# 系统分析

## 系统调查

配载是把要装船的集装箱按船舶的运输要求以及码头的作业要求而制定的具体装载计划，最终的呈现形式是把已经在堆场的每一个待装集装箱（那些直装的集装箱则是直接运到岸桥下，直接装船，但是也要做配载计划）计划到具体的船箱位。配载必须满足船舶的运输要求，即船舶的船行安全、集装箱及其货物的运输质量船舶营运的经济效益，同时也要兼顾集装箱码头装卸工艺和操作方式，使码头能合理、有效地组织生产。

船舶配载一般有预配船图，预配船图是由船公司提供，作为集装箱码头实配的依据，该信息主要包括：船上每一具体船箱位将要装箱的规格信息（尺寸、箱重、卸货港、箱型---该信息主要包括是GP箱还是其他例如框架箱之类的信息），一般情况下，只要符合这些箱重的要求，那么船舶的重心、航行的要求都能得到满足。

## 船舶配载计划的制订规则

集装箱船舶预配是配载员最重要的配载依据，其制定过程需要满足预配中定义的放箱约束。预配限定的是某一类箱子应该放那些区域；实配限定的是某个箱子应该放在哪个位置上。如下图所示，是某个贝位的预配图。在预配图中定义了船上某个堆存位可以装载哪个卸货港的箱子，在此基础上还可以设定所需要的箱子的箱型、尺寸、是否超箱、是否是危险品、是否可放45ft等特殊属性。

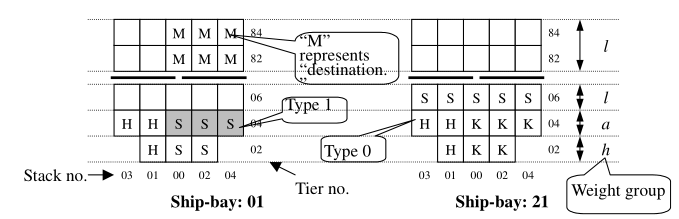


图1.船舶预配图（某一贝位）

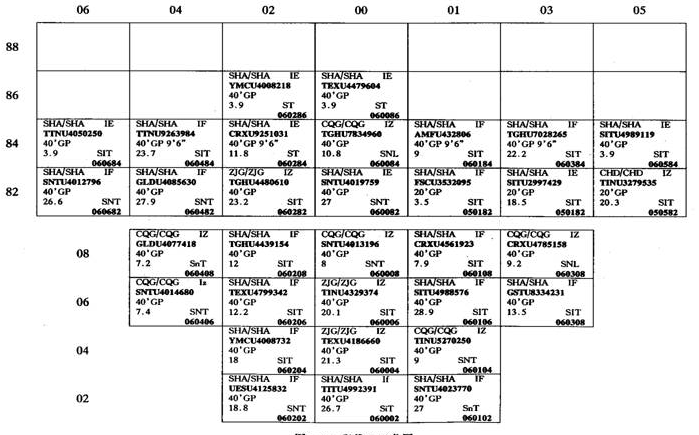


图2实配图（某一贝位）

配载的一般制作过程：配载制作过程一般分为五步进行，分别是分类、索箱、选倍、划块、配箱。1）分类：是指将对该航线出口箱按照箱子的卸货港、尺寸、箱型、箱重等属性来划分。2）索箱：是指配载员选择某类箱子，然后检索出该类箱子在场地里面的堆存位置，并且要查看和分析箱子分布在几个箱区以及在箱区的集中程度等场地状态信息，从而辅助配载员构思配载方案。3）选倍：选倍是指为刚刚索箱的这组集装箱选择想要配载的船舶倍位。4）划块：划块是指在已选倍内划出一个目标配载区域。5）配箱：即要场地内的各个区位上的某些箱子按照一定的配载顺序配载到刚刚划块作业所圈定的船箱位上。

配载遵循的原则：第一、满足预配船图的要求；第二、满足后续装船作业的有序开展（对于自动化集装箱码头来说，要满足这个要求，一般要包括：堆场的装船作业和其他能尽可能分开，堆场发箱顺序能够和后续的岸吊计划能匹配，堆场机械的作业能力能满足后续装船的需要）。

## 组织结构与功能分析



装卸调度部门：负责集装箱装卸过程中的人工作业，如集装箱在运输至岸桥时，工作人员进行拔插箱锁作业

维修部门：负责日常岸桥设备的维修工作

任务调度部门：负责集装箱装卸作业的任务调度工作，决定堆场的哪个箱子需要作业（具体到某一个贝位下的某一层中对应的某一列）、何时作业、运输到船舶上的哪个箱位（具体到船舶某个贝位下的某层中对应的某一列）。根据当前船舶集装箱分布，以及实际业务需求，决定是否要进行倒箱操作。

## 业务流程分析





## 数据与数据流程分析

### 数据汇总

数据类 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 同步 数据字典部分

### 数据分析

U/C 矩阵 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 数据流程分析

装船

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* visio

卸船

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*visio

## 功能/数据分析

U/C 矩阵进一步分析

# 系统设计

## 系统总体结构设计

### 子系统划分

* + 1. 网络设计 \*\*\*\*\*\*\*\*
    2. 设备和网络的配置 \*\*\*\*\*\*\*\*

## 代码设计

\*/\*\*\*\*\*\*\*\*\*\* 见文档 copy

谈 各种编码的设计 如 00线 贝位结构定义 等数据库部分的定义

## 数据结构和数据库设计

ER图

各个数据库表格

主键 外键 约束

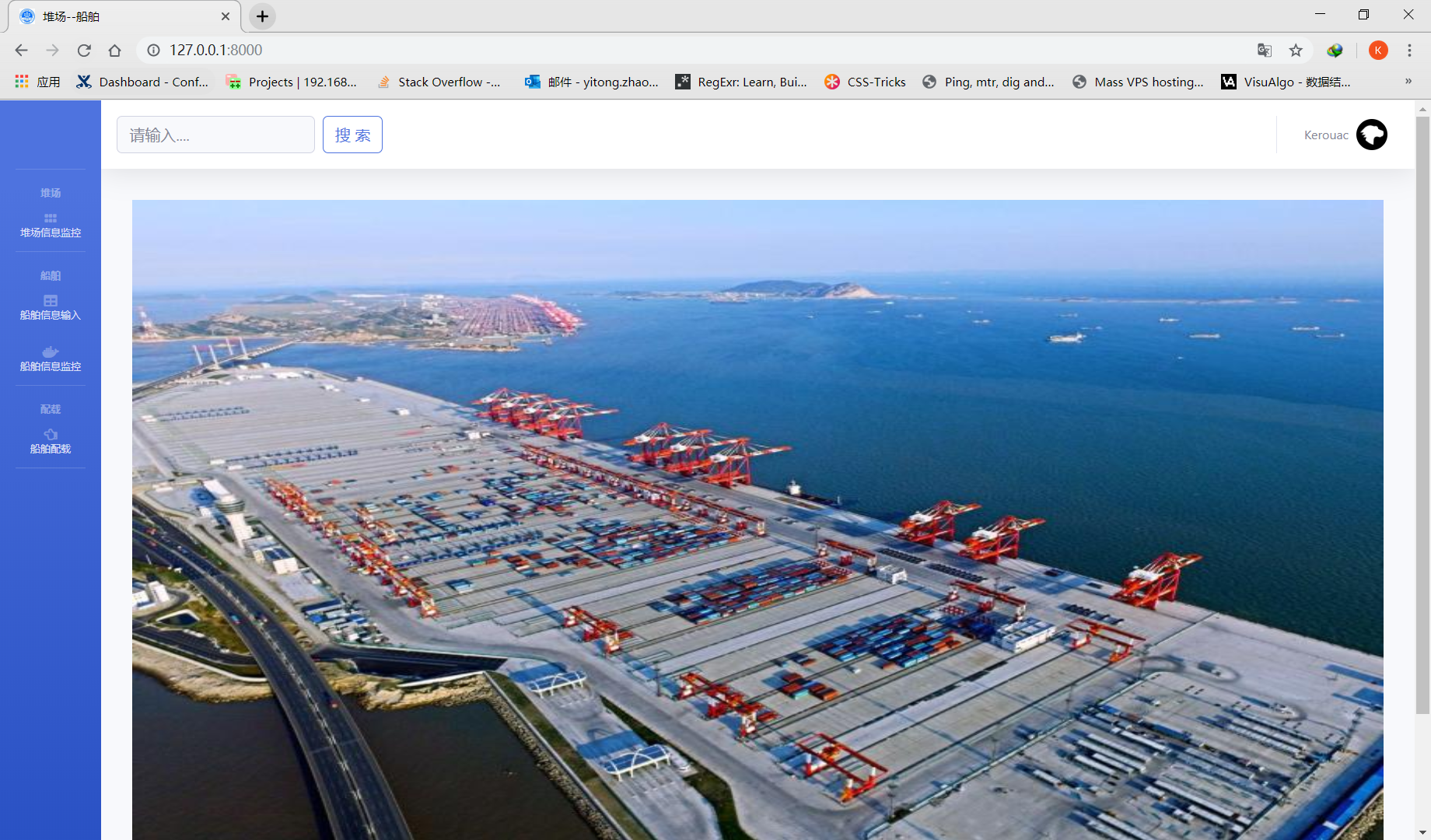
## 输入输出设计

### 用户界面设计：

#### 路由设置

访问本地URL，端口号为 8000

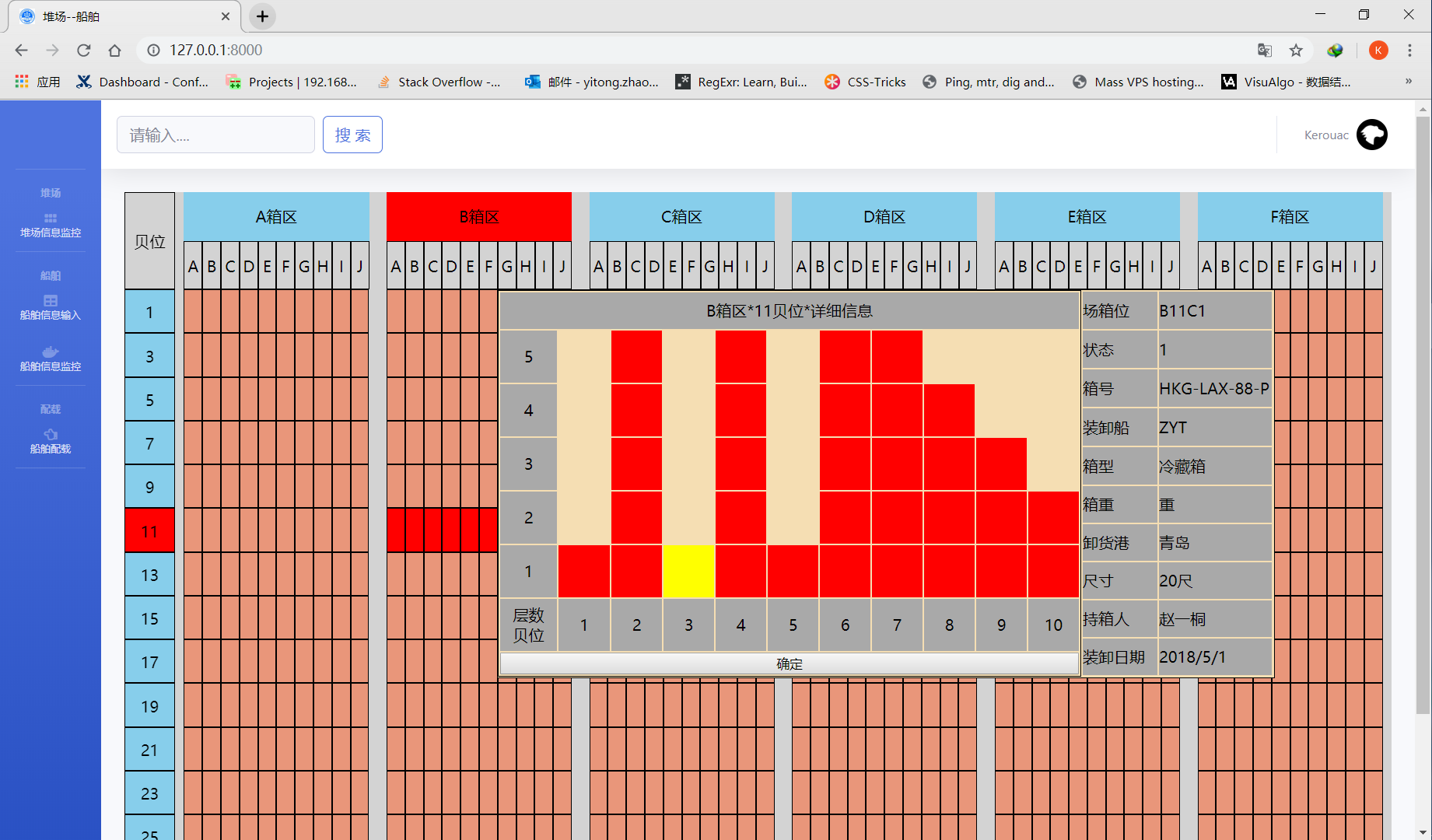
127.0.0.1:8000



#### 功能展示

##### 堆场监控

左侧导航栏点击堆场信息监控，选择视图中的堆场监控，即加载出堆场监控页面

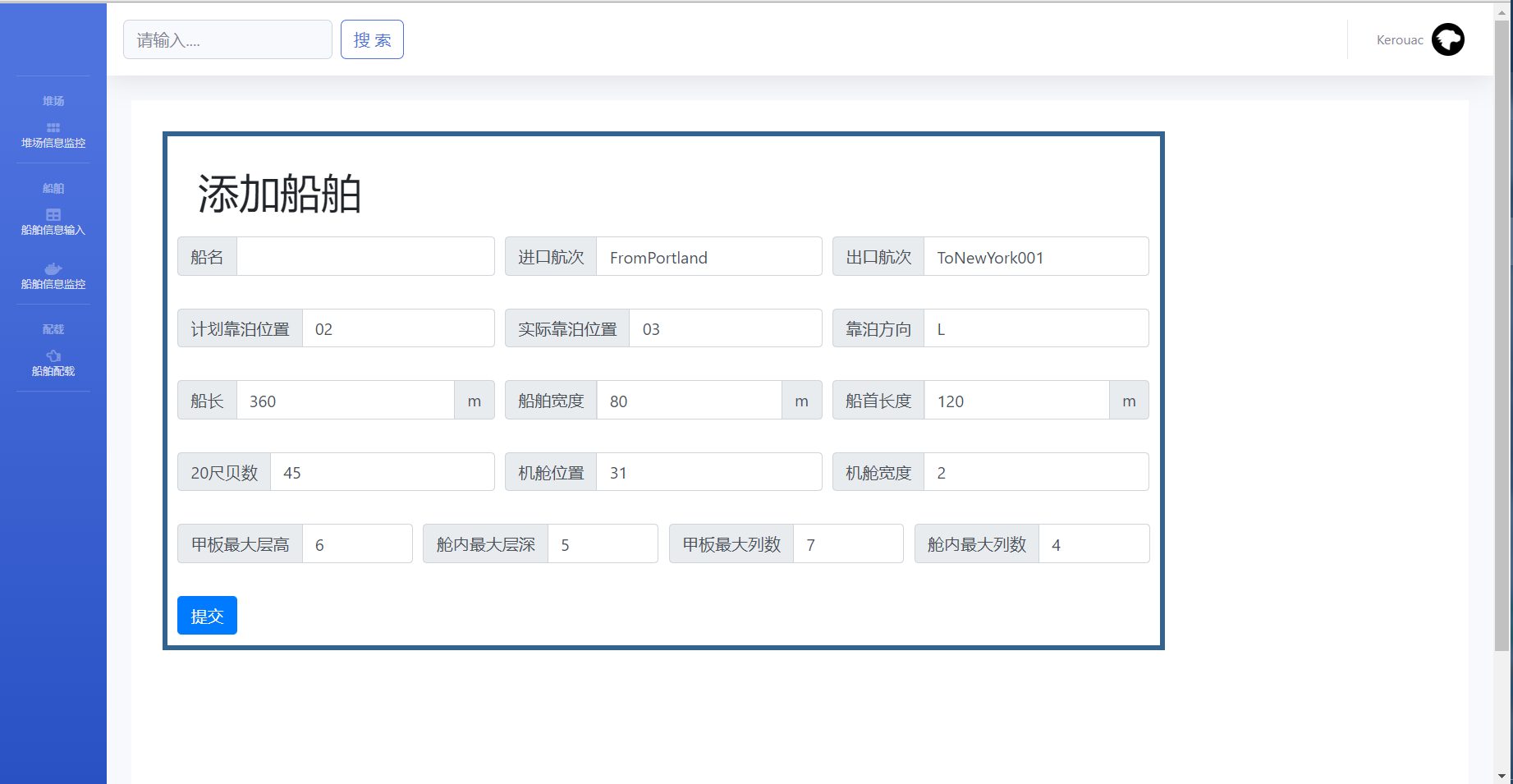


如图：鼠标通过点击不同的箱区中的不同贝位，弹框中会显示该箱区该贝位下的集装箱整体堆放信息，图中红色部分即为集装箱，鼠标悬浮在不同箱位，可以显示不同箱位下的信息，包括场箱子位，箱号，持箱人，装卸日期等信息。

###### 定义船舶结构

定义整体船舶

选择导航栏中的船舶信息输入，在表单输入中选择增加船舶项，具体页面展示如下



相关数据包括靠泊方向，20尺贝数，机舱位置，机舱宽度（占据贝位数），甲板最大层高，舱内最大层深，甲板最大列数，舱内最大列数，输入完成后，点击提交按钮即可增加一条新船舶。

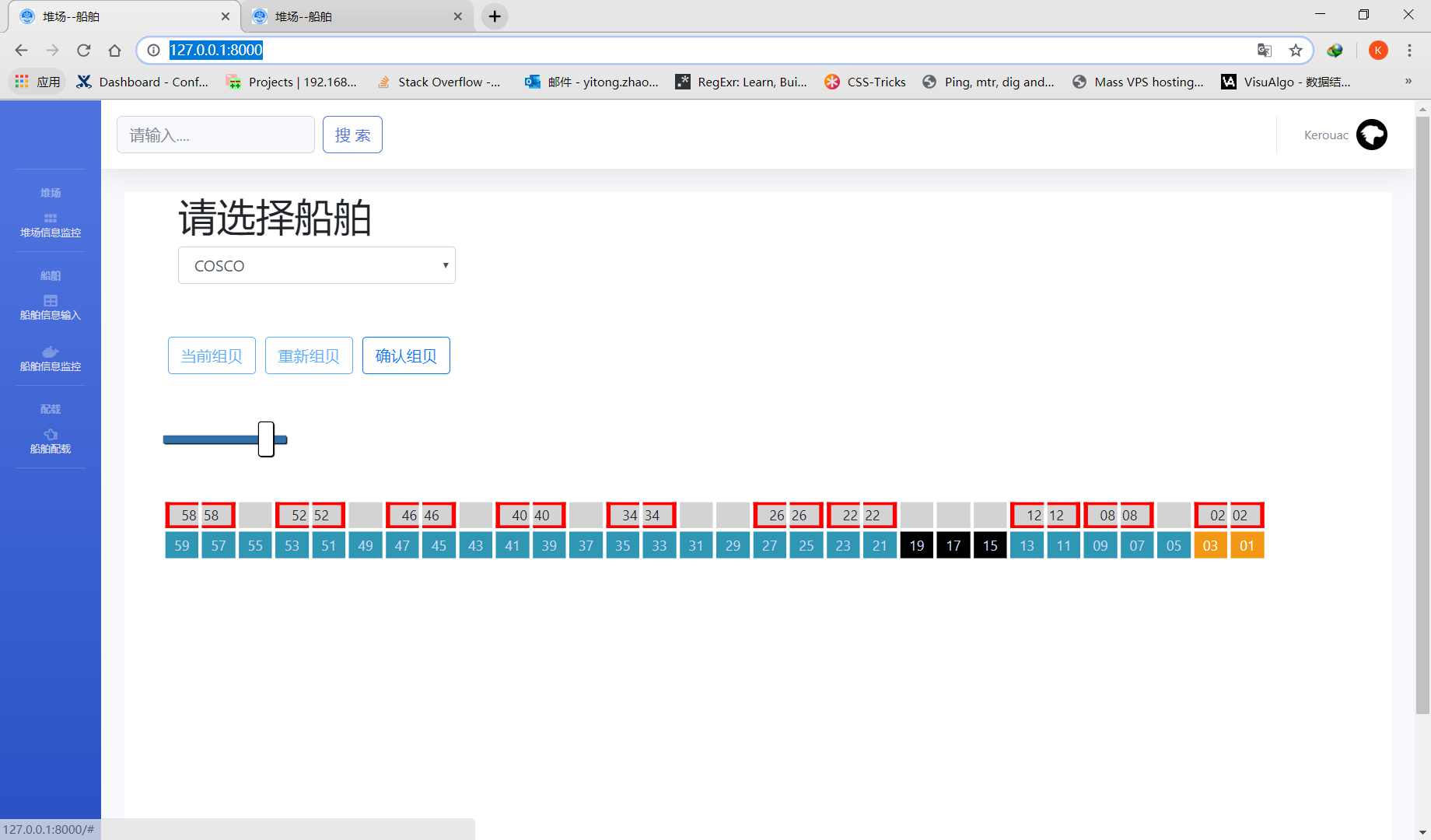
定义贝位结构



船舶信息输入中选择定义贝位结构，

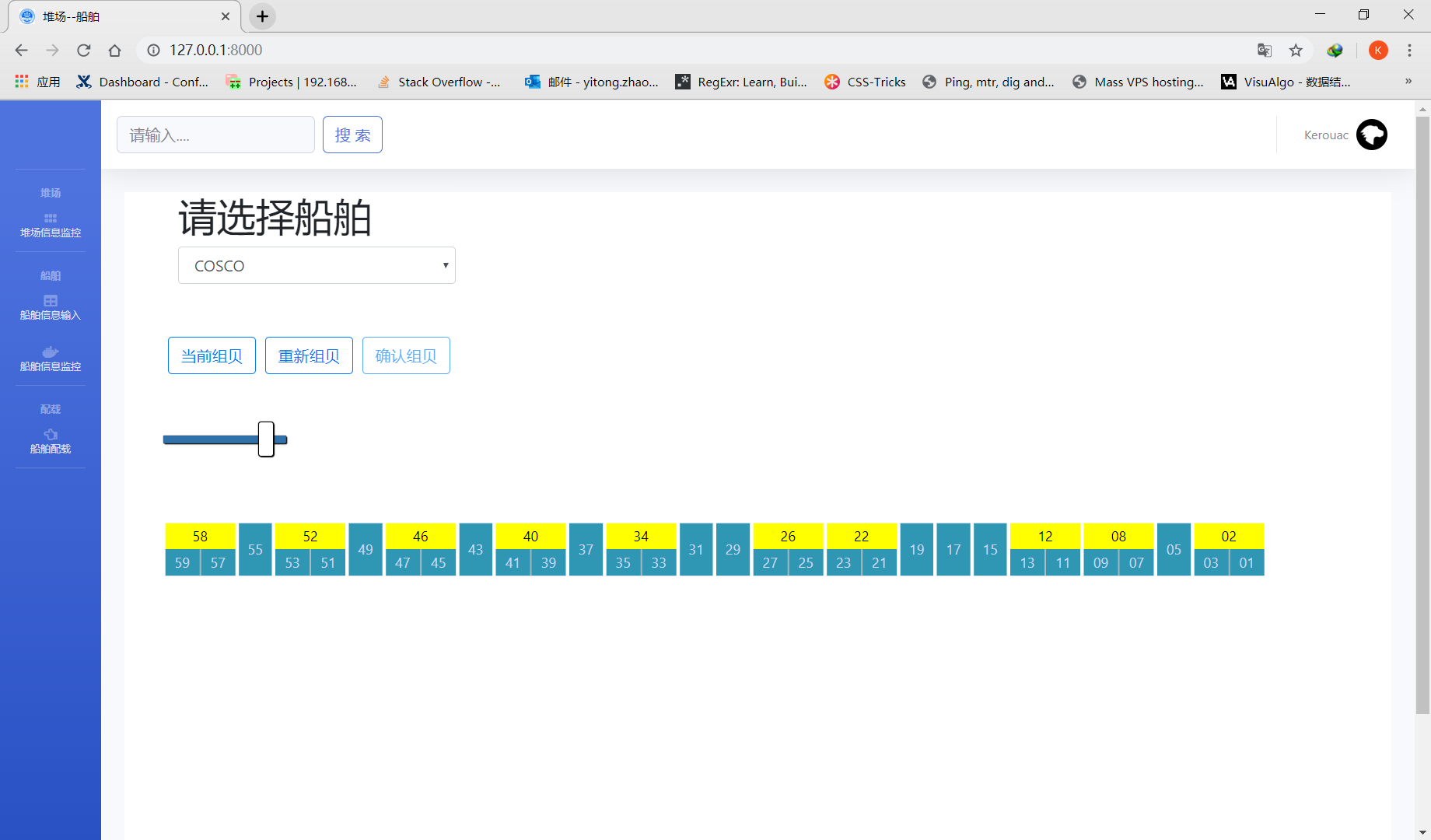
通过下拉框可以选中需要定义的船舶，点击当前组贝，即可查看当前船舶的组贝情况

（未组贝时默认全部为20贝位，已组贝时则混合显示）



点击重新组贝，即可对当前选中的船舶对应贝位进行组贝，其中，贝位涂黑处表示船舶机舱位置（实际组贝作业中，船舶机舱位置是不允许组贝的），这里页面进行了组贝限制操作，不允许机舱位置进行组贝操作。

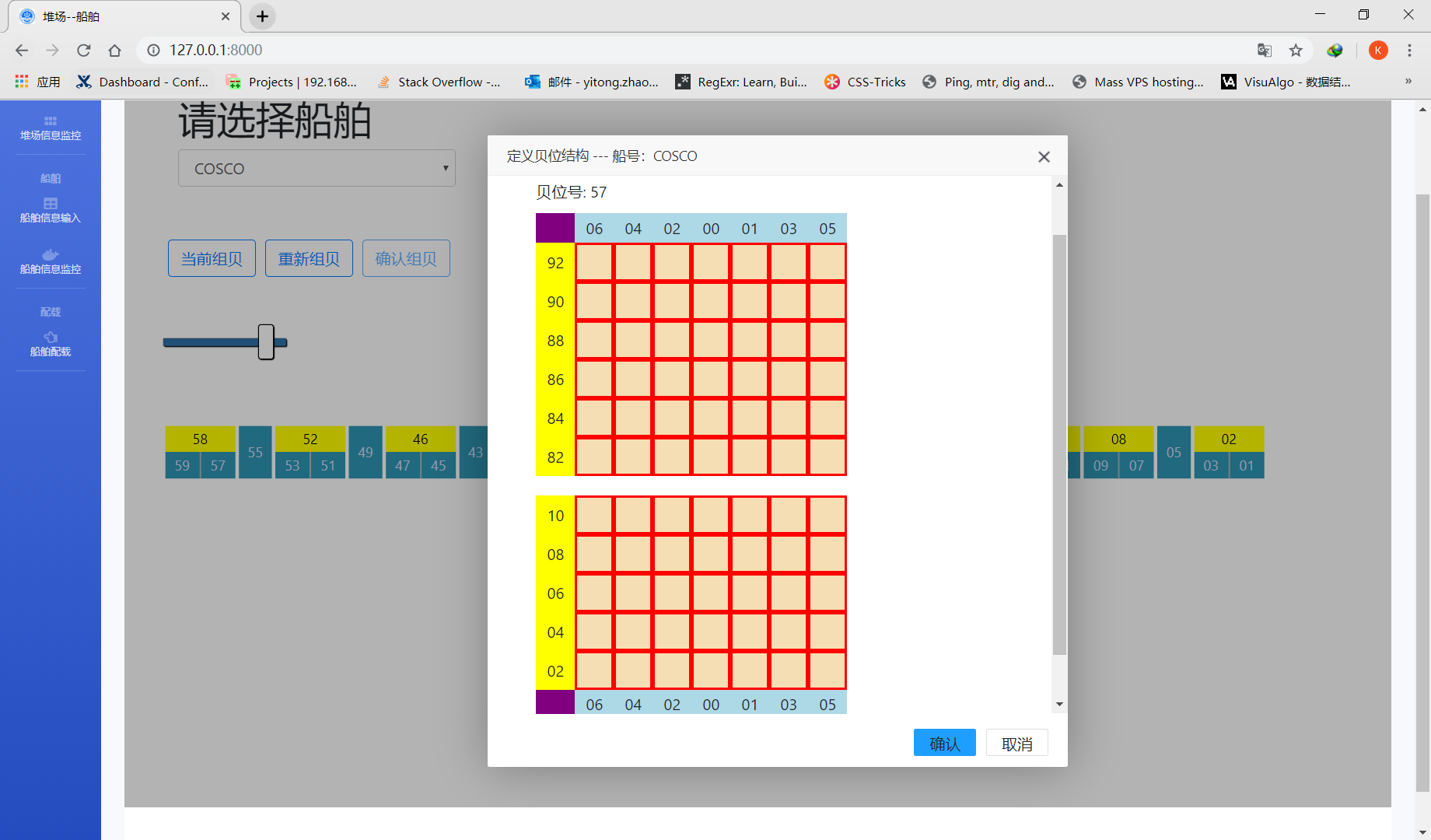
当组贝完成后，点击确认组贝，即可提交当前组贝信息，再次选择对应船舶，点击当前组贝，即可查看当前组贝信息。



如果需要修改当前组贝信息，点击重新组贝，重复以上操作，即可更新组贝信息。

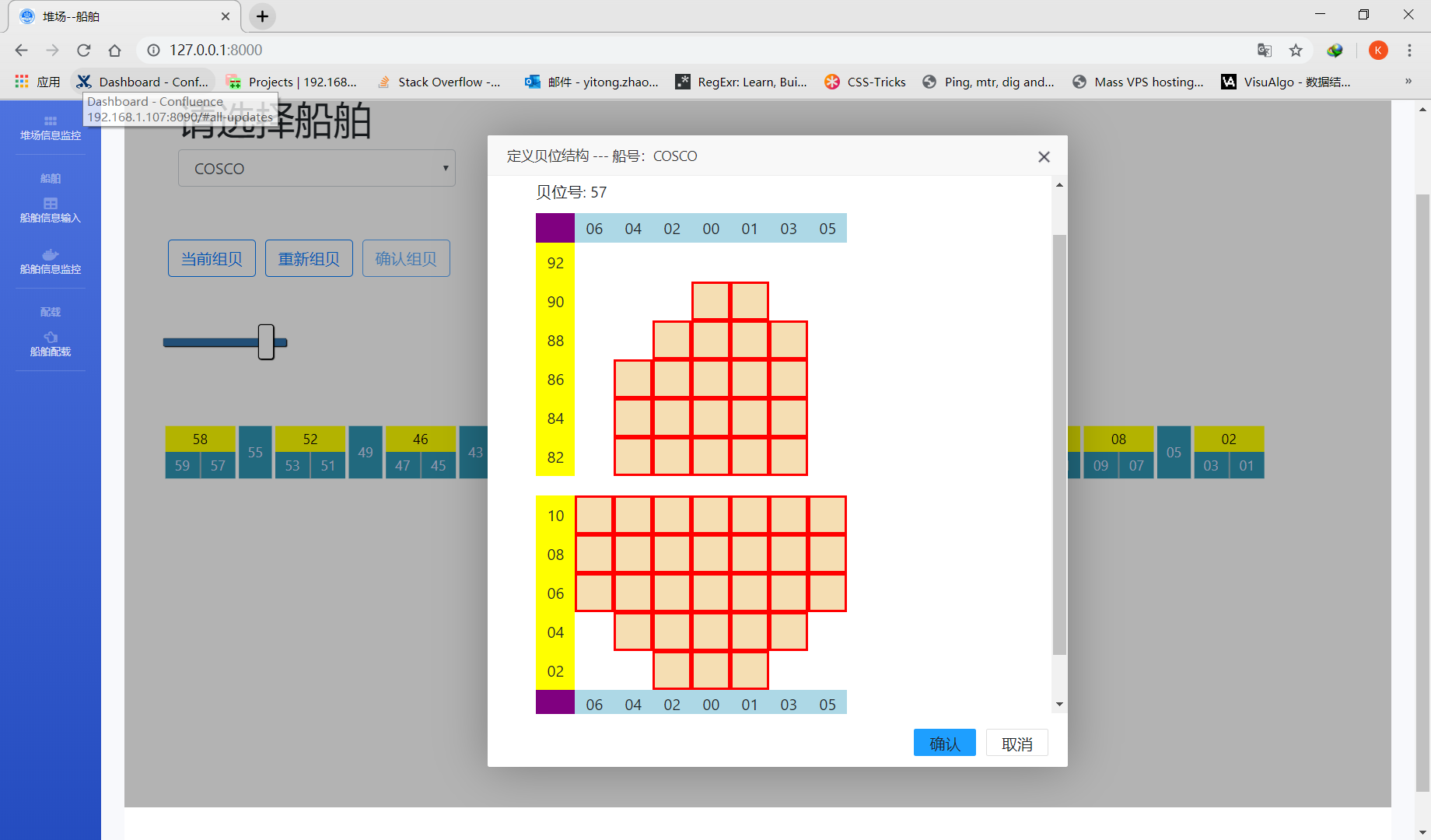
组贝功能结束后，考虑到实际船舶配载过程中，船舶结构不一，部分箱位设置需要特殊处理，故需要进行贝位结构定义。

在当前组贝的页面下，点击其中一个贝位（20尺），即可查看当前贝位结构信息



如图，弹出框中，显示当前贝位结构信息（未定义状态），基本信息包括船名，贝位号，贝位层号，贝位列号

通过双击不同层下的不同列，即可取消当前预设箱位，

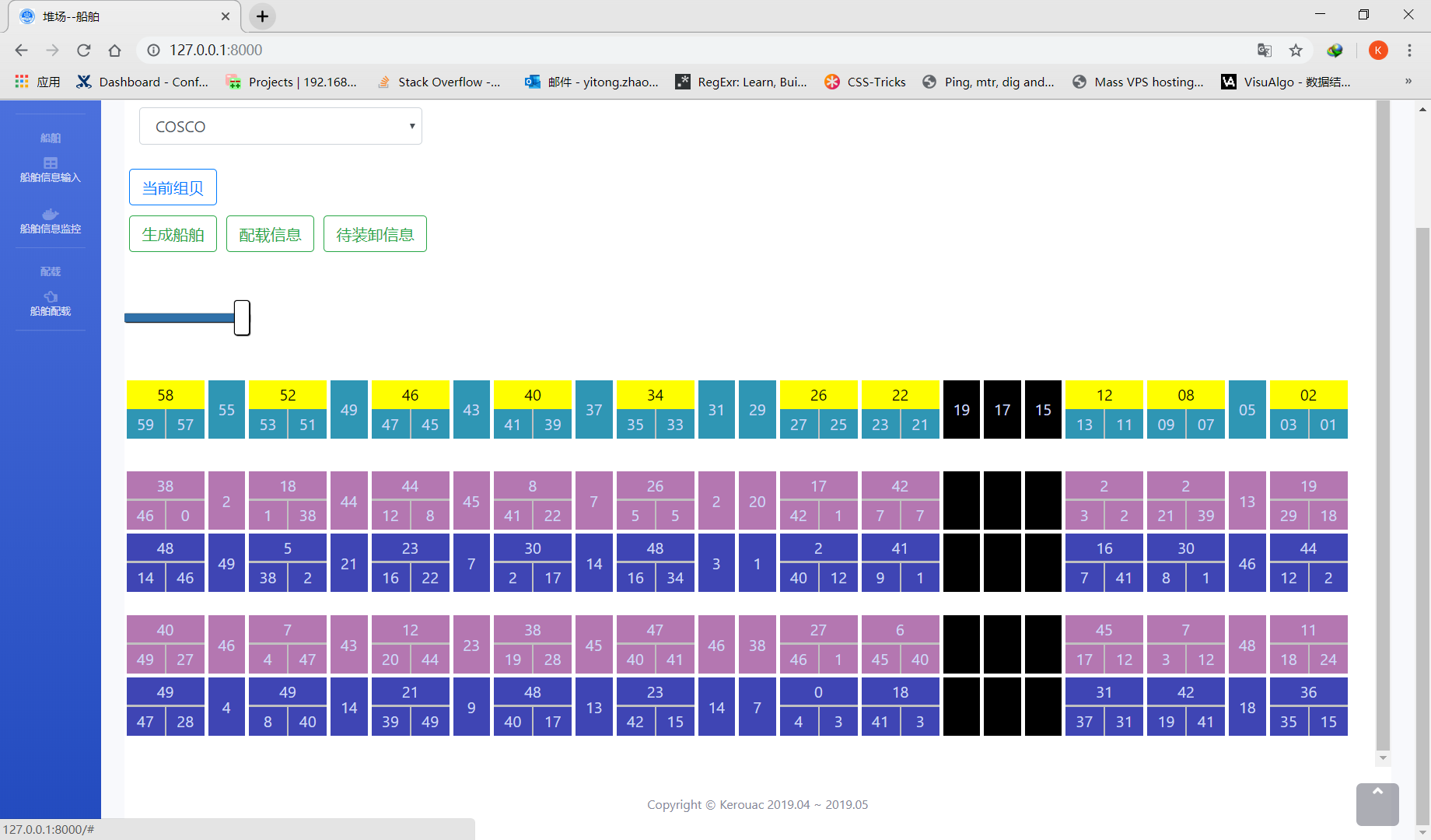


点击确认后，即可提交当前贝位结构信息，其他贝位结构定义，同上。定义结束后，再次点击某个贝位，即显示最新的贝位结构信息。

###### 船舶信息监控

配载信息

选择船舶信息监控， 下拉框中选择一个船舶，点击待装卸信息，即可显示船舶装卸信息

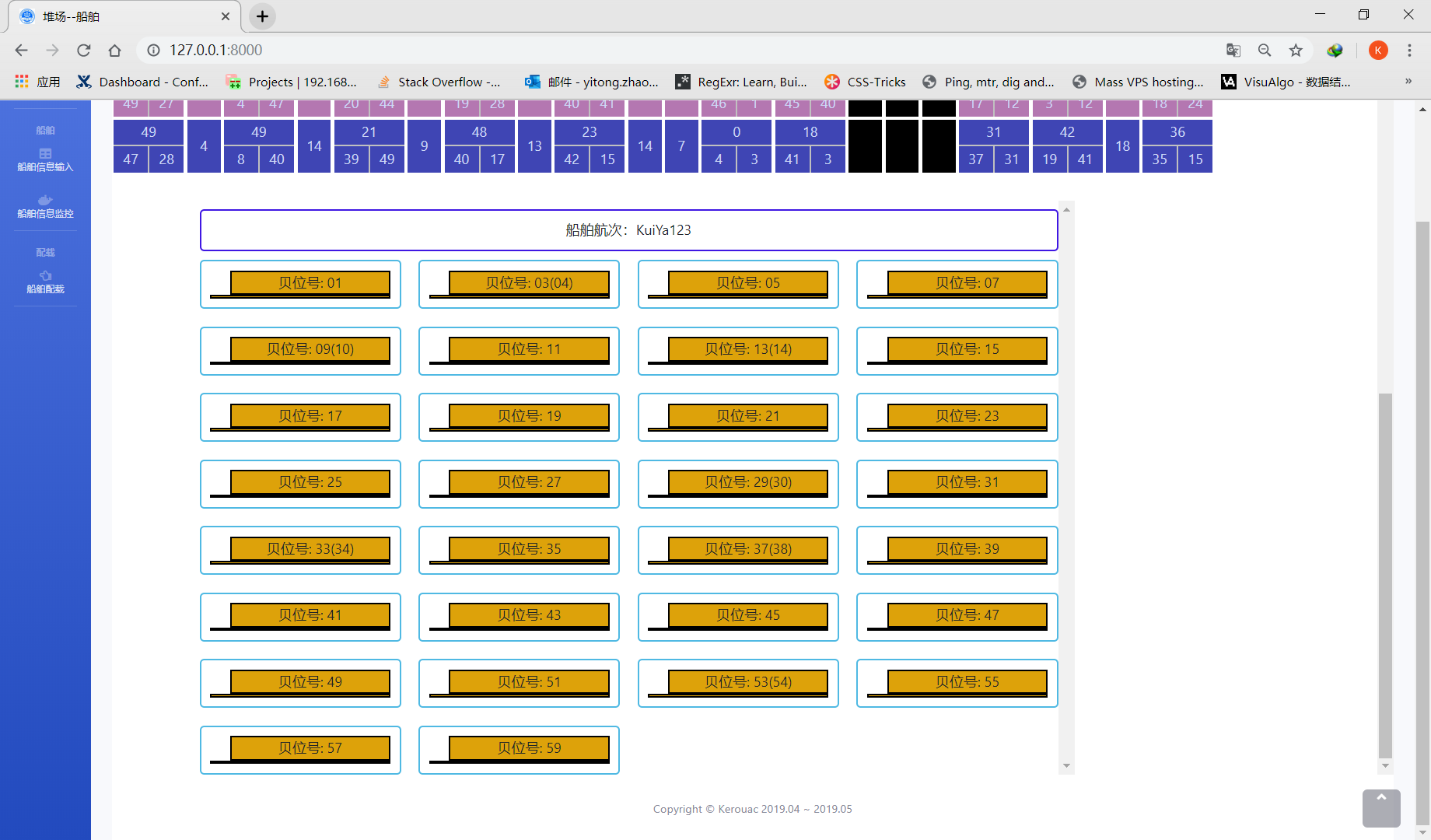


考虑到实际船舶装卸作业需要先卸再装，分为甲板箱和舱内箱，图中，涂黑部分即船舶机舱位置，上下一次是每个贝位的甲板卸，甲板装，舱内卸，舱内装（图中装卸数据的输入部分，有船舶配载部分完成）

点击生成船舶按钮，即可生成当前船舶集装箱装载状态，

图\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

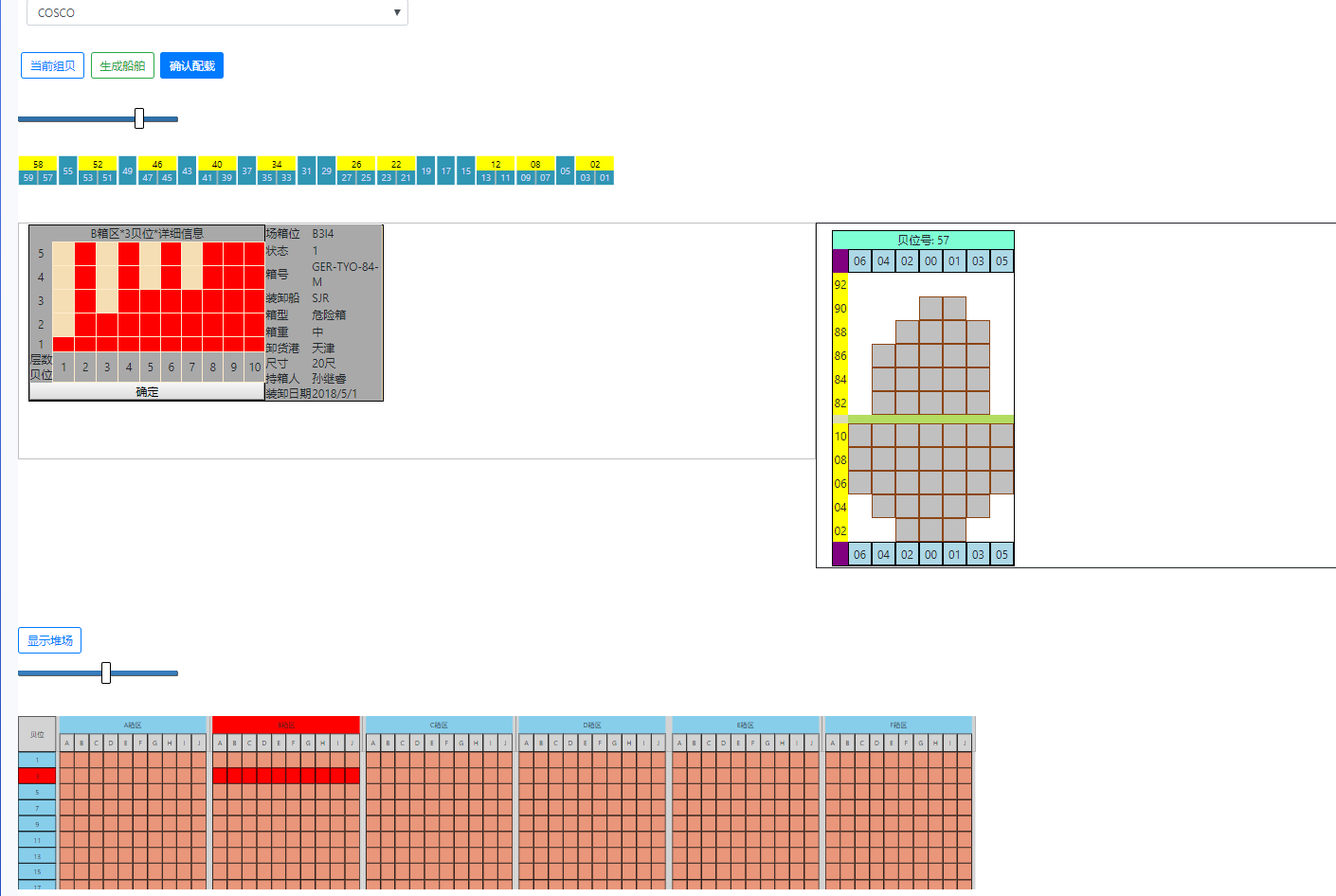
点击配载信息按钮，即可查看所有贝位的配载信息



////////////\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

###### 船舶配载

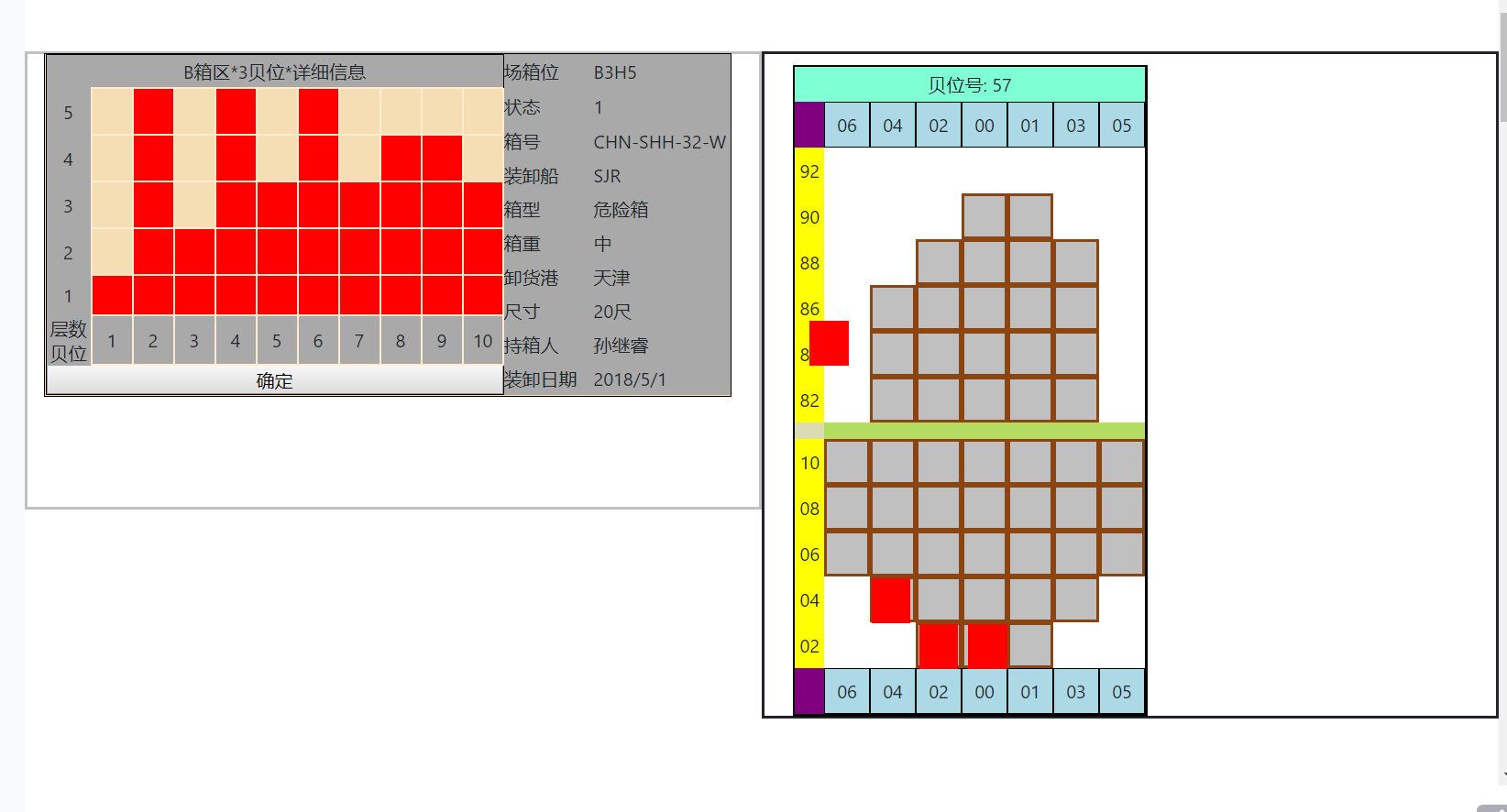
由于本系统主要研究从堆场到船舶的集装箱作业，故这里增加了堆场集装箱到船舶贝位的手工集装箱配载功能



如图，点击船舶配载，即可加载出以下页面功能，内容包括贝位结构信息查看，贝位现有集装箱信息明细， 堆场某一箱区某一贝位的集装箱信息

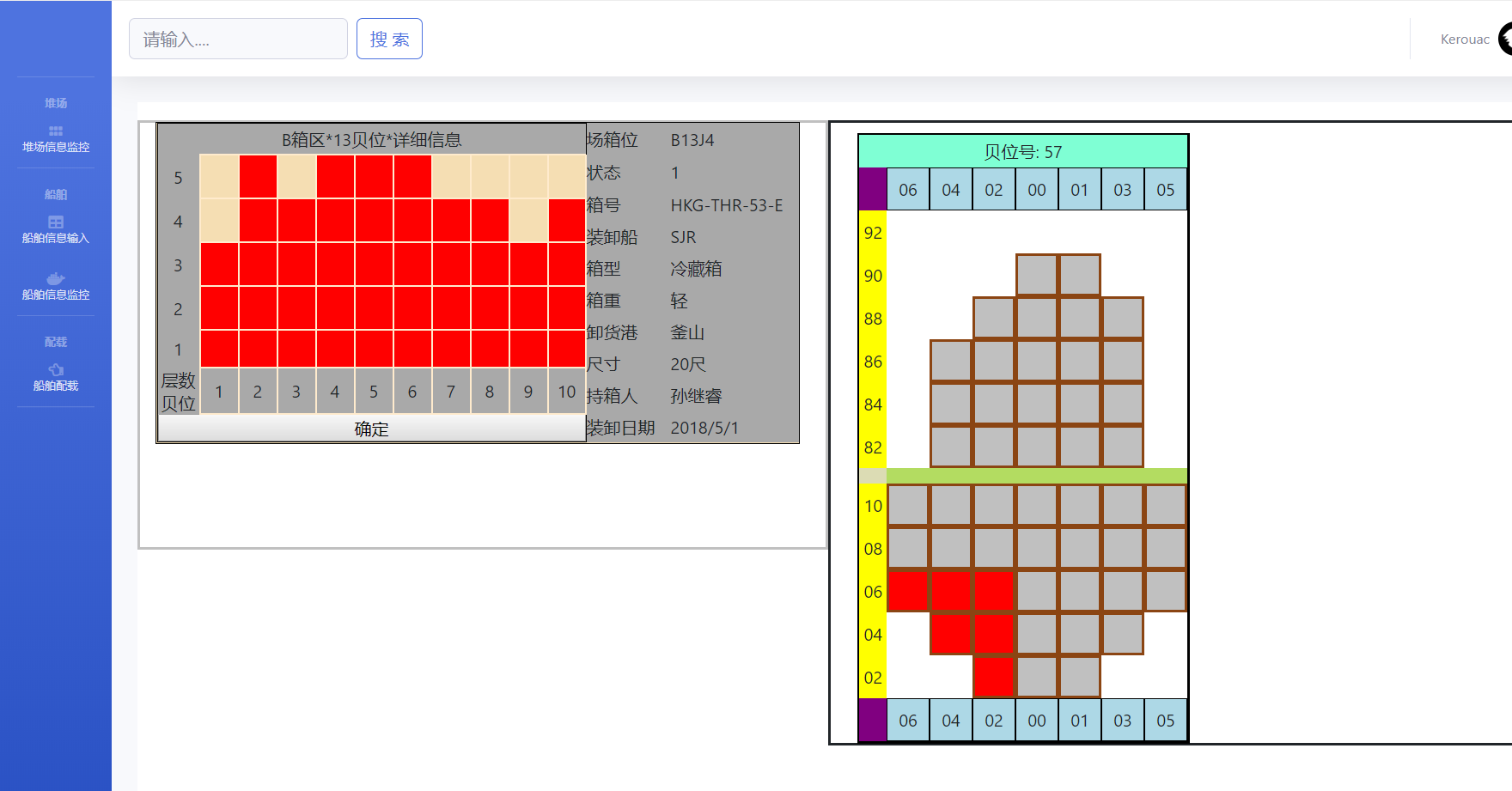
配载部分，这里引入手动配载功能，即鼠标在堆场箱区中，选择需要配载的集装箱，点击并拖动至目标贝位对应箱位，即可完成配载

拖动过程如下



如图，集装箱依次被手工拖动到指定箱位，点击确认配载按钮，即可保存当前配载信息。

具体结果如下



输出设计

* 1. 模块功能与处理过程设计

HIPO 图

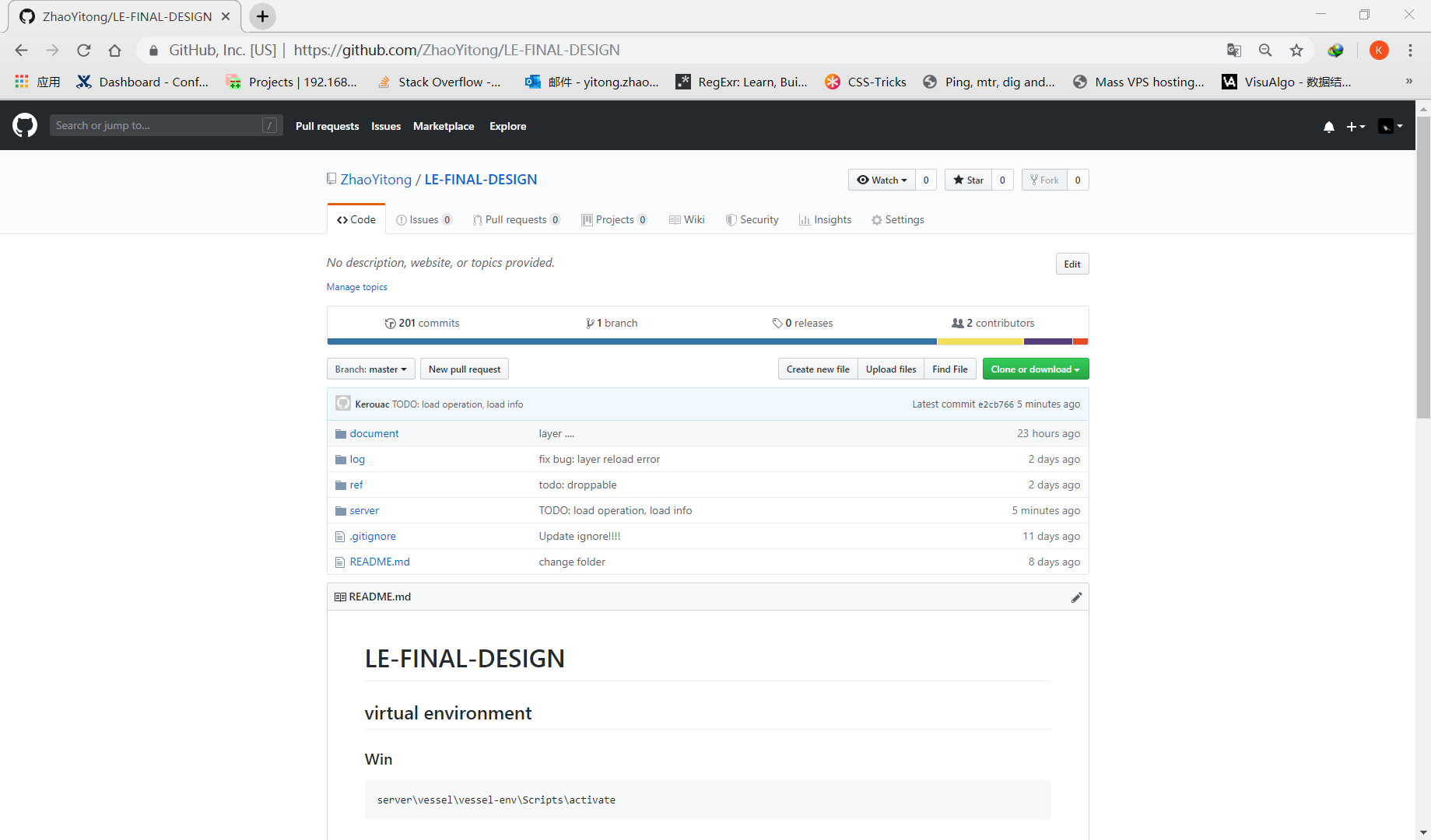
层次模块结构图

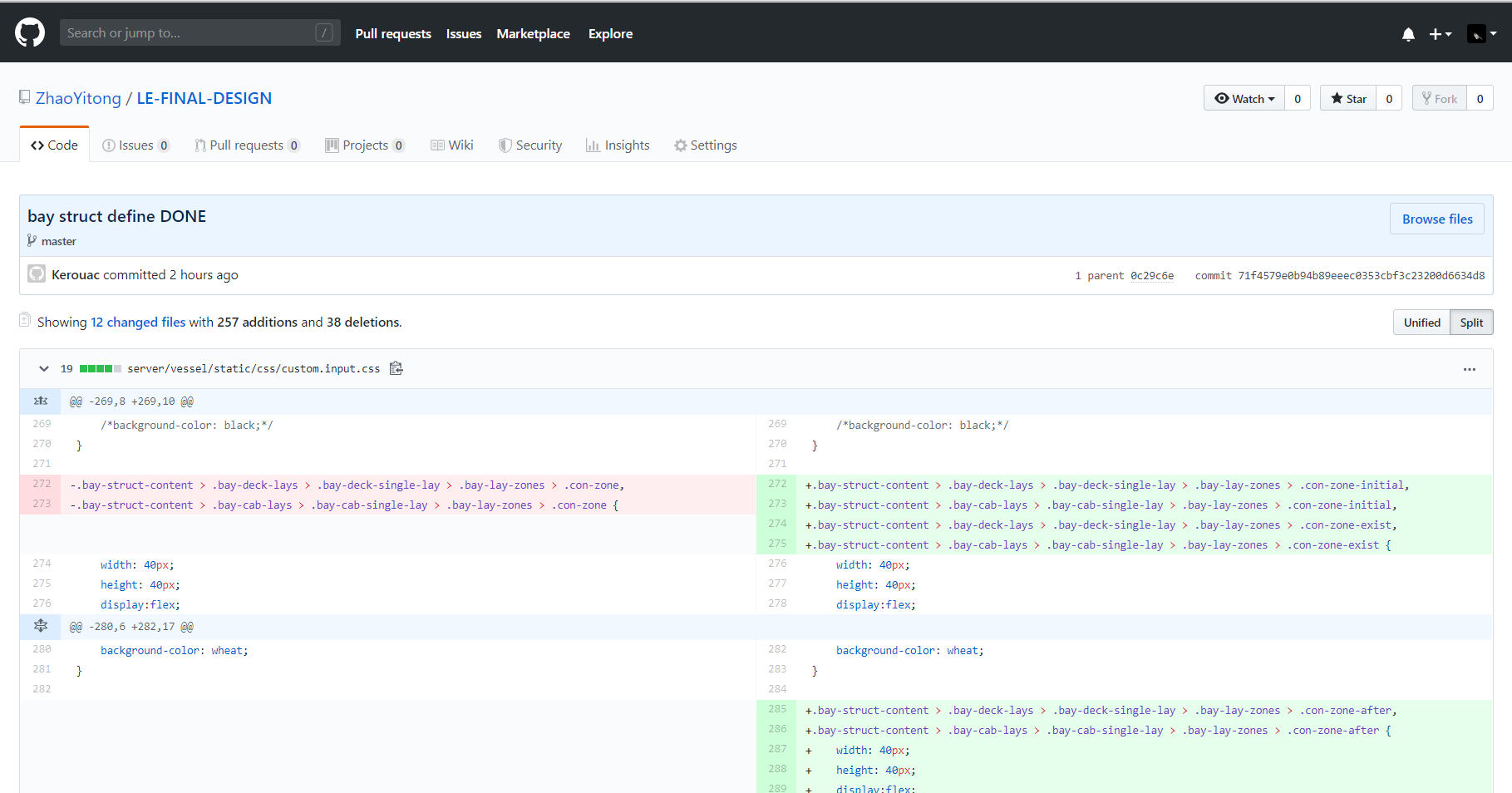
IPO 图

1. 系统实施、评价、运行
   1. 系统实施
      1. 程序设计与调试

版本控制工具

Github 代码托管平台





Git 代码控制工具

1. 项目拉取 见图

git clone <https://github.com/ZhaoYitong/LE-FINAL-DESIGN.git>

1. 更新本地代码库

git pull –rebase

1. 提交并更新远程代码库：

git status #查看当前已经增加、修改或者删除的文件

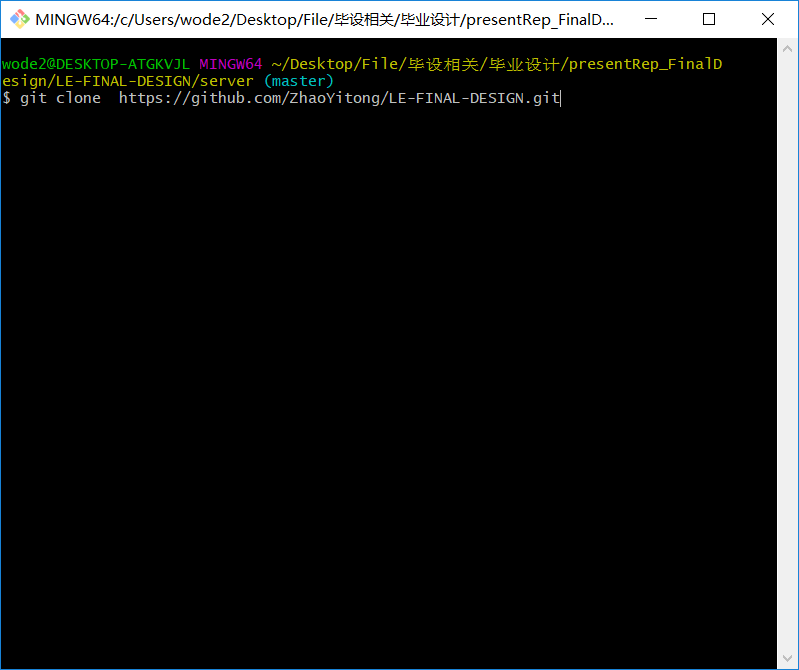
git add . #将当前所有修改放置 stash 暂存区

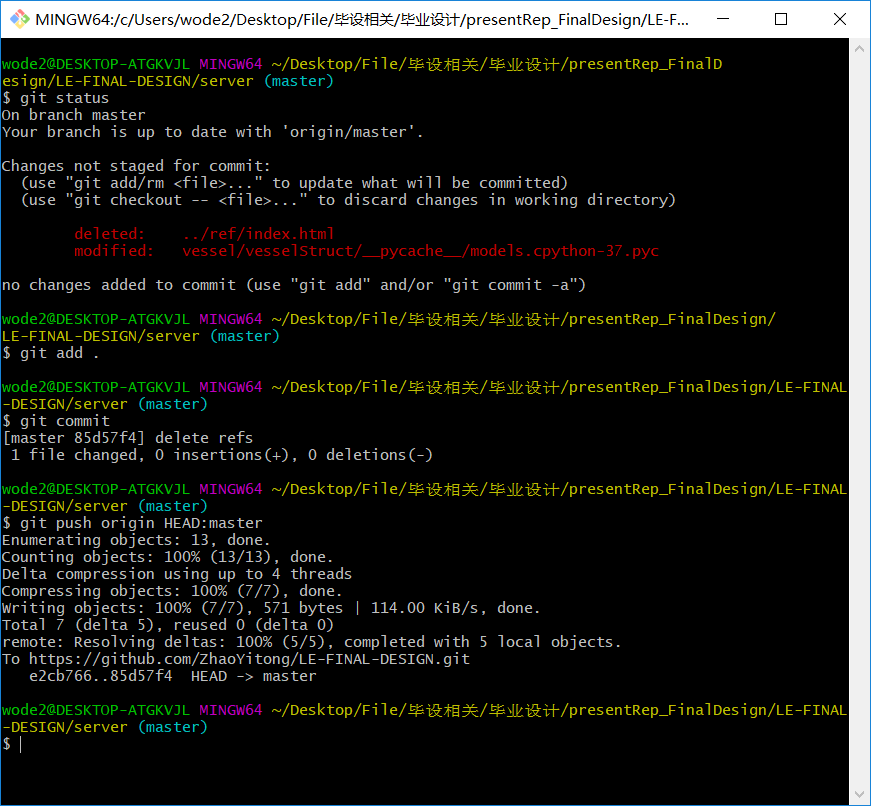
git commit #做提交指令，并可以选择性输入本次commit 做了哪些修改

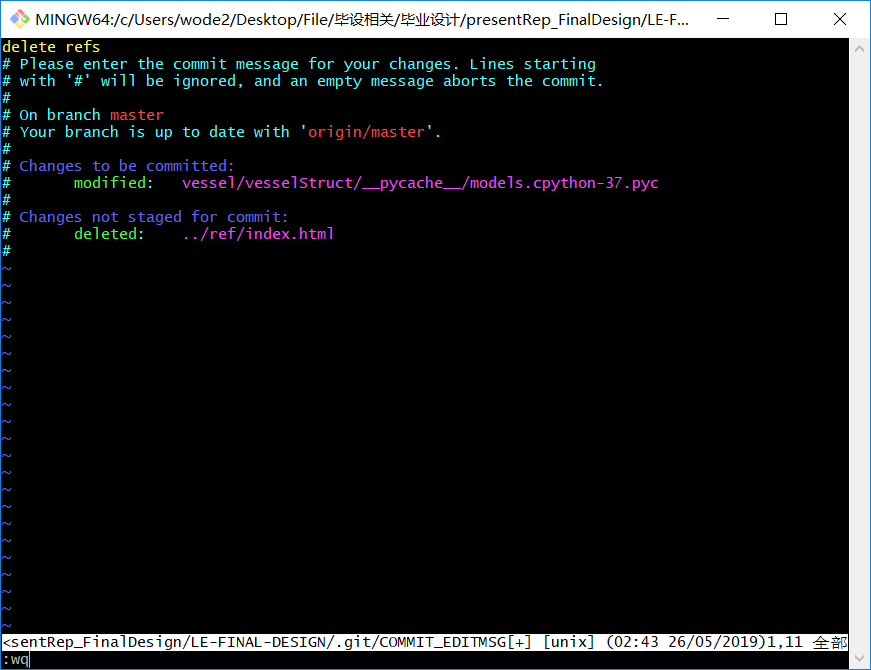
git push origin HEAD:master 将本地commit 提交到远程分支（即github托管的代码仓库）

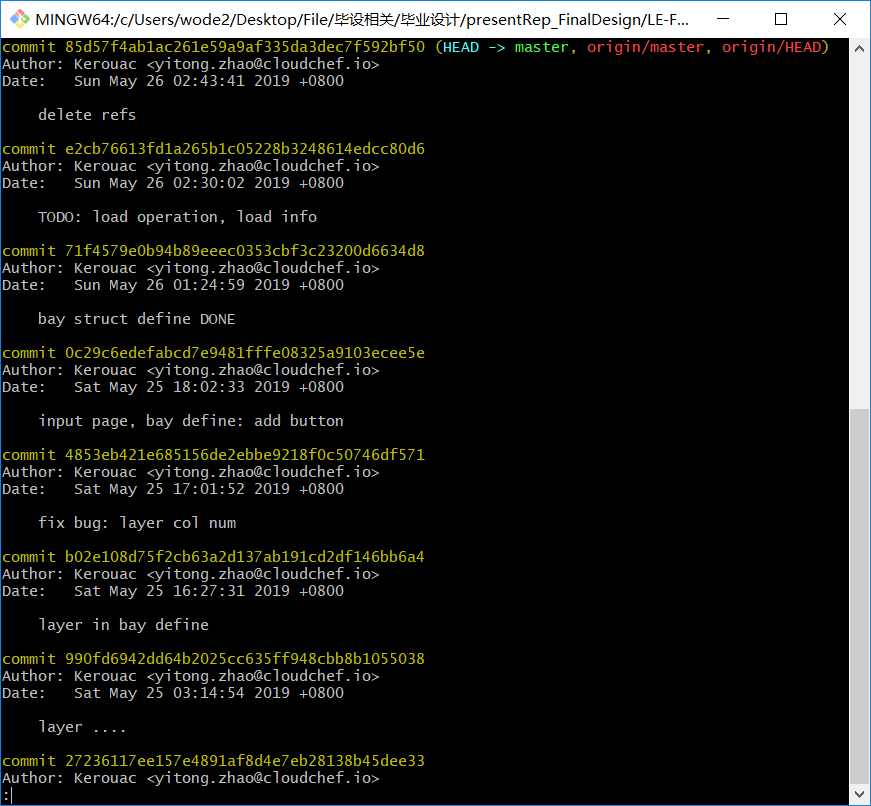
另外：本次开发设计到多个人的功能集成，以及代码库版本的不断更新，中间或遇到版本回退问题

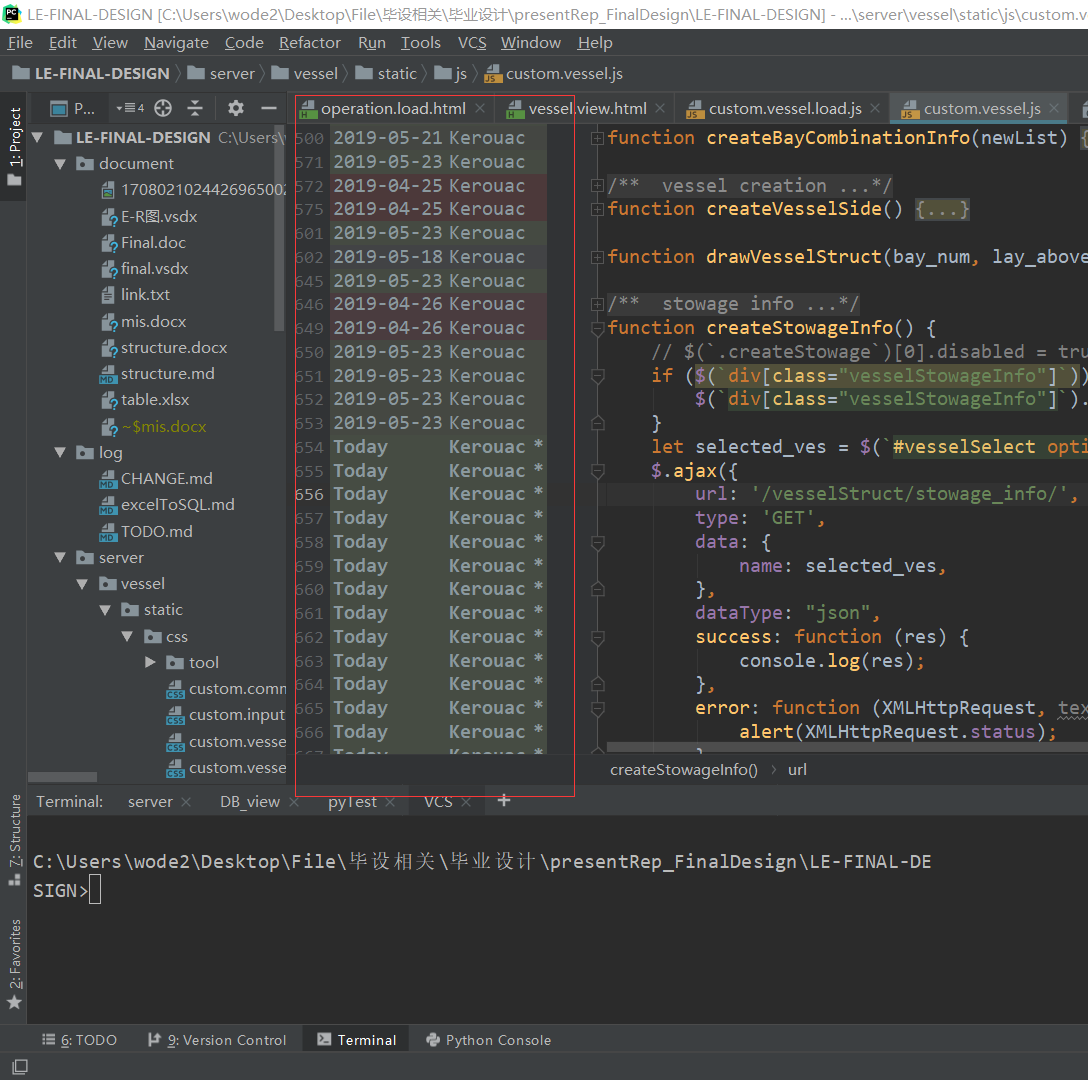
git revert –hard <commit ID> 即通过 git log 记录查看commit 信息，选择相应ID 进行版本回退， 见图………

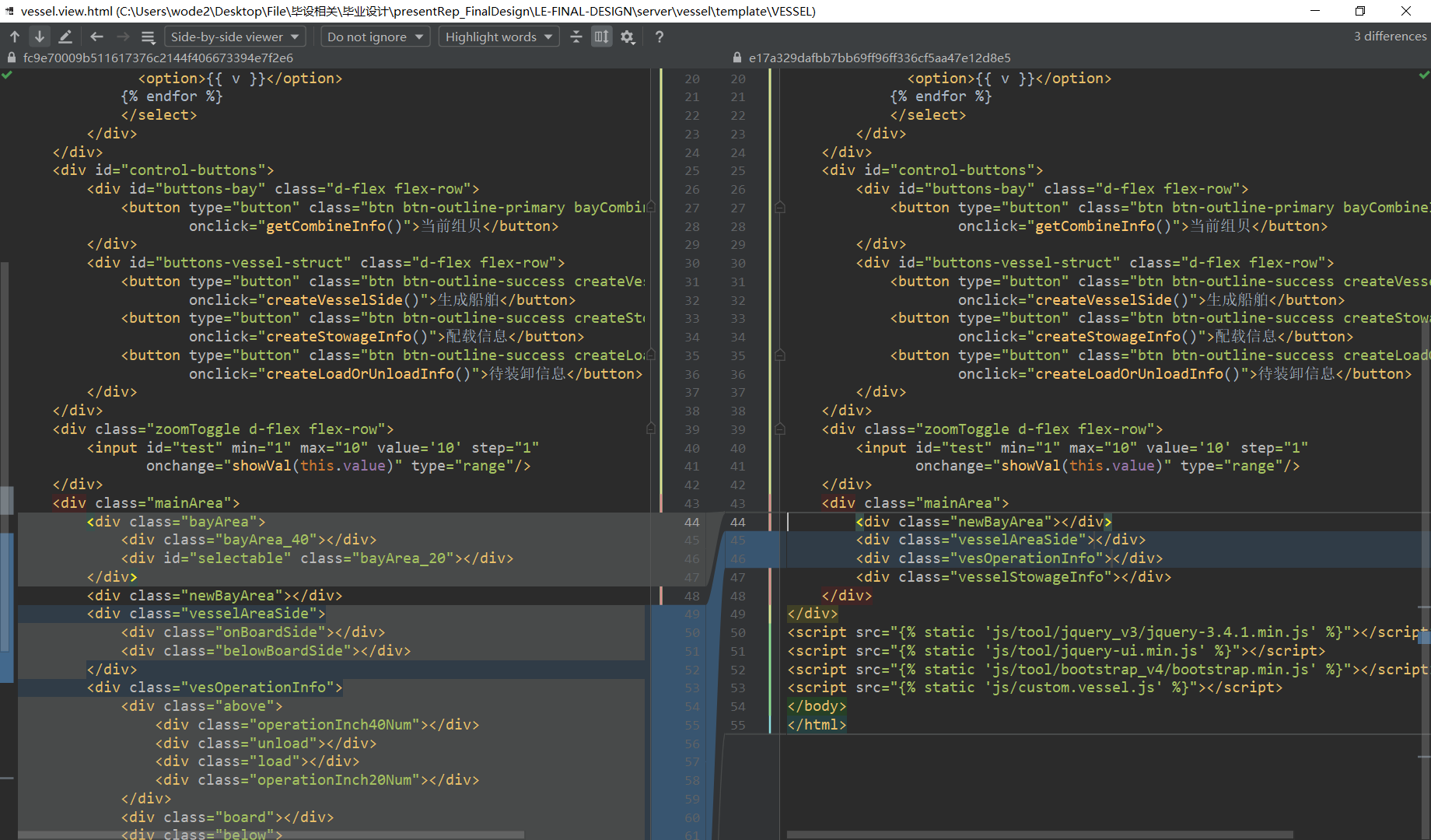












编程工具

语言：

前端 Html, JavaScript, CSS, jQuery,

后端: python SQL

框架：

前端：jQuery UI, SB-admin

后端：Django

编程软件：

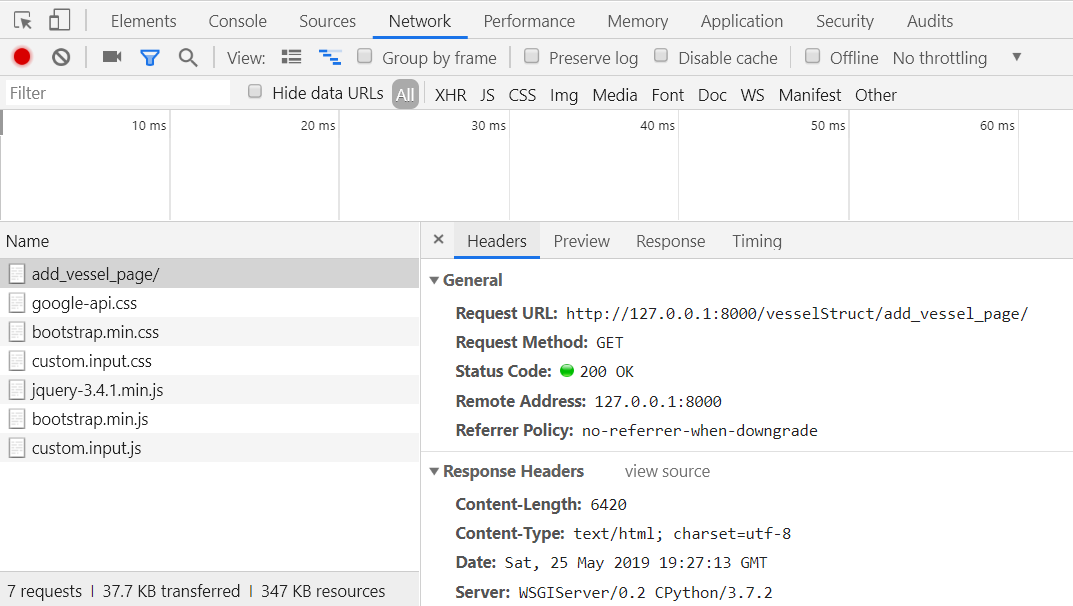
JetBrain PyCharm



编程调试工具

前端

Chrome 浏览器控制台



后端：

Django Shell

Python manage.py shell

1. 项目部署

以下简要讲解部署过程：

* 1. 数据库部分

数据库部分使用 mysql， 通过命令行操作如下

Mysqldump -uroot -Ppassword –database > NAME.sql

导出最新的数据库

Source NAME.sql 导入数据库

后端部分

需要安装 python 环境， django 框架， python 宏包部分需要 mysqlclient,

前端部分

代码库本地已经集成所有静态文件

后台启动服务

Python manage.py runserver <端口号（默认 8000）>

