# 《登录页》



输入用户名密码后开始向后台发起登录请求，交互过程中要体现“登陆中”、“登陆失败”等状态，登陆成功后将跳转至《主页》，并且不可以通过“后退”动作重新回到登录页。再次回到登录页需要通过“退出登录”操作、或者令牌超时后触发。（$router.replace(‘/home’);）

登录后记录登录信息，下次打开app后可以免登录直接以上次登录身份进入app。

# 《主页》

# 

这里列有3个固定按钮，和两个信息显示。信息显示包括：“当前用户名”、“登陆时间”。

点击“拣配”按钮将会跳转至《查询出库单》页面。点击“查询拣配单”按钮将会跳转至《查询拣配单》页面。点击“退出登录”按钮将会询问用户是否确认退出登录，确认后重新回到登陆页面。并且不能再通过“后退”动作回到主页。（$router.replace(‘/login);）

# 《查询出库单》



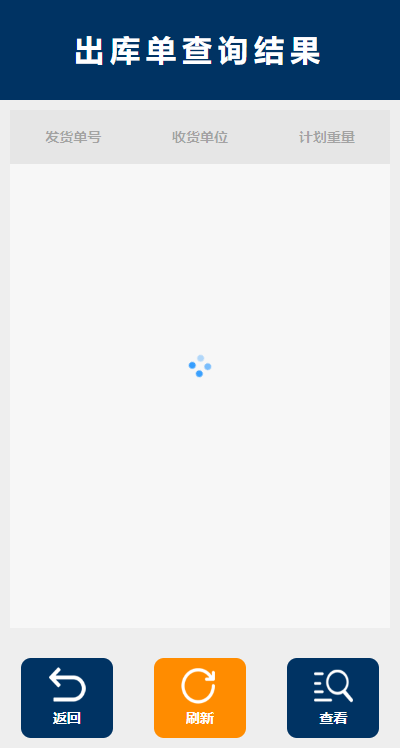
上有3个查询条件，选择日期区间、发货单号（输入框）、库区（单选）。其中库区的可选项值是静态的（前端写死即可），且只有一个选项：

[{ label: "1403库区", value: 1403 }]

默认选中此项。

点击“返回”按钮后执行“后退”操作（路由后退），点击“查询”按钮后跳转至《出库单查询结果》页。

# 《出库单查询结果》



根据上级页面传过来的参数查询并展示表格数据。

点击“返回”按钮触发路由后退动作。点击“刷新”按钮则触发重新查询动作。点击“查看”按钮后判断是否已经选中了表格当中的一条数据，如果没有则提示用户选中一条表格数据，如果已经选中某一条数据，则附带参数跳转至《出库单内容》，参数为整条数据（根据选中的数据展示其具体详情）。

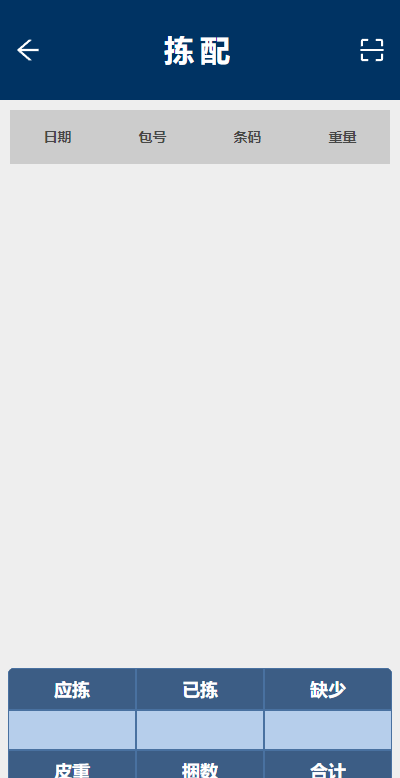
# 《出库单内容》



根据传参将值分别填入以上空格（只读）内用以展示。

点击“车号”右侧的按钮用来选择车号，跳转至《选择车号》。点击“返回”按钮触发路由后退。点击“扫描”按钮需判断必须已经选择了车号（车号不能为空），跳转至《扫描结果》

# 《扫描结果/拣配》





点击左上方箭头触发路由后退。点击右上角“扫描”触发手机摄像头扫描。

此页面需添加监听事件，如果长时间（7200秒）停留在此页面没有动作的话将会自动进行登出动作并跳出到《登录页》。

当监听到扫描的新数据后，需要对数据进行如下判断：

* 先验证条码是否符合标准格式，方法函数如下：
* //检查条码是否符合规范
* tool.checkBarcodeIfqualified = barcode => {
* const standard = new RegExp(
* /^124010[1-9][0-9]{2}(0[1-9]|1[0-2])((0[1-9])|((1|2)[0-9])|30|31)[0-9]{12}$/,
* );
* const bool = standard.test(barcode) && barcode.length === 25;
* return bool;
* };
* 不符合规范时，提示用户“此条码不符合规范！”并停止后续动作。符合规范时，向后台请求验证条码是否已经拣配CheckBatchNo。如果已经拣配，则提示用户“该批次已经拣配装车，请勿重复选择！”，如果未拣配则继续解码操作。
* //如果未拣配
* else {
* //按规则解析条码
* return decode(data)
* .then((result) => {
* let date = result.shengchanriqi,
* number = result.kunxuhao,
* barcode = result.barcode,
* weight = result.kunjingzhong;
* let databar = {date, number, barcode, weight};
* return checkIfExist(existDatabar, databar);
* })
* .then(({result, obj}) => {
* if (!result) {
* return obj;
* } else if (result) {
* this.refs.tips.show('该批次已经选择，请勿重复选择！');
* return;
* }
* });
* }

//解码规则

const decode = (barcode) => {

  let arr = barcode.split('');

  //企业代码 三位

  const qiyedaima = arr.slice(0, 3).join('');

  //产品类别代码 二位

  const chanpinleibie = arr.slice(3, 5).join('');

  //产品等级代码 二位

  const chanpindengji = arr.slice(5, 7).join('');

  //生产日期代码 六位

  const shengchanriqi = arr.slice(7, 13).join('');

  //产品批号 四位

  const chanpinpihao = arr.slice(13, 17).join('');

  //捆序号 三位

  const kunxuhao = arr.slice(17, 20).join('');

  //捆净重代码 五位

  const kunjingzhong = arr.slice(20, 24).join('') + `.` + arr[24];

  return new Promise((resolve, reject) => {

    resolve({

      barcode,

      qiyedaima,

      chanpinleibie,

      chanpindengji,

      shengchanriqi,

      chanpinpihao,

      kunxuhao,

      kunjingzhong,

    });

  });

};

//检查是否已存在

const checkIfExist = (arr, obj) => {

  let result = arr.find((item) => {

    return item.barcode === obj.barcode;

  });

  return new Promise((resolve, reject) => {

    resolve({result, obj});

  });

};

解码完成后要判断此解码数据是否已经存在于上方表格中，如果已存在，则提示用户“不要重复添加”。如果未存在，则将此解码数据作为表格数据添加在上方表格当中。注意，新加入的数据条要放在最上面。

下方表格的数据展示：

* “应拣”：由上及页面跳转至此页面后带过来的参数中（不变）。
* “已拣”：上方表格所有数据字段“weight”的加和。
* “缺少”：“应拣” - “已拣”。
* “皮重”：由上及页面跳转至此页面后带过来的参数中（不变）。
* “捆数”：上方表格的数据条数（默认为0）。
* “合计”：“已拣” + “皮重”。

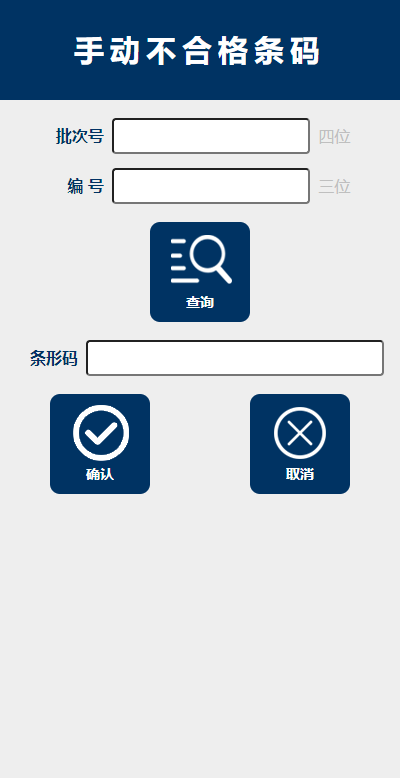
以上数据均精确到小数点后四位。

点击“确认”按钮，询问用户是否确认扫描完成？确认后将上方表格内的数据全部上传到数据库中保存。保存成功后清空表格内的数据。

点击“手动”按钮，带上方表格内的数据作为参数跳转至《手动不合格条码信息》

点击“剔除”按钮，将上方表格中选中的一条数据从表格中剔除出去，并重新计算下方表格应展示的数据。

# 《手动不合格条码信息》



批次号、编号为必填项。

点击“查询”按钮，根据批次号和编号查询条码。将查询到的条码填充至下方“条形码”输入框内。如果没查询到则提示用户查不到。此时用户可以在“条形码”手动输入条码。

点击“取消”触发路由后退。点击“确定”后按照《扫描结果》页内的条码是否符合规范的逻辑进行检验一遍，通过后带着条码作为参数，返回《扫描结果》页并将之作为一条新数据添加在的上方表格内。

# 《选择车号》



上方查询条件是日期区间。

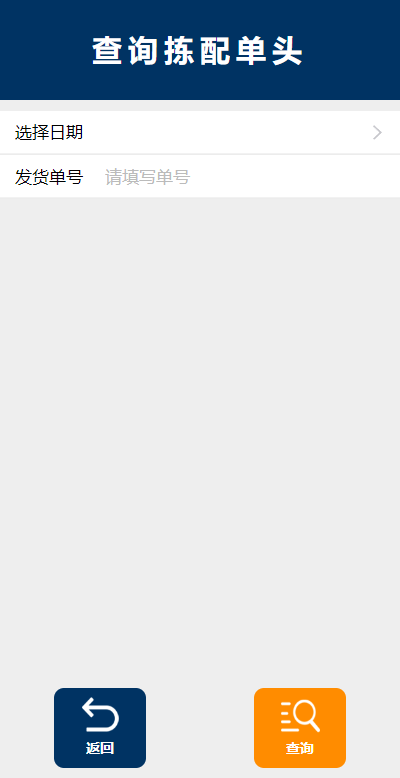
点击“查询”按钮，根据上方条件查询表格数据。点击表格内某一条数据后，将当中的车牌号作为值填充至下方“车牌号”中。

点击表格内的一条数据将此数据中的车牌号填入“车牌号”。

点击“确认”按钮，先判断“车牌号”是否是空的，若为空则提示用户必须选择车号。若不为空则将车号作为参数返回页面《出库单内容》，将之填在“车号”中。

“手动确认”按钮可以删除掉。

# 《查询拣配单》

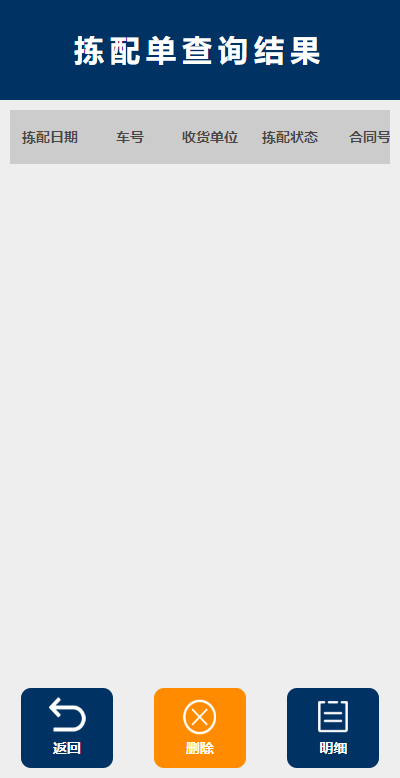


两个查询条件，手动输入框“发货单号”和日期区间选择器。

点击“返回”触发路由后退。

点击“查询”，带着条件跳转至《拣配单查询结果》

# 《拣配单查询结果》



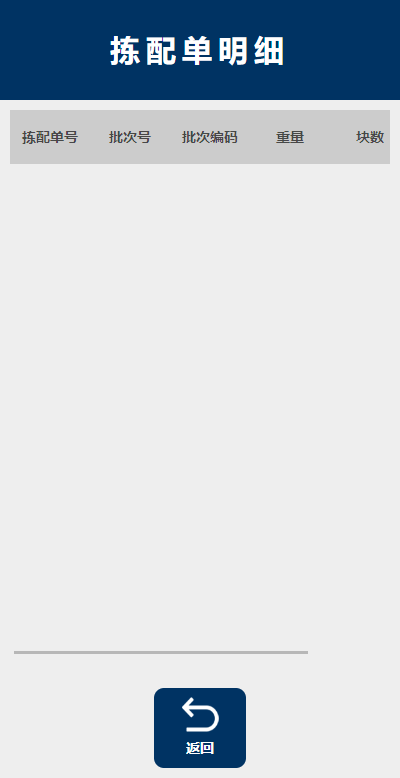
根据上级页面传来的查询参数查询表格数据。

点击“返回”触发路由后退。

点击“删除”按钮（要求先选中一条表格数据），判断词条数据是否已经装车，若已装车则不允许删除，否则询问用户是否要删除此条数据。确认后调用后台接口删除此条数据。

点击“明细”按钮（要求先选中一条表格数据），跳转至《拣配单明细》

# 《拣配单明细》



用路由带来的参数作为查询根据查询详情数据展现在表格中。

点击“返回”触发路由后退。