U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Logistics

Final Report

Information Systems

2020/2021

Class 2 - Group AH:

Ana Margarida Ruivo Loureiro - up201705749@fe.up.pt

Carlos Jorge Direito Albuquerque - up201706735@fe.up.pt

João Miguel Ribeiro de Castro Silva Martins - up201707311@fe.up.pt

Tito Alexandre Trindade Griné -up201706732@fe.up.pt

Index

1. Project Overview

1.1. Description

Vicino is a wine company with strong connections to the Douro region and local wine producers. Unlike other wine sellers, Vicino is not a winery, that is, it does not produce wine. Its focus is merely stocking and distributing the best wines of Portugal and the world. With such responsibility, warehouse management is a major factor for the company's success, and therefore it must be done right.

Our project tackles this by integrating with their existing ERP (Jasmin) and extending it with warehouse management capabilities. For that, we developed a webapp focused on presenting the company's orders and stock as frictionless and intuitively as possible together with good algorithms to help warehouse employees replenish and pick inventory faster, and therefore reducing costs.

1.2. Warehouse Description

The warehouse (see image below) is divided into 5 sections, from A0 to A4, each with 4 zones, from A to D. Furthermore, there is a section used for unloading supplier orders (D0) and a section for loading client orders (D1). Every replenishment process starts in the D0 and every picking process ends in the D1.
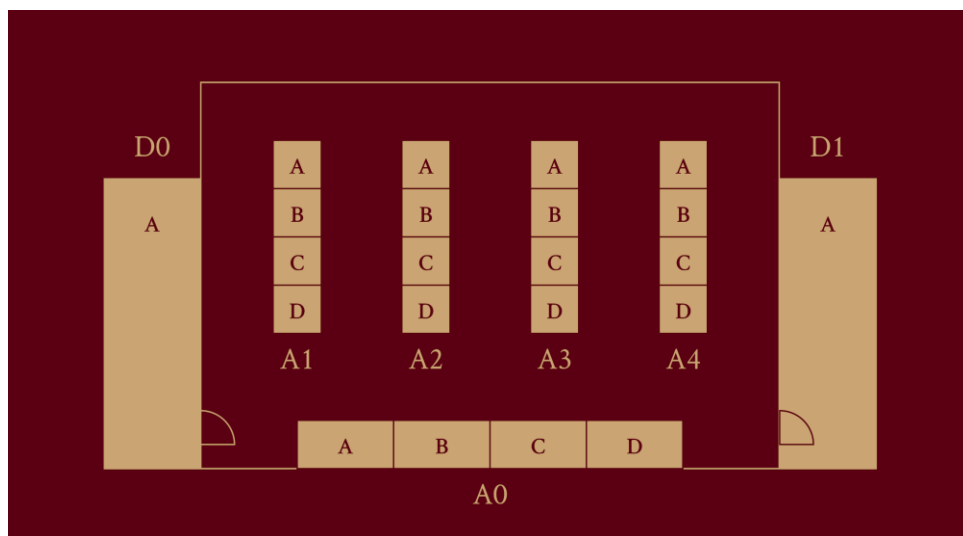

Figure 1: Vicino's warehouse plant

Vicino sells a variety of wines, including some premium, high price wines. Their warehouse distribution follows the pattern presented in the table below. For simplicity reasons and to keep this report small, we are not presenting each individual item Vicino sells.

Section	Items Description
A0	Alentejo Red Wine
A1	Lisbon and Douro Red Wines
A2	White and Dessert Wines
A3	Rosé, Fortified and Sparkling Wines
A4	Premium Wines

Table 1: Vicino's wine distribution by warehouse section

1.3. Picking Algorithm

Regarding the picking process, the warehouse map helps to define the distance between zones for the picking algorithm. In order to maximize the performance of each picking wave, we have defined a greedy algorithm to plan the best route for the employee to pick up the necessary items. The premise is that we are starting our picking from the entry of the warehouse (D0) and go through the required zones, ending in the final stop, which is the warehouse's exit (D1). The rationale to get the best route is to find the closest zone to the current. So, in the start step we go from the warehouse entry and see which zone of the set is closest and move on that, and repeat the process until the last zone, which from there the last stop is the warehouse exit. Programmatically, the distances between zones are mapped in a json file, which is very easy to traverse and calculate the shortest path.

1.4. Sitemap

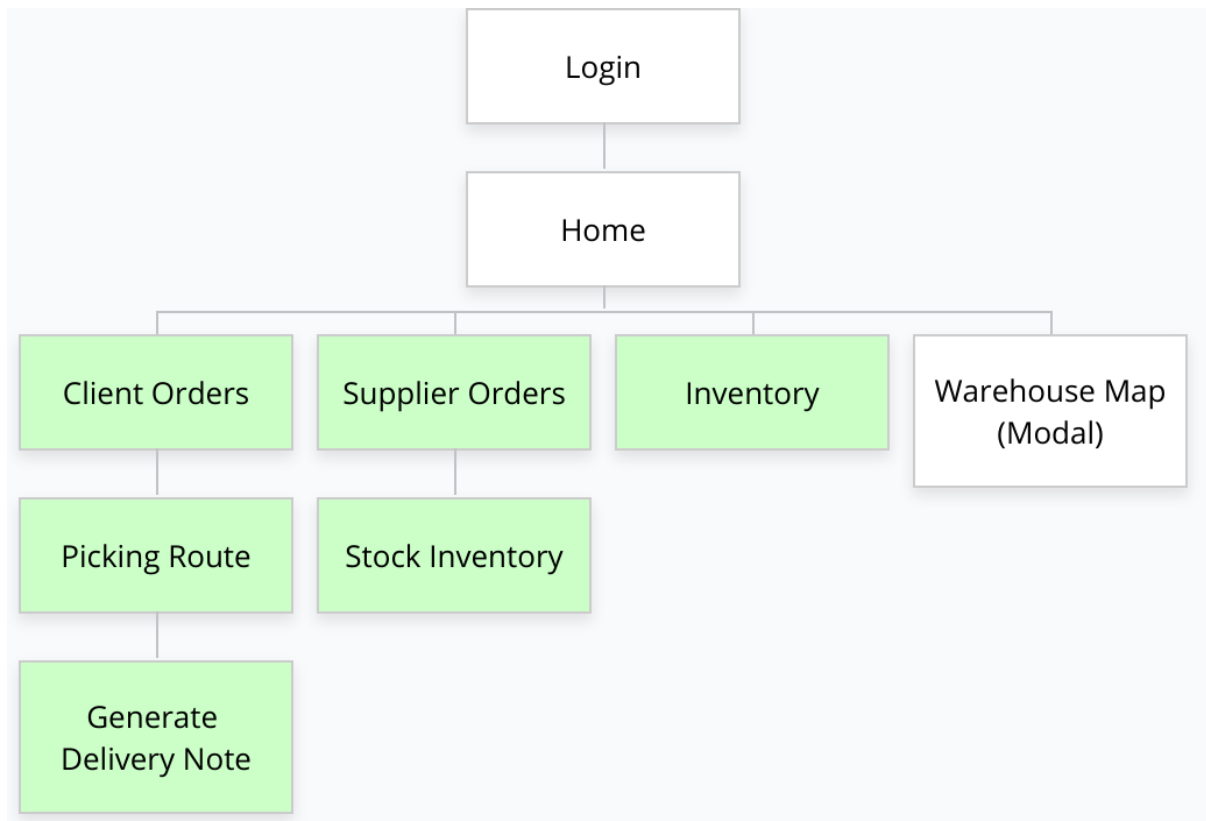The following image represents an overview of our webapp's structure.

Figure 2: Webapp's sitemap with all core (light green) and non-core (white) views.

2. Functionalities

- **Detailed Information**:

  Easy to read information about supplier and client orders as well as existing stock, containing only the essential data for warehouse staff and managers. When appropriate, there is also the possibility to order by relevant information, such as date or client name. There is also a map of the warehouse accessible in every page for easy consultation.

- **Inventory Filtering**:

  Inventory page that allows results filtering by item ID and/or warehouse zone, in order to easily find stock.

- **Generate Delivery Notes**:

  Automatically generate the delivery notes to accompany the shipment of picked inventory.

- **Generate Goods Receipts**:

  Automatically generate goods receipts when replenishing inventory.

- **Partial Stocking**:

  Register partial stock deliveries and shipments while still generating the appropriate documents.

- **Partial Picking**:

  Even after selecting the items and quantity to pick, while on the picking route, if some bottle is missing, in bad conditions or even breaks, it's still possible to pick only part of the selected quantity without impacting the flow or delivery note generation.

- **Picked Stock Transfer**:

 Stock Transfers are automatically generated upon confirming the picking route.

- **Best Picking Routes**:

  Generate the most efficient picking route for expediting orders based on the items selected, using a clean and intuitive interface.

3. BPMN

To better understand where our product would fit in, we developed two BPMNs, which are related, respectively, to the sales process and purchase process. The rationale behind the sales process is that a client initiates the process and the request is handled by the company, going through several steps, which involve warehouse, sales and accounting. The purchase process is similar to the sales, in a reverse direction, and also includes warehouse, sales and accounting.

3.1. Sales Process
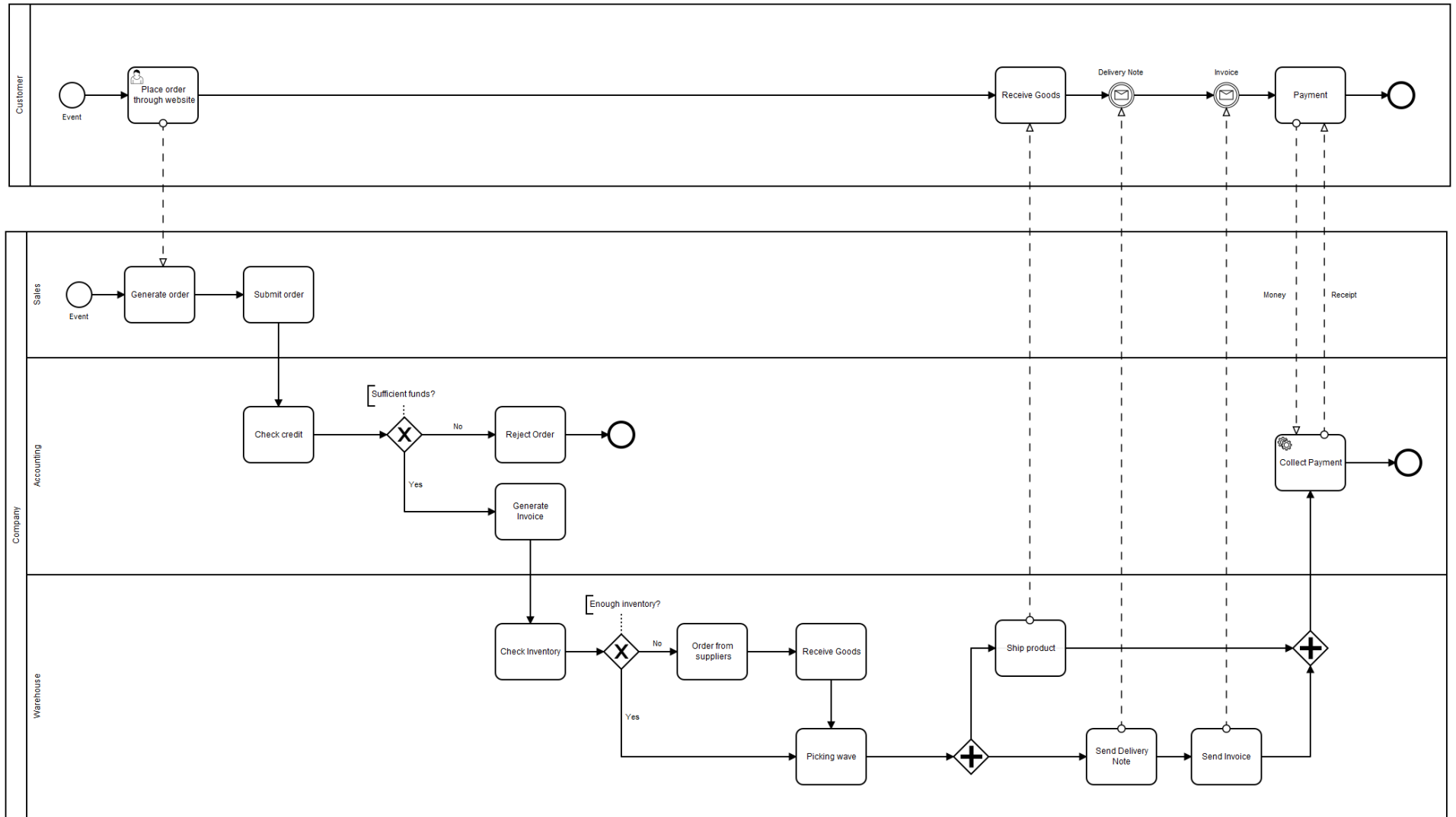
3.2. Purchase Process
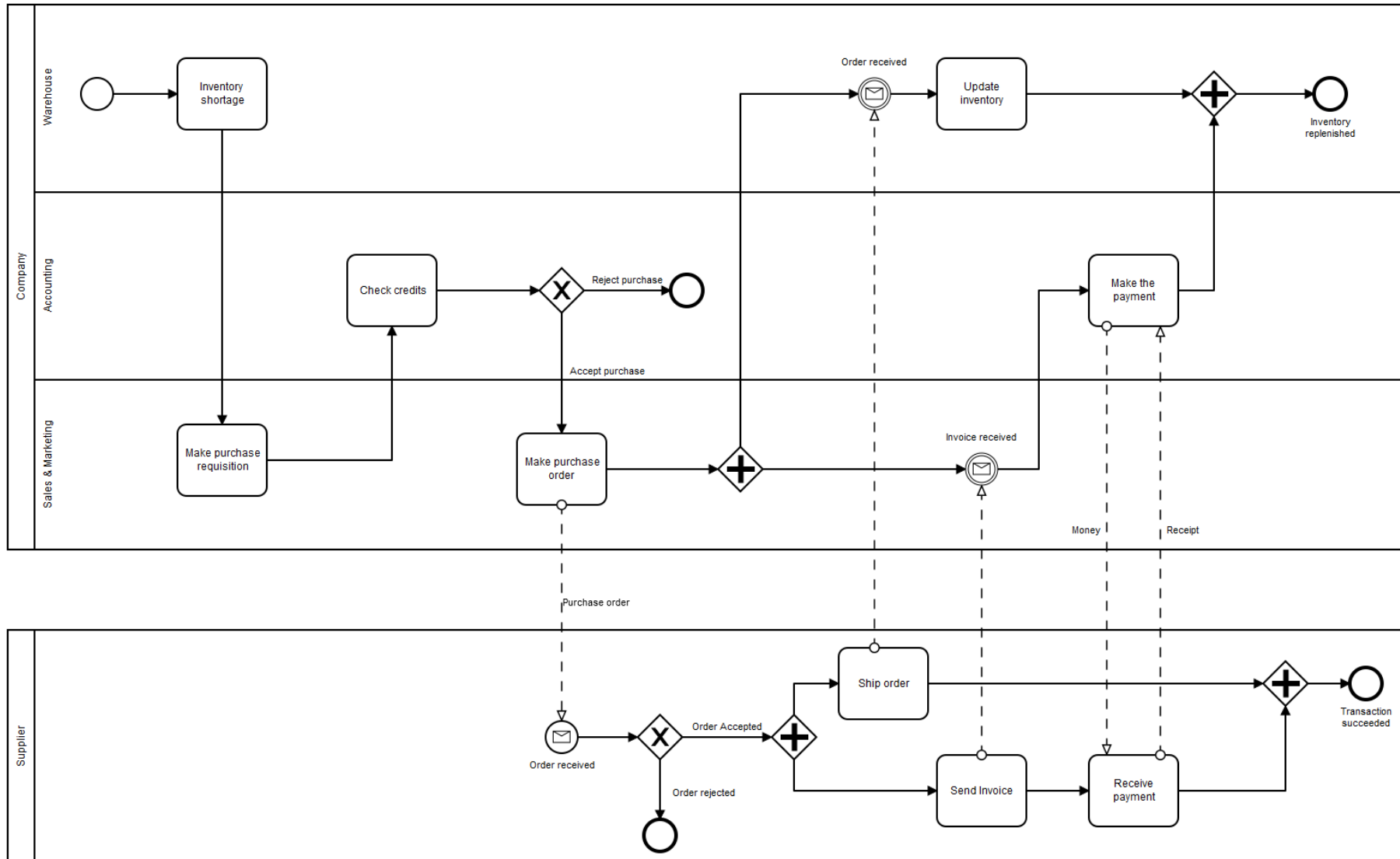
4. Core Views

Core views represent crucial pages where the user is actually interacting with the underlying ERP. This chapter specifies each one following an Information Architecture approach. For each view we define:

- **ID Core View**: a unique identifier for each core view.
- **User & Business Goals**: small description of the core view's main features and value to the user.
- **Inward Paths**: list of ways the user can reach the core view.
- **Outward Paths**: list of ways the user can generate value inside the core view; this value results from interacting with the core view and helps the user achieve his/her goals.
- **Elements of the core**: listing of visual elements that compose the core view; to achieve this we encoded the elements types using the following code:
    - LIST - represents a generic listing;
    - BTN - represents a button;
    - CHK - represents a checkbox;
    - INP - represents a text or number input;
    - DIAG - represents a visual diagram;
- **Project ID**: identifier of the project related to this specification and the date it was last updated.

ID Core View

VICINO_CLI_ORD

User & Business Goals

- Overview of pending sales orders, ordered by date (oldest first)
- Selection of orders to generate new picking waves
- View to be used as a warehouse management tool

Inward Paths

- Homepage
- 'Client Orders' button (second button of navbar)

Outward Paths

- Select items for picking
- Generate picking route based on selected orders ('Generate Route' button)
- Visualize more information about an order (icons on 'More Info' column)

Elements of the core

Project ID

VICINO_IA

19/12/2020

- LIST_001 (LIST) | Client Orders List
- BTN_001 (BTN) | Order Info Button
- CHK_001 (CHK) | Item Selection Checkbox
- INP_001 (INP) | Item Quantity to Pick Input
- BTN_002 (BTN) | Generate Route Button

10

ID Core View

VICINO_SUP_ORD

User & Business Goals

- Overview of pending purchase orders, ordered by date (oldest first)
- Selection of orders to stock in the warehouse
- View to be used as a warehouse management tool

Inward Paths

- Homepage
- 'Supplier Orders' button (first button of navbar)

Outward Paths

- Select items for stocking
- Stock inventory based on selected orders ('Stock Inventory' button)
- Visualize more information about an order (icons on 'More Info' column)

Elements of the core

- LIST_001 (LIST) | Supplier Orders List
- BTN_001 (BTN) | Order Info Button
- CHK_001 (CHK) | Order Selection Checkbox
- INP_001 (INP) | Item Received Quantity Input
- BTN_002 (BTN) | Generate Goods Receipt Button

Project ID

VICINO_IA

19/12/2020

11

ID Core View

VICINO_PICK_ROU

User & Business Goals

- Management of product picking, with partial picking available
- Display of the suggested picking route
- View to be used as a picking management tool

Inward Paths

- 'Client Orders' page

Outward Paths

- Register a product as picked ('Picked' checkbox)
- Choose quantity to pick ('Picked Quantity' field)
- Cancel the picking process and generate no documents ('Cancel' button)
- Check back on previous section picked items ('Back' button)
- Conclude picking in this section, moving on to the next ('Next' button)
- Generate a stock transfer after concluding the picking process, i.e. after confirming the last section picking

Elements of the core

- DIAG_001 (DIAG) | Zones Diagram
- CHK_001 (CHK) | Picked Product Checkbox
- LIST_001 (LIST) | Items List
- INP_001 (INP) | Picked Quantity Input
- BTN_001 (BTN) | Cancel Picking Button
- BTN_002 (BTN) | Previous Section (Back) Button
- BTN_003 (BTN) | Next Section Button

Project ID

VICINO_IA

19/12/2020

12

ID Core View

VICINO_GNRT_DLV

User & Business Goals

- Overview of picked items and picked quantity
- Generate a Delivery Note with the listed items for shipping
- View to be used as the end of the picking process

Inward Paths

- 'Picking Route' page

Outward Paths

- Confirm the results of the picking process
- Generate a Delivery Note on the ERP containing all listed items ('Generate Delivery' button)

Project ID

VICINO_IA

19/12/2020

Elements of the core

- DIAG_001 (DIAG) | Picking Route Diagram
- LIST_001 (LIST) | Picked Items List
- BTN_001 (BTN) | Generate Delivery Button

13

ID Core View

VICINO_STK_IVNT

User & Business Goals

- Interface for inventory replenishment
- Partial stocking available through the 'Stored Quantity' input
- View to be used as a replenishment management tool

Inward Paths

- 'Supplier Orders' page

Outward Paths

- Register a product as stocked ('Stocked' checkbox)
- Insert the stocked quantity ('Stored Quantity' field)
- Conclude inventory replenishment, generating a Goods Receipt on the ERP ('Confirm' button)

Vicino

Stock Inventory

Stocked	Location	ProductId	DocumentId	Ordered Quantity	Stored Quantity
☐	A4C	BOAL	ECF.2020.3	5	Number 2
☐	A3D	VULCANICO	ECF.2020.3	9	Number 6

1-2 of 2   ‹   ›

Confirm

Elements of the core

Project ID

VICINO_IA

19/12/2020

- LIST_001 (LIST) | Items List
- CHK_001 (CHK) | Stocked Product Checkbox
- INP_001 (INP) | Stored Quantity Input
- BTN_001 (BTN) | Confirm Stocking Button

14

ID Core View

VICINO_IVNT

User & Business Goals

- Displays warehouse inventory information
- Search and filter inventory
- View to be used as an inventory management tool

Inward Paths

- Homepage

- 'Inventory' button (third button of the navbar)

Outward Paths

- Search for a specific Product ID ('Product ID' input)
- Filter inventory by zone ('Zone' input)
- Sort by Product ID, Description and Stock
- Filter quantity by zone ('Zone' input in each item row)

Elements of the core

- INP_001 (INP) | Product Search Input
- INP_002 (INP) | Zone Filter Input
- LIST_001 (LIST) | Items List
- INP_003 (INP) | Item Zone Input

5. Other pages

Not every page handles information on the ERP, but all of them are important for the usability of our webapp. As such, this chapter presents non-core views with a simple description and explanation as to why they are relevant to our product.

5.1. Login

- Since every user must be logged in to access warehouse information, this is the entrance page for the system.
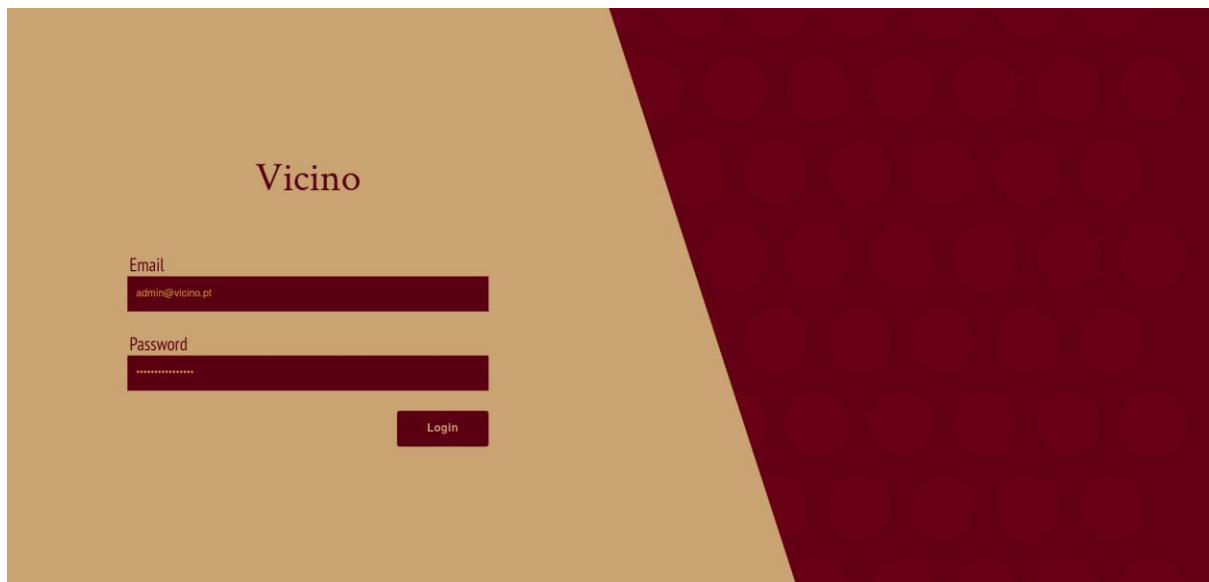- After logging in, the user is redirected to the homepage.

Figure 3: Screenshot of the web app's login page

5.2. Home

- Homepage with all navigation links to access core pages.

- The user can also quickly logout on this page.

Figure 4: Screenshot of the webapp's homepage

5.3. Warehouse Map

- Modal (i.e. a small popup) containing the warehouse map.

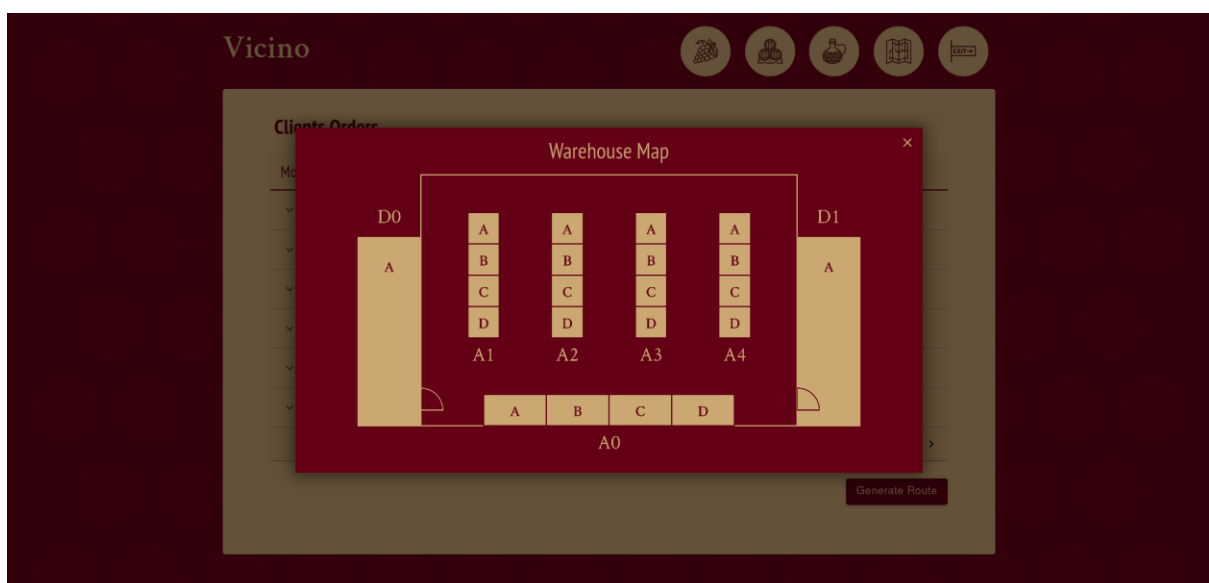- The warehouse map can be accessed anywhere on the webapp.

Figure 5: Screenshot of the webapp's warehouse map modal

6. Gantt Planning

We developed a Gantt chart to help plan our work. This chart includes most of the activities needed to deploy our Jasmin integration. It also includes milestones, which are important dates where we must deliver a product increment.
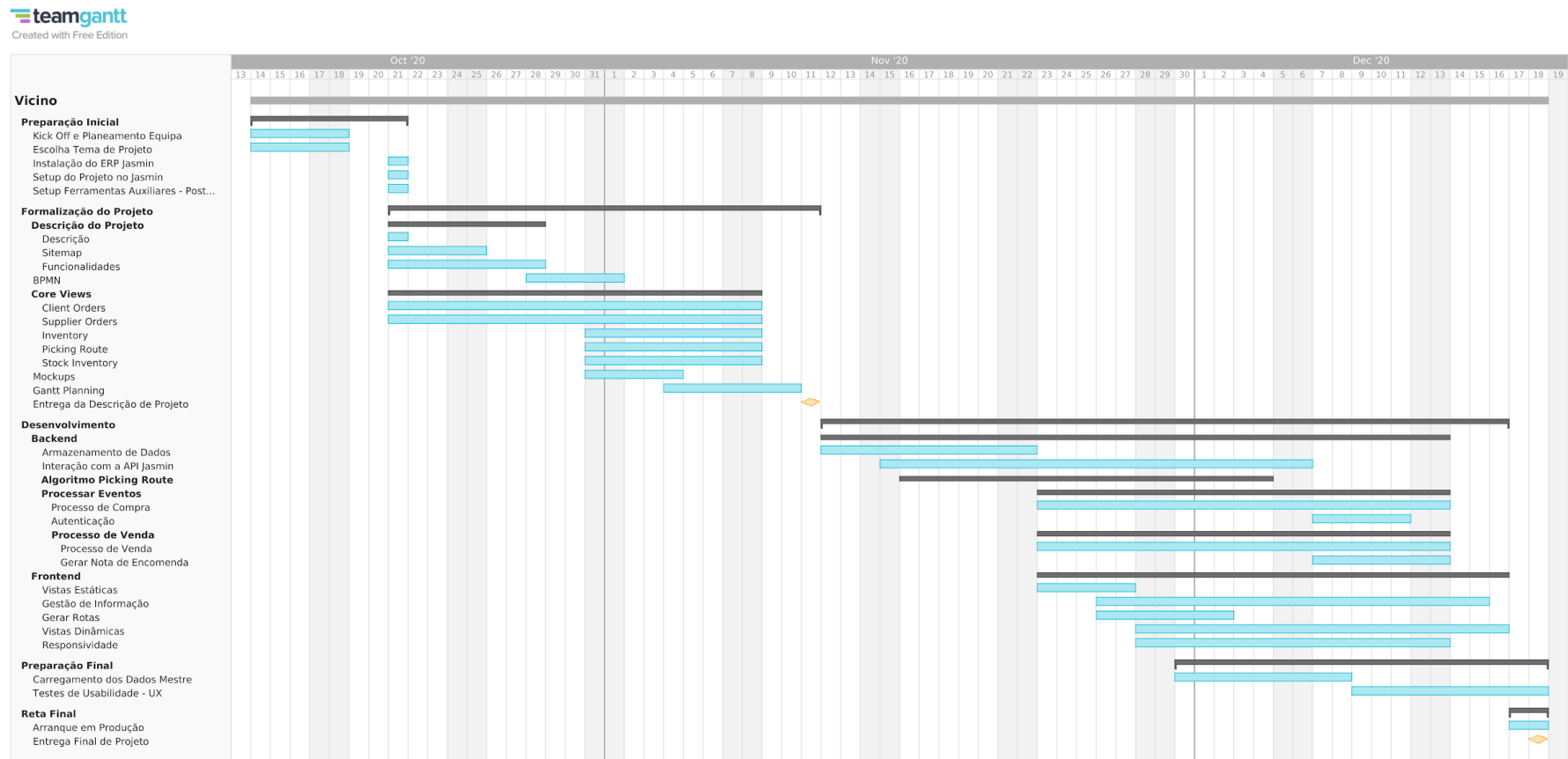

Figure 6: Gantt planning chart

Major Tasks

- Initial preparation:
    - Theme discussion and work planning
    - Installation and setup of the necessary tools for the project's development

- Project description:
    - Definition of the main ideas and features to implement, as well as the application mapping (sitemap)
    - BPMN development for sales and purchases business processes

- Core views:
    - Definition of the main cores to implement

- Mockups:
    - Development of mockups to define the application's design

- Gantt Planning:
    - Planification of the finished and future work to maintain the organization of the project

- Backend:
    - Setup of a database to persist necessary data related to the picking waves
    - Routes setup and calls to the Jasmin API to make the necessary operations work
    - Picking route algorithm implementation
    - Process events triggered by the user, related to replenishment and picking

- Frontend:
    - Implementation of the views (pages) and handling the application behaviour regarding the interactions with the user
    - Manage the information presentation

- Final preparation:
  - Load data related to the existent transactions
  - Usability testing
- Last Mile:
  - Final arrangements before the project delivery

Milestones

For the project there were two milestones for the development, a more conceptual one and another more practical, which culminates with the final delivery of this project.

The first was the delivery of the project description, which set in stone the project theme and features to implement. The first delivery happened on the November 11th.

The second and final, was the project delivery, after finishing the project main features and testing. The last delivery happened on the December 20th (2 days after initially planned).

7. Final Considerations

This project enabled us to understand in more detail and explore some functionalities of an enterprise resource planning system like the Jasmin Software. Thus, we were able to work and explore how the software fulfils the logistics operations and better understand the importance of an application like ours that reduces the complexity and focuses on a particular purpose, allowing it to have additional functionalities. Furthermore, the implementation of an inventory picking algorithm to optimize the route a warehouse worker has to go through to collect each item turned out to be a very interesting part of the project and an excellent example of classical programming algorithms applied to real world problems.

All in all, we are satisfied with the knowledge and experience we were able to retain among the used technologies, as well as all the concepts associated with information systems. However, we experienced some challenges with the Jasmin documentation, that proved to be incomplete and confusing in some instances, which made our development process slower and a bit frustrating.