

LINFO2402

Open Source Project : Odoo Carpooling

AUGUST, 2022



Farnoodian, Nima - 68372000

Professors : Yves Deville and Olivier Bonaventure

Academic Year : 2021-2022

Odoo is a collection of tools for managing businesses, such as CRM, e-commerce, billing, accounting, manufacturing, warehouse, project management, and inventory management. The GNU LGPLv3 is the license for the Community edition of the free software. Odoo's flexible architecture enables a considerable number of independent contractors and businesses to create Odoo Apps or Modules and offer them for sale or free download in the marketplace. The framework consists of core apps (aka modules), and thousands of community modules, making up the primary parts of Odoo. The goal of this open source project, as agreed in the OpenWeek at UCLouvain, is to build a Carpooling app that allows the Odoo users to share their trips for free. In the following sections, the app features are discussed.

My journey to Odoo began during the OpenWeek 2022 at UCLouvain when my team and I were assigned to build an Odoo App. The defined project was Carpooling app that could be installed on the Odoo framework. Odoo Team suggested the project during OpenWeek since the Odoo employees had already utilized simple excel sheets to carpool. Although The team could manage to build the app during OpenWeek, due to the short developing time, the final carpooling app was elementary, covering only simple user needs, and lacked necessary features such as commenting, complete reservation system, necessary trip information, complex queries, and views, etc. The related repository can be found here ¹. Thus, upon an agreement with Prof. Olivier Bonaventure on the first day of Openweek, I decided to extend the project and enrich it with more features. However, the presented work is different in a number of respects as I re-engineered everything and accomplished the project from scratch. It is worth mentioning that a few functions implemented in the present work were adopted from our OpenWeek project, thus commented in the code accordingly.

2.1 Carpooling App Repository

In order to build an Odoo app and contribute to the Odoo repository, two ways have been suggested by Odoo employees during the OpenWeek :

- Forking the repository, installing Odoo as stated here ², and contributing by creating the Odoo app or website.
- Installing Odoo with Docker, creating an individual repository for the new app, and mounting the app within the Odoo container. The complete documentation for the official Odoo docker image can be found on odoo - Official Image | Docker Hub ³.

As suggested by the Odoo employees, in order to obtain a compatible and correct installation and avoid any installation errors that Windows OS may cause, the docker installation approach was chosen, and the Carpooling app was individually implemented in this repository ⁴.

2.2 Learning

In order to learn Odoo and how to build an Odoo app, I completed the first 13 chapters of Odoo Developer training ⁵ (Core training except Chapter 14) meant for Odoo employees. According to this training guideline, this core training is expected to be completed within a month. This core training explains incrementally how to build a real estate app that allows users to sell and buy properties. Although the training walks the learners through each step of the app designing, it does not provide any solution as it expects the learner to learn Odoo development through investigating others' examples and playing with them. However, these trials took me around a month, and the first 15 commits in the associated repository ⁶ belong to the training process. This training was complete enough for

1. <https://github.com/hlibiouille/OpenWeek-odoo-carpooling>

2. https://www.odoo.com/documentation/15.0/developer/howtos/rdtraining/02_setup.html

3. https://hub.docker.com/_/odoo/

4. <https://github.com/nima-farnoodian/odoo-carpooling>

5. <https://www.odoo.com/documentation/15.0/developer.html>

6. <https://github.com/nima-farnoodian/odoo-carpooling>

preparation for the actual contribution. It is worth saying that since the repo contains the real estate app, it could also stand as a valuable source for those people wishing to complete the training.

3.1 Technical Overview

3.1.1 Backend

In Odoo, an app (module) consists of several business objects declared as Python classes extending Model, which integrates them into the automated persistence system. Models can be configured by setting attributes in their definition. Each model is indeed translated into Database table and the Odoo ORM layer is responsible for this translation. Contrary to Carpooling app developed during OpenWeek, the presented Carpooling app consists of one inherited model and four models which are as follows :

- `res.user` : `res.user` is an existing model in Odoo that stores the Odoo user information. In order to hold the driver information for carpooling, such as Car Name, Car Image, etc., `res.user` has been extended to store the carpooling-related information once the user installs the carpooling app. In this case, the passengers can access more details regarding the driver when booking a trip.
- `car.pooling` : This model is responsible for storing the trip information such as Driver, Trip Date, Source City, Destination City, etc. Using this model, a driver can add a trip, and passengers can book the trip accordingly.
- `Car.pooling.tag` : Each Trip can be associated with tags such as No-smoking. Thus, this model has been proposed to let the drivers add tags that may represent the summarized trip information.
- `Car.pooling.passenger` : As a passenger, you may choose a trip and would like to book that trip. However, a key point here is that the driver should be able to refuse or accept a booking request. Therefore, for a trip, there might be many book request which needs attention from the driver. To this end, this model has been suggested to achieve this goal because it works like a reservation system but for carpooling.
- `Car.pooling.comment` : As a passenger, you may add some comments for a trip after departure or before departure when the booking request is accepted. This model aims to facilitate commenting on the trips.

A model does not only contain simple attributes. It can consist of several functions for computing dependent values, adding constraints, doing some actions when clicking a button on the frontend, etc. For all models above, actions, functions, etc., have been carefully defined to support automation, user interactions, and so on. The functions' behaviors have been tested using a regression test to validate the app's functionality.

3.1.2 Frontend

Odoo framework generally cares about the Frontend and User Interface (UI). However, in order to link a model to a UI and let the user interact with the models, the view should be defined with respect to xml format. Accordingly, several UIs, such as trip UI and User UI, were designed to support the app's functionality. In Carpooling Directory¹, the “views” folder holds the designed UIs. For example,

1. <https://github.com/nima-farnoodian/odoo-carpooling/tree/main/carpooling>

trips_view.xml is responsible for showing the list of trips, trip details as a form once the trip is clicked, and Searching abilities, which allows the user as driver or passenger to search, filter, or group the trips according to some inputs. In the User Guide section, the user experience will be discussed.

3.2 User Guide

3.2.1 App Installation

After installing Odoo under the assumption that it is installed using Docker, the app should be mounted within the Odoo container as stated here². It is worth noting that the app can be added even without Docker if the whole repository has been cloned and installed. In this case, the *car_pooling* directory should be copied and pasted on the addons directory. However, once the app is mounted and the server runs, the carpooling app can appear on the app page and installed, as shown in Fig 3.1.

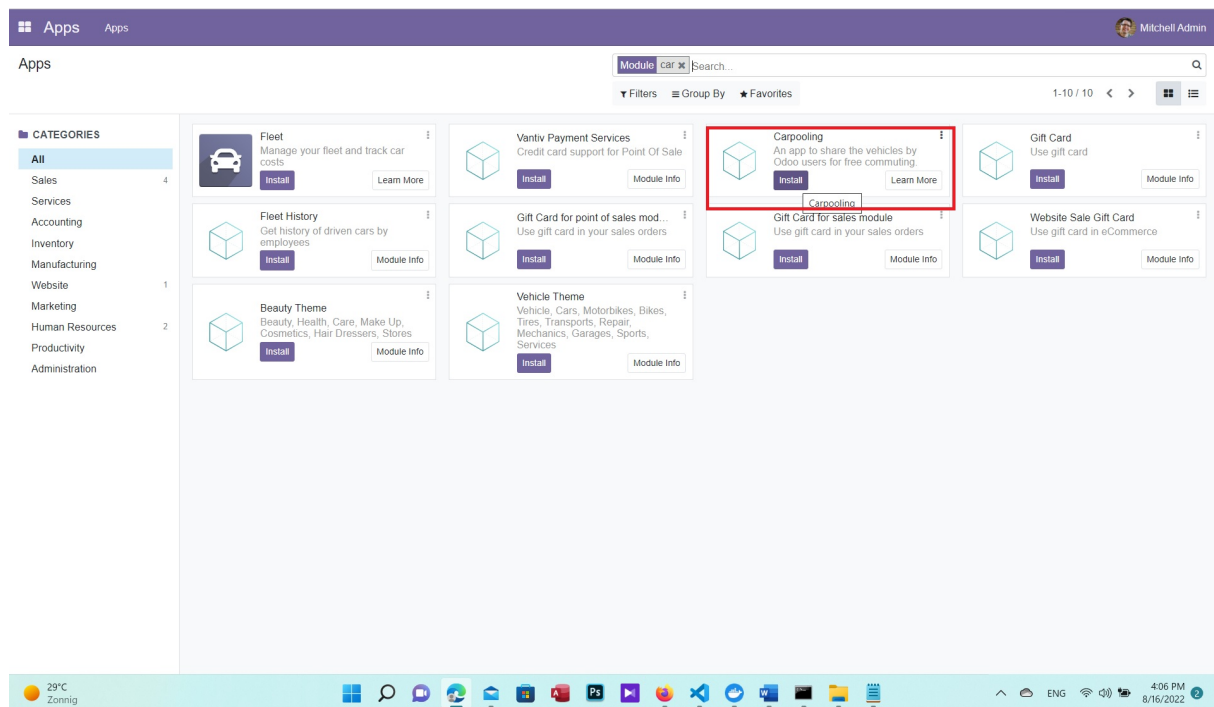


FIGURE 3.1 – Carpooling app appearing after searching car.

3.2.2 User Information Completion

After app installation, three tabs automatically appear on the user's page. First, the Vehicle Information form should be filled out by the user. This information is then automatically added to trip information when the user volunteers to share his/her vehicle. The other tabs, "My trips as drive" and "My Booked trips as Passenger" represent the information of the trips the user has participated in as either driver or passenger. The user can then see, for example, the status of a trip he has booked in "My Booked trips as Passenger".

2. https://hub.docker.com/_/odoo/

The screenshot displays the 'Users / Mitchell Admin' settings page. The 'Vehicle Information' tab is active, showing fields for Car Name (BMW), Vehicle Model (ix1), Vehicle Type (SUV), Vehicle plate Number (24gg165), and Vehicle Color (a light blue circle). There is also an 'Upload Vehicle Image' section with a small car image. The top navigation bar includes 'Settings' and 'Users & Companies'. The bottom status bar shows the temperature as 29°C and the date as 8/16/2022.

FIGURE 3.2 – User Interface modified after Carpooling Installation

3.2.3 Share a Trip

In the carpooling app, the user can add a trip to share it with others. Once pressing the “Create” button, a form appears, which can receive the trip information such as Source City, Date, Tags, Driver’s Comment, etc. Notice that the vehicle information is automatically filled out according to the user’s information already inserted in the user form. Moreover, the driver can see the passenger’s comment even though s/he cannot modify the comments. Once saving the trip, using the same form structure, all the information except the list of passengers and passenger comments will be shown to anyone wishing to book the trip. Fig 3.3 shows an example of the trip form filled out by a driver, and Fig 3.4 represents an example of a trip form shown to a user who booked the trip while the booking status is still “Undecided”.

It is mentionable that on the top right corner of the form, the trip status is shown as a status bar. The status of the trip is automatically changed (except “Departed” and “Canceled” status) based on the available seats, passenger booking status, and the date of the trip. A user can only book a trip if it is in the “Available” status, which means the trip still has available seats and it will depart in the future.

Once the trip receives one booking request, the driver can decide to accept the booking or refuse it through his/her user’s form, in “My trips as drive” tab, or in the trip form shown in Fig 3.4. The Available seat number is reduced by one if the booking is accepted. Moreover, the user can be aware of the decision both on the trip form, “Booking Status”, or in his/her user form, in the “My Booked trips as Passenger” as shown in Fig 3.6.

Carpooling Trips

Car pooling: Trips / New

Save Discard

Departed Cancel Available Full Unavailable Departed Canceled

From Louvain-la-Neuve to Antwerp

Driver (Car owner) Mitchell Admin

Number of seats 2

Available seats 2

Round Trip ☒

Trip Information Vehicle Information Driver Comments Tags Passenger Passenger Comments

Departure Date and Time 08/16/2022 16:29:03

Source City Louvain-la-Neuve

Source Address Rue de La gare

Destination City Antwerp

Destination Address Central

Return Date and Time 08/17/2022 16:29:03

29°C Zonnig

4:41 PM 8/16/2022

FIGURE 3.3 – A trip form shown to a driver. Moreover, on the top left corner, there are two buttons for putting the trip status in "Departed" or "Canceled" status. These two buttons are only visible to the driver.

Carpooling Trips

Car pooling: Trips / From Louvain-la-Neuve to Antwerp

Edit Create Action

1 / 1 < >

Available Full Unavailable Departed Canceled

From Louvain-la-Neuve to Antwerp

Driver (Car owner) Mitchell Admin

Booking status Undecided

Number of seats 2

Available seats 2

Round Trip ☒

Trip Information Vehicle Information Driver Comments Tags

Departure Date and Time 08/16/2022 16:29:03

Source City Louvain-la-Neuve

Source Address Rue de La gare

Destination City Antwerp

Destination Address Central

Return Date and Time 08/17/2022 16:29:03

29°C Zonnig

4:44 PM 8/16/2022

FIGURE 3.4 – A trip form shown to a passenger. Passenger Comments and Passenger List are invisible. Once the booking request is accepted, Passenger Comments will be visible to the passenger.

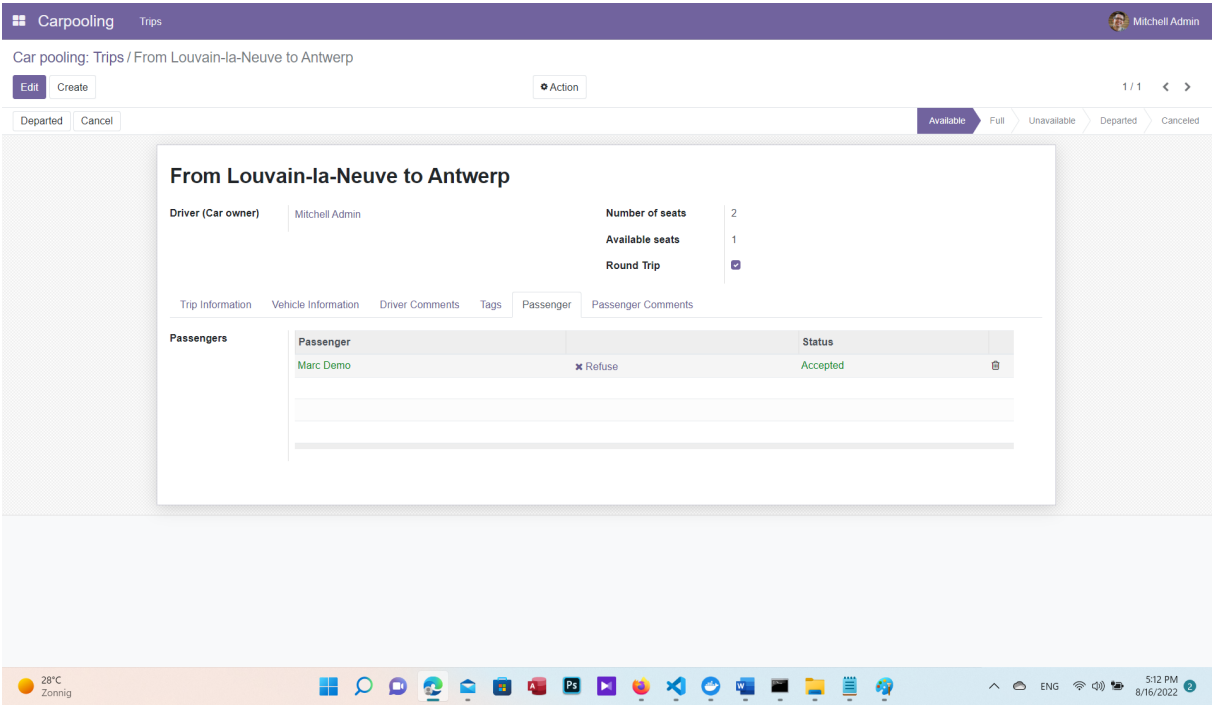


FIGURE 3.5 – The figure shows that a booking request received from Marc and the request was accepted by the Driver.

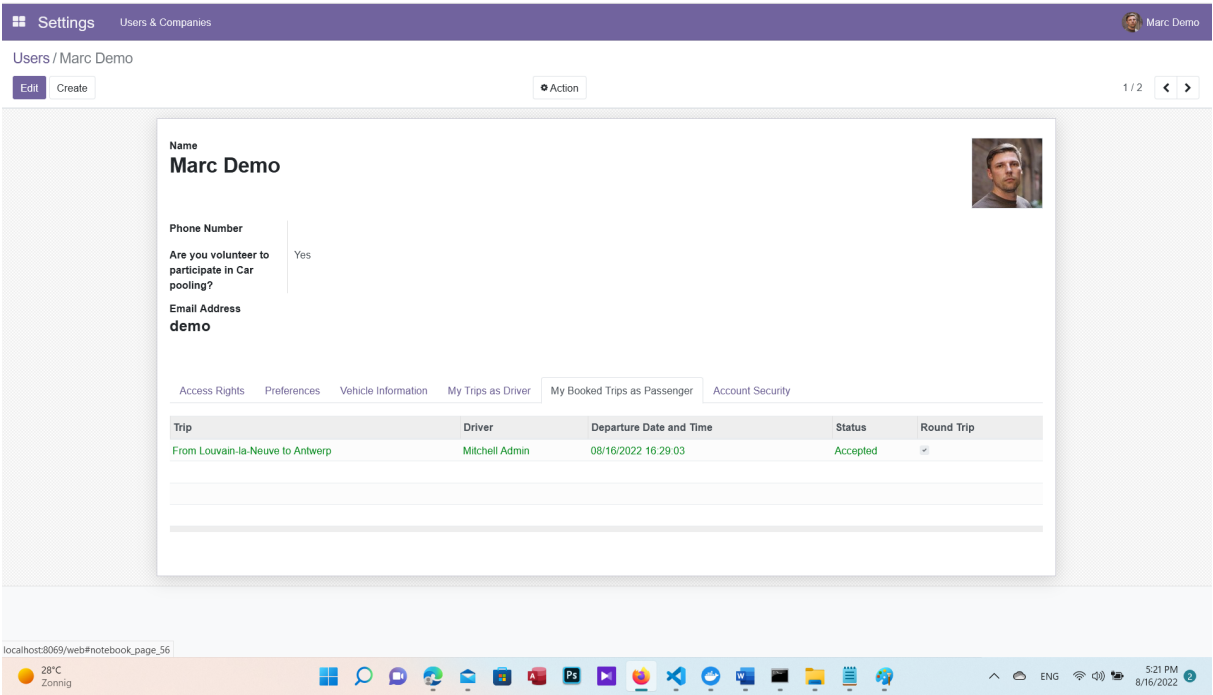


FIGURE 3.6 – The figure illustrates the booking status of a passenger in his/her user’s form. Accepted requests are decorated by green color and the refused requests are decorated by red color.

3.2.4 Trip Search

One key feature is searchability, which empowers the users as drivers or passengers to search for their desirable trips (e.g., trips from Louvain-la-Neuve) or filter out the trips according to some criteria. To this end, different possible searching abilities have been added to the app. To point out some important ones, filtering by the trip status or grouping by the source or destination cities are worth saying. The users can thus find their wishing trips by combining different searches (depart time and round trip),

filtering, and grouping. Fig 3.7 shows an example of a trip list view along with available grouping — though the users are able to manually add their own filtering and grouping.

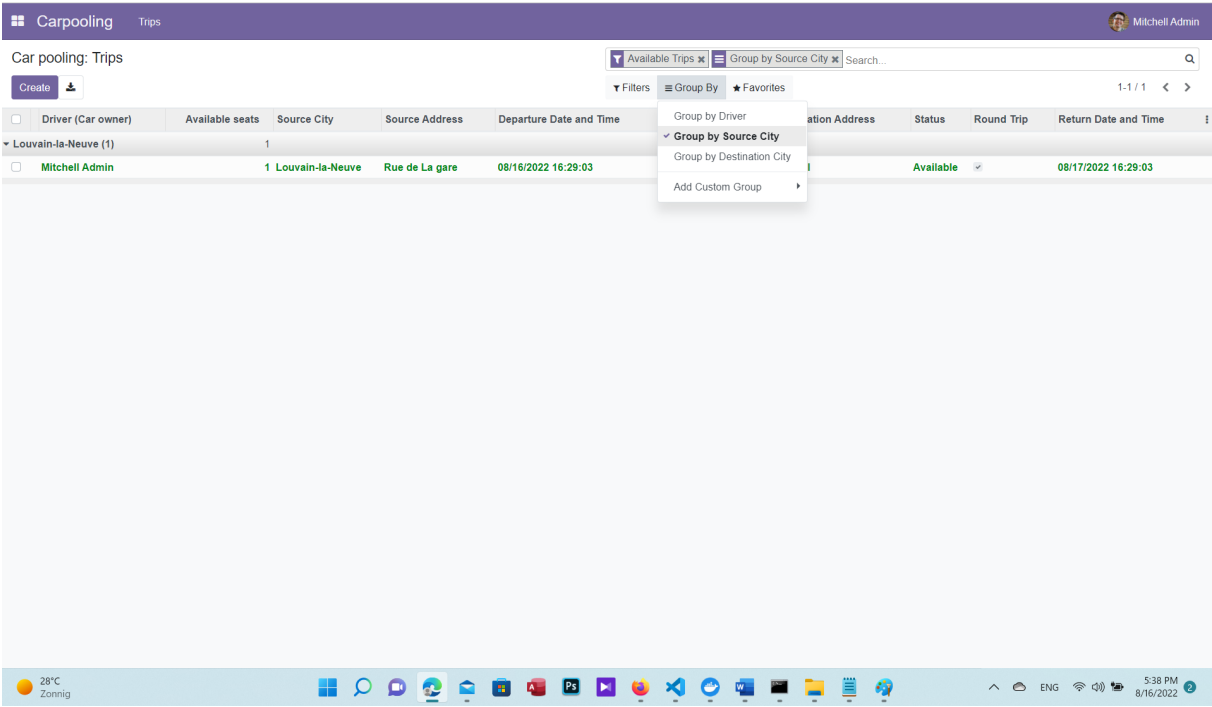


FIGURE 3.7 – An example of searching and grouping the trips

Carpooling is nowadays a way people use to share their transportation costs or to reduce fuel consumption and traffic. In order to provide a concert and safe solution for carpooling, an Odoo app has been proposed in this project, which can be used and installed on the Odoo platform as stated earlier. It has been attempted to design the app to cover most users' needs. Compared to the very first version of the app, which was proposed during the OpenWeek at UCLouvain, this app is accompanied by many features such as a more complex booking system, commenting, more views, and even more automation. It is no exaggeration to say that the app has been attempted to be designed in such a way that it could comply with the Odoo standard, and many things have been accomplished behind the scene to ensure the app's functionality.