data** | **Desc** |
| [{  "WIId": "<wi\_id>",  "Status": "<status>",  "DateUpdated": "<date\_updated>"  },…] |  |

### **Luồng cập nhật vị trí container thực tế trên bãi (OPS Web – OPS Core)**

1. PUT api/v2/{siteId}/che/{cheId}/wis/{wiid}

* Desc: Cập nhật vị trí thực tế ngoài bãi của container này vào hệ thống. API được cài đặt phia OPS Web.
* Input:

|  |  |
| --- | --- |
| **Paramatter** | **Desc** |
| siteId | * Mã site, Vd: CTL, HP,... |
| cheId | * Mã CHE |
| wiid | * Mã WI |

* Output:

|  |  |
| --- | --- |
| **Data** | **Desc** |
| {  } |  |