Order picking

Final report

Universidade do Porto

Faculdade de Engenharia

FEUP

Master in informatics and computer engineering

Information Systems

Group:
Duarte Nuno Pereira Duarte - 201109179 - ei11101@fe.up.pt
Luís Filipe Correia Cleto - 201104279 - ei11077@fe.up.pt
Miguel Rui Pereira Marques - 201109178 - ei11099@fe.up.pt
Ruben Fernando Pinto Cordeiro - 201108177 - ei11097@fe.up.pt

Faculdade de Engenharia da Universidade do Porto
Rua Roberto Frias, sn, 4200-465 Porto, Portugal

December 19, 2014

Contents

1 Introduction

The order picking or order preparation operation is one of a logistic warehouse's process. It consists in taking and collecting articles in a specified quantity before shipment to satisfy customers' orders. It is a basic warehousing process and has an important influence on the supply chain's productivity.

The putaway is an internal logistics process that comprises a set of tasks from receipt of a load to when it is stocked in its final destination within a warehouse.

The goal of the system as a whole is to optimize and automate this process step.

2 Added value

The Management Information System will be used to support the decision of the order picker regarding the logistics of the order shipping and storage inside the warehouse. The client application will be used in a mobile setting.

The application will enable the order picker to:

- See the pending orders from customers.
- Generate a picking list from customer orders, indicating the location of the items to be sent for shipping.
- See the pending orders from the suppliers.
- Generate a putaway list from pending orders from the suppliers.

3 Solution architecture

The application architecture follows a client-server model, as shown in the following diagram:

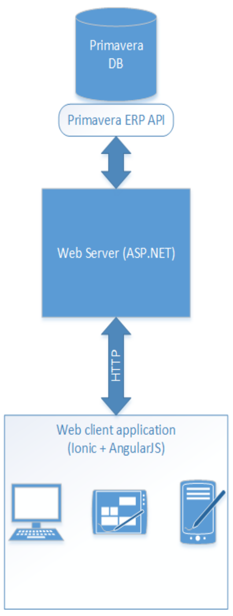


Figure 1: Application architecture diagram

The server was implemented with the ASP.NET framework, as explained in section 4.1. The client was implemented with Ionic Framework and AngularJS, as explained in the sections 4.2 and 4.3.

4 Adopted Technologies and frameworks

4.1 ASP.NET

ASP.NET is an open source server-side Web application framework. We did not use this framework to develop the website, instead, we used this framework in order to implement the REST API server. The language of choice was C#.

Since the Primavera ERP API is also implemented with .net and C#, no additional overhead was introduced server side.

4.2 Ionic framework

Ionic is an open source HTML5 Mobile Framework for building cross-platform hybrid native apps with HTML, JavaScript, and CSS.

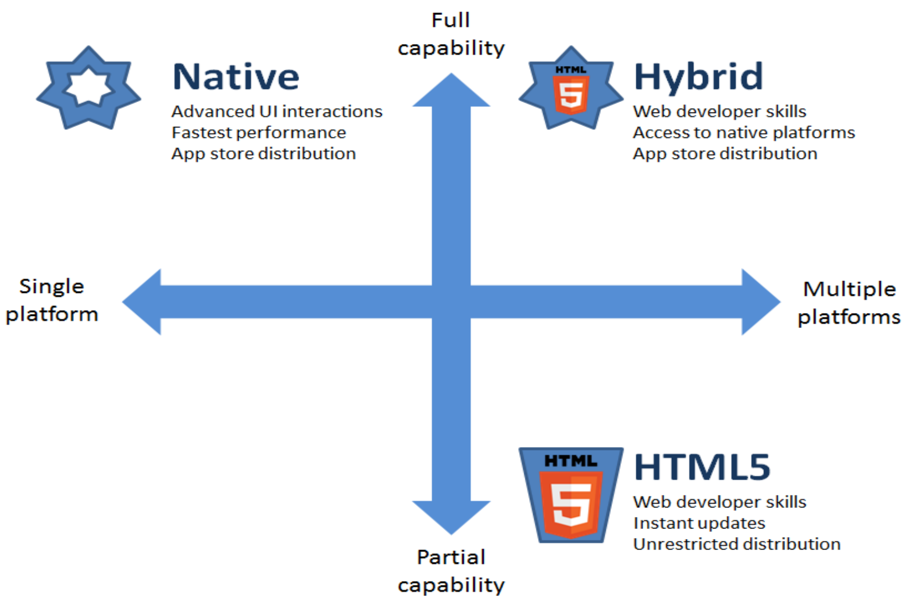


Figure 2: Native vs hybrid applications comparison diagram

This framework provides many UI components that work seamlessly across multiple mobile platforms, which are the target devices of this application. It also provides a hybrid deployment solution that combines:

- Cross platform functionality provided by the web technologies.
- Platform affinity and performance provided by the web view engine bundled with Ionic: Cordova[1].

[1]`http://cordova.apache.org/`

4.3 AngularJS

Instead of a set of static HTML pages, our client application is an AngularJS single page application. AngularJS is a javascript MVVM[2] framework.

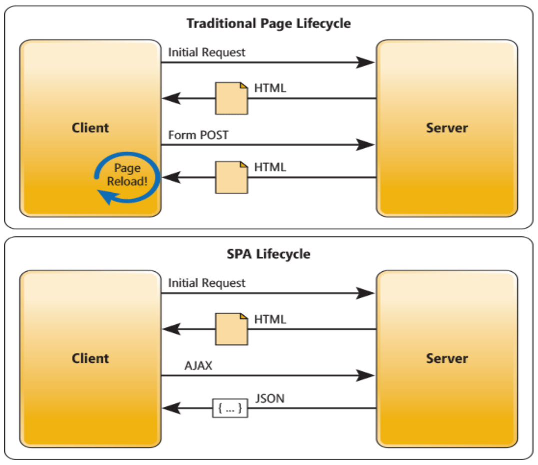


Figure 3: Single Page Application diagram

The adoption of this framework had several design goals:

- Decouple DOM manipulation from application logic. This substantially reduced boilerplate code and improved modularity and testability.
- Decouple the client side of the application from the server side. This reduced much of the burden on the backend and allowed development work to progress in parallel with minor hiccups.
- Provide an incremental and iterative development process for the client, from the UI design to the writing of the business logic.

[2]Model-View-View-Model

5 User stories

The following table contains the user stories for the application:

User Story	Description
US-001	As a User I want to login in the application.
US-002	As a User I want to see the customer orders to be picked for shipping.
US-003	As a User I want to generate the picking list based on a set of orders from clients.
US-004	As a User I want to see the orders from the suppliers to be stored in the warehouse (putaway).
US-005	As a User I want to generate a putaway list based on a set of incoming orders.

Table 1: User stories

6 Core views

6.1 UI01 - Customer orders

The user can see the undelivered orders in this view.

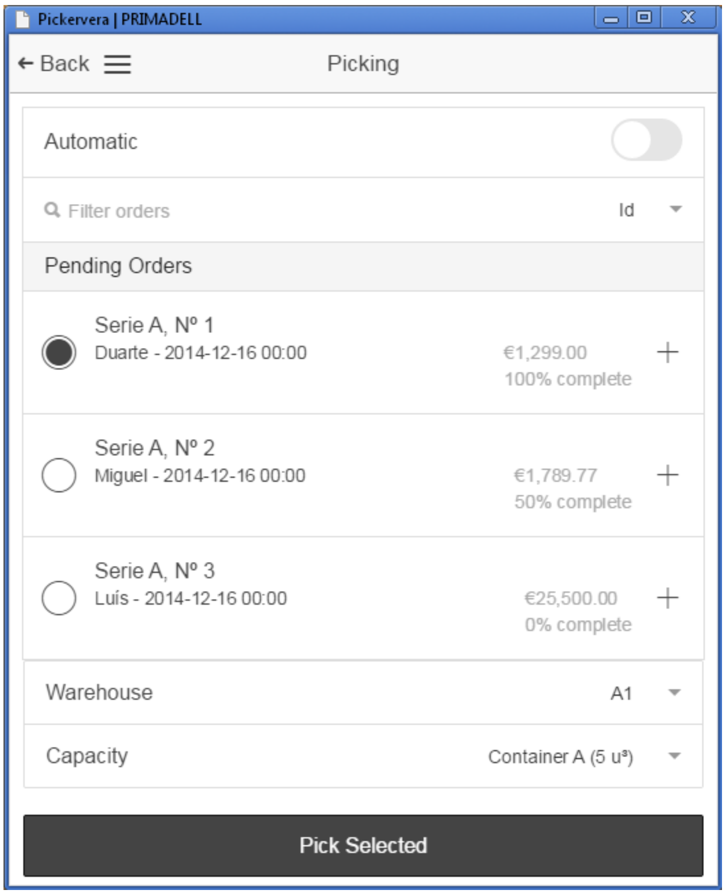
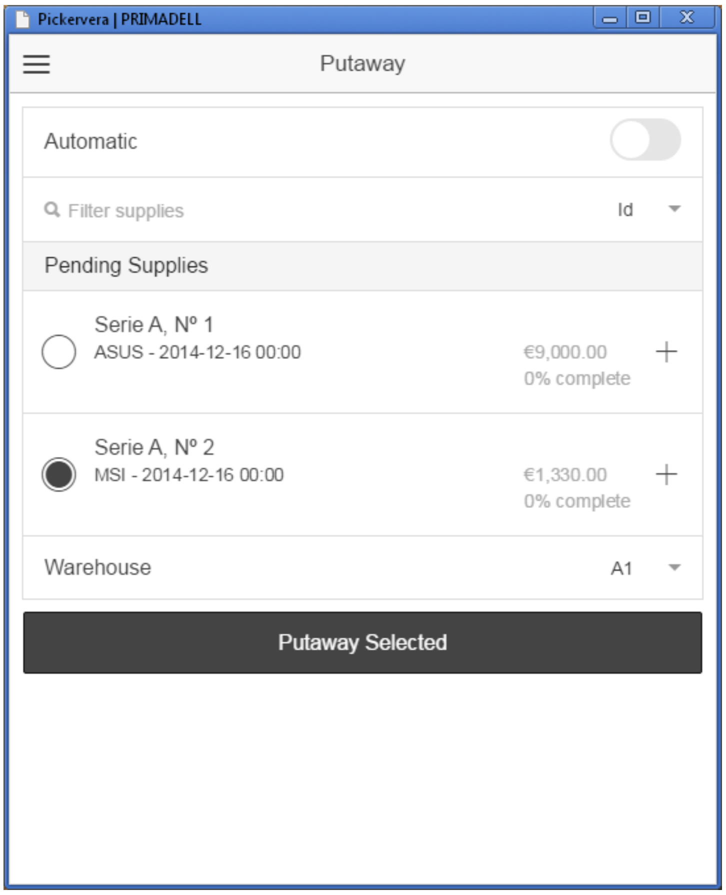


Figure 4: Customer orders user interface

In order to generate a picking list for a set of orders from customers, he must choose the orders to process and press the "Pick Selected" button. After this, the user will be redirected to the Picking list view (see section 6.3).

6.2 UI02 - Supplier orders

The user can see the orders to be picked up from the suppliers in this view.



Serie A, Nº 1
ASUS - 2014-12-16 00:00
€9,000.00
0% complete

Serie A, Nº 2
MSI - 2014-12-16 00:00
€1,330.00
0% complete

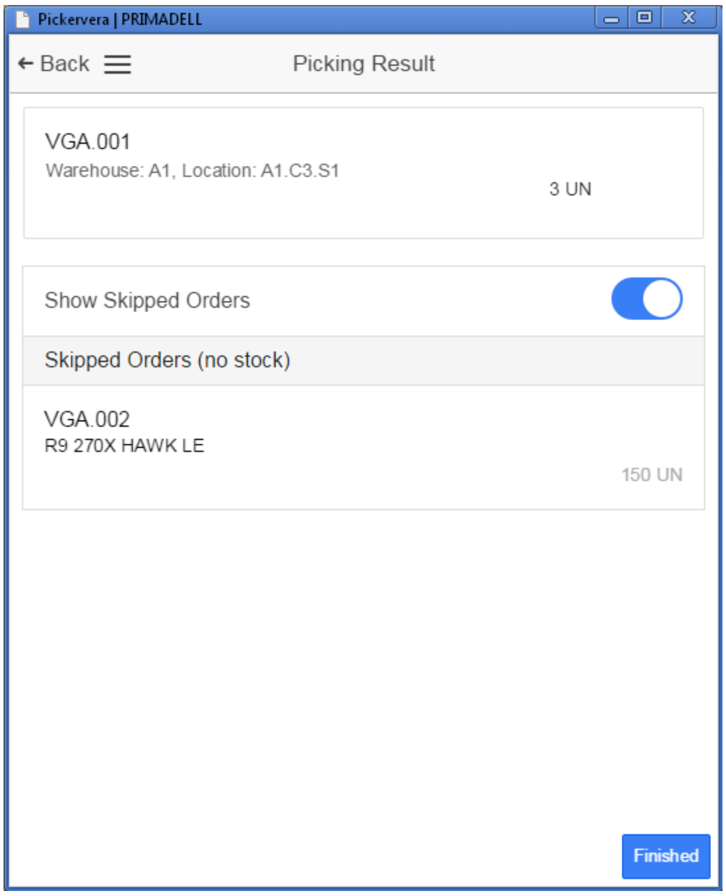
Warehouse A1

Figure 5: Supplier orders user interface

In order to generate a putaway list for a set of orders, he must choose the orders to process and press the ”generate putaway list” button. After this, the user will be redirected to the Putaway list view (see section 6.4).

6.3 UI03 - Picking list

In this view, the user can see the generated picking list for a set of customer orders.



VGA.001
Warehouse: A1, Location: A1.C3.S1
3 UN

Show Skipped Orders

Skipped Orders (no stock)

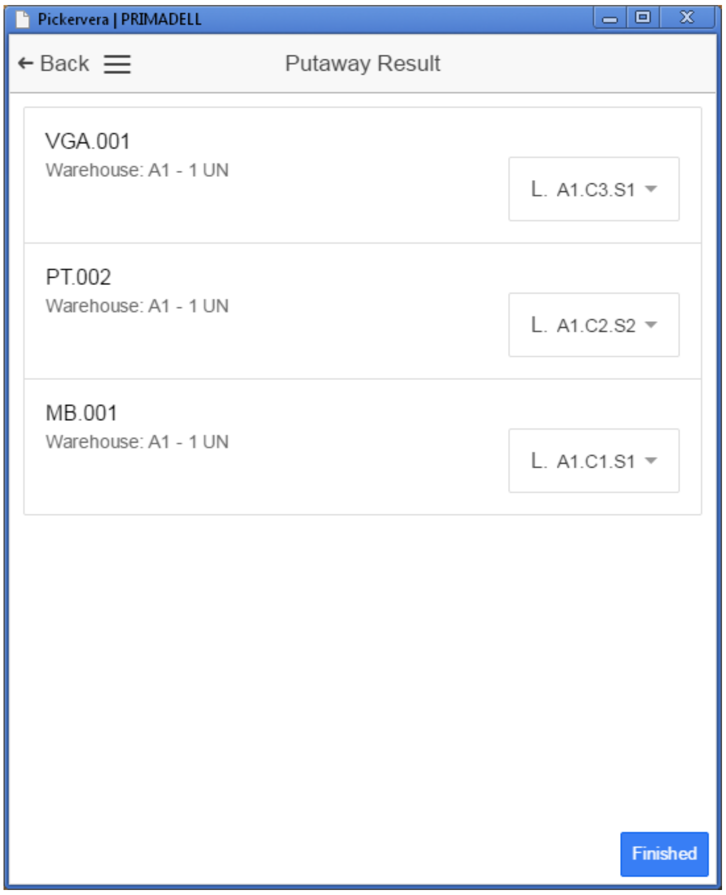
VGA.002
R9 270X HAWK LE
150 UN

Figure 6: Picking list user interface

Before the picking list is marked as finished, he must confirm the picked up quantities for each item in the list. Once all the picking item quantities are confirmed, the picking process is finished and the items are transfered from the warehouse to an output buffer.

6.4 UI04 - Putaway list

In this view, the user can see the generated putaway list for a set of supplier orders.



VGA.001
Warehouse: A1 - 1 UN
L. A1.C3.S1

PT.002
Warehouse: A1 - 1 UN
L. A1.C2.S2

MB.001
Warehouse: A1 - 1 UN
L. A1.C1.S1

Figure 7: Putaway list user interface

Before the putaway list is marked as finished, he must insert the location of the items in the warehouse. Once all the locations are confirmed, the putaway process is finished and the items are transfered into their respective locations.

7 The picking process

7.1 Algorithm

The picking algorithm for a set of orders minimizes the distance that the picker has to travel while picking up and delivering the order items. It follows a greedy approach.

For a set of $\{O_1, O_2, ..., O_k\}$ orders, there is going to be a set of locations where the order items will be picked up. These locations, as their quantity, will be refered to as picking items.

Let $\{p_1, p_2, ..., p_w\}$ be the set of picking items to pick and $dist_{ij}$ be the distance for traveling from the location of p_i to the location of p_j, the goal is to generate a sequence of picking items that minimizes the sum of the distances between the items, that is:

$$\sum_{i,j}^{w} dist_{ij}$$

The order lines are first ordered by quantity. The algorithm then chooses the nearest location of the current picking item as the next place to travel to.

This algorithm does not guarantee the best solution for the picking list generation, but it terminates in a reasonable amount of steps and provides a good solution in most cases.

7.2 User interface scenario

The batch picking process is rather simple with the application, as shown in the diagram below:

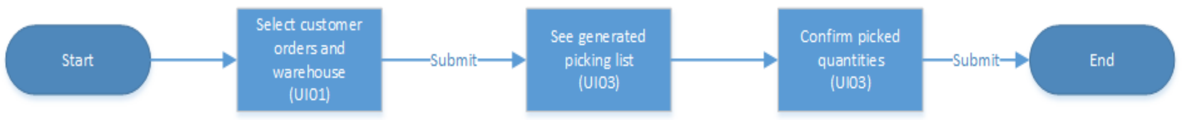

Start
Select customer orders and warehouse (UI01)
Submit
See generated picking list (UI03)
Confirm picked quantities (UI03)
Submit
End

Figure 8: Generate picking list process

7.3 Exceptional scenario

There is a notable exception in the system in the picking process. When the amount of items to be picked in a location exceeds the available stock, a stock removal document is generated (*"documento de saída de stock"*).

8 The putaway process

8.0.1 Algorithm

The algorithm finds a suitable distribution of the order items by the warehouse while minimizing the traveling distance of the person responsible for the putaway.

The algorithm first tries to store the items in a location with the same items already stored. This is made so that there is minimal item dispersion in the warehouse.

If there are no locations with the same item already stored, the closest location is chosen. For two locations with the same distance, the one with the biggest amount of stock will be chosen.

The algorithm repeats this process for all the order items.

8.0.2 User interface scenario

The putaway process for the orders is very similar to the picking process. The process is shown in the diagram below:

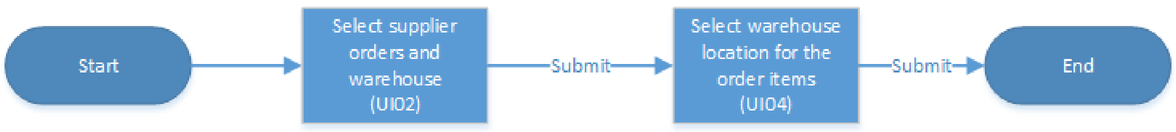

Start
Select supplier orders and warehouse (UI02)
Submit
Select warehouse location for the order items (UI04)
Submit
End

Figure 9: Generate putaway list process

9 Execution instructions and login information

In order to initiate the system, execute the "Pickervera - PRIMADELL" file in the desktop, the server and the client will be started up automatically. If the browser does not open automatically, navigate to *http://localhost/app*.

Login information:

- Username: admin
- Password: admin

10 Notes

10.1 Warehouse configuration

The warehouse locations were split into two dimensions: corridors and sections.

The numbering scheme of the locations is shown in the diagram below:

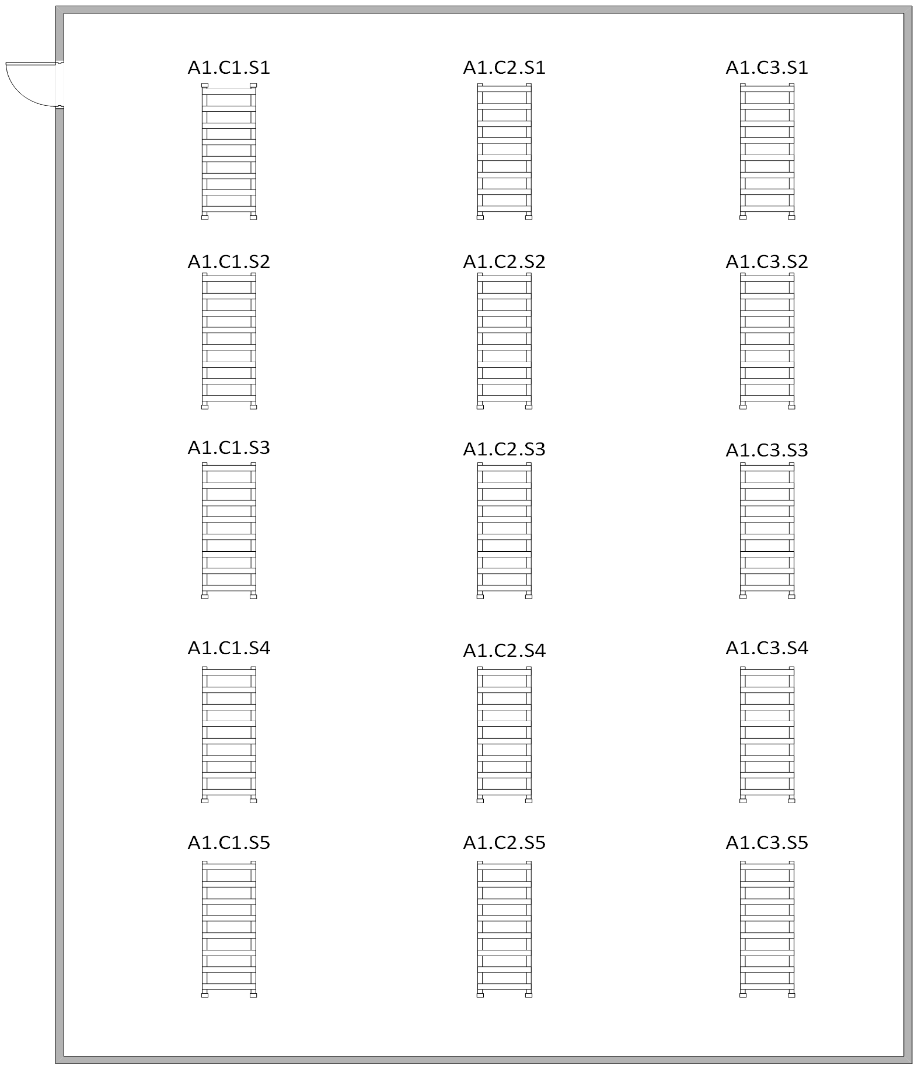


Figure 10: Warehouse location disposal diagram

10.2 Entity relationship diagram

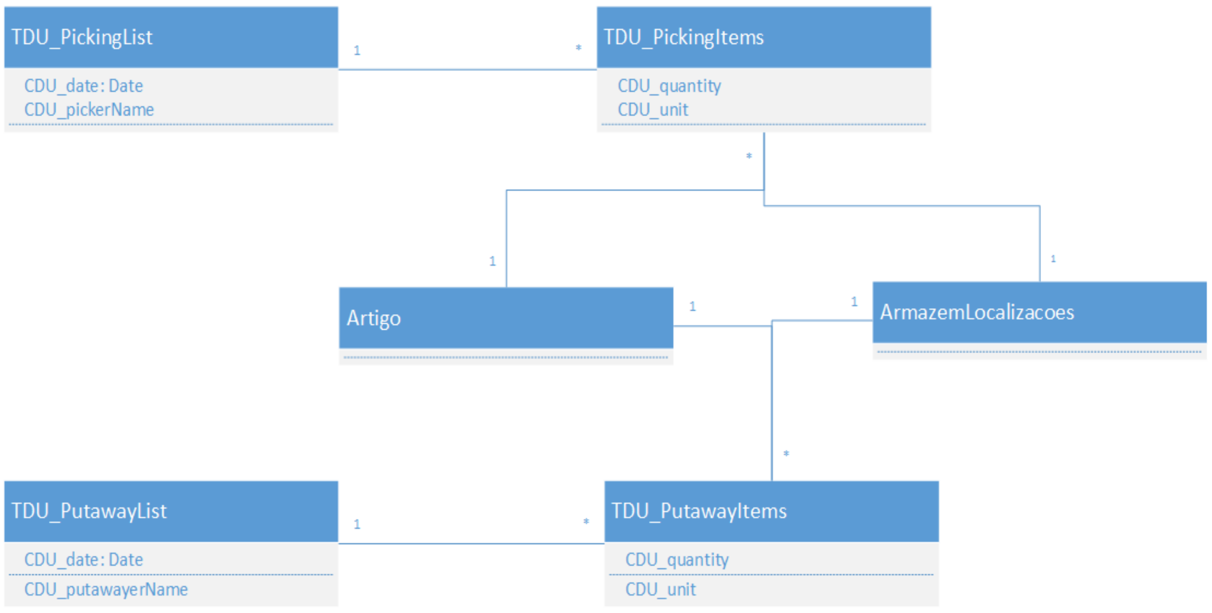


Figure 11: User tables entity relationship diagram