THE PECULIARITIES OF OPTIMAL GROUPAGE CARGO ARRANGEMENT IN A SEMITRAILER ALGORITHM

DOI: 10.36724/2072-8735-2023-17-6-54-60

Manuscript received 24 April 2023;
Accepted 17 May 2023

Sabeena M. Isajeva,
FSAEI HE Russian university of transport, Academy of basic training, Moscow, Russia, ***isaeva-sabina@yandex.ru***

Egor A. Shobanov,
FSAEI HE Russian university of transport, Academy "High engineering school", Moscow, Russia,
egorco.sa@gmail.com

Keywords: *python; shipments; cartesian coordinate system; designing; cargo; semitrailer*

This article dwells upon the peculiarities of optimal groupage cargo arrangement in a semitrailer algorithm. Nowadays it is quite important to optimize logistics processes due to high demand of this kind of service. The author describes a developing and designing process of special application that automatically arranges cargos in a semitrailer. The text gives valuable information referred to a semitrailer structure and logistics. It is spoken in detail about eight steps explaining the whole process of development. Much attention is given to one particular step explaining application operation in stages. Such an application is rather user friendly, so any logistics manager can use it without instructions. The solution is applicable to real use case with some final improvements towards compatibility with other systems and also this solution helps in speeding up some workflows in logistics services, so some funds can be saved. The exact amount of savings can differ from company to company due to individual internal structure. Summing up the results, it can be concluded that the solution is rather multipurpose, in other terms it can be applicable to multiple companies with some refinements due to specific company's requests.

Information about authors:
Vitaly G. Dovbnya, *Doctor of Technical Sciences, Associate Professor, Professor of the Department of Space Instrumentation and Communication Systems, Southwestern State University, Kursk, Russia*
Dmitry S. Koptev, *Senior Lecturer, Department of Space Instrumentation and Communication Systems, Southwest State University, Kursk, Russia*
Leon Rea Herman Floresmilo, *2nd year master student, Department of Space Instrumentation and Communication Systems, Southwestern State University, Kursk, Russia*
Georgy I. Podkhaldin, *2nd year undergraduate student of the Computer Engineering Department, Southwestern State University, Kursk, Russia*

Для цитирования:
Исаева С.М., Шобанов Е.А. Особенности разработки алгоритма визуализации оптимальной расстановки груза в грузовом полуприцепе // Т-Comm: Телекоммуникации и транспорт. 2023. Том 17. №6. С. 54-60.

For citation:
Isajeva S.M., Shobanov E.A. The peculiarities of optimal groupage cargo arrangement in a semitrailer algorithm. *T-Comm*, vol. 17, no.6, pp. 54-60. (*in Russian*)

Introduction

In the modern world exists the problem of arranging cargos inside semi-trailer, which has always been done by humans. This approach can take a lot of time and results may not satisfy cargo company. Some similar systems exist nowadays, but these systems are made by special IT companies exclusively. In fact, some systems can upload some arrangement tips to worker's tablet [3]. However, this tips and arrangement suggestions are not useful enough and there are no calculation methods to finally make 3D model – the real applicable thing.

Problem statement

AP GRUPP logistics* has been in need of a software product for visualization of cargo arrangement in the car body and automatic formation of shipments according to the destinations specified in the cargo parameters. This product will be used by a shipment manager.

Company manager has put forward the following demands:

- the form of input and output should be based on database tables. (Output for checking);
- visualization output as future possibility;
- editing of the existing dispatches;
- an indentation included in generated tables output;
- editing the dispatches to enter the approved date.

Analysis of initial data

For the implementation of the project, the AP GRUPP logistics company obtained the initial data containing the consecutive consignments with their parameters (weight, length, width, height, price of shipment and etc.) [5].

On the analysis there have been agreed some important details such as:

1) There is a departure and arrival point in the cargo parameters, so it is convenient for the customer to ship cargoes that have departure points within a 100 km radius, as well as arrival points. However, arrival points can similarly be located within a 100 km radius of any point on a straight line connecting the centers of the two circles within which most departure and arrival points are located. The diagram of the delivery area described in this paragraph looks like this (Fig. 1):

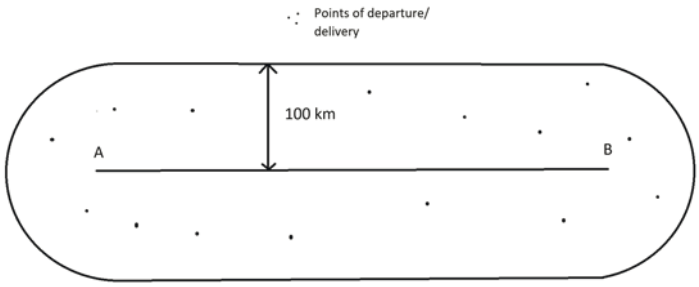


Fig. 1. Operating scheme

2) Some shipments may have a priority or some benefit for the company, so the manager should be able to add the packages to the shipment, even if the arrival point of the shipment is not defined by the program as intermediate.

Designing and creating an algorithm

The algorithm development is rather simple. Due to the employer requirements, a block diagram has been made (Fig. 2):

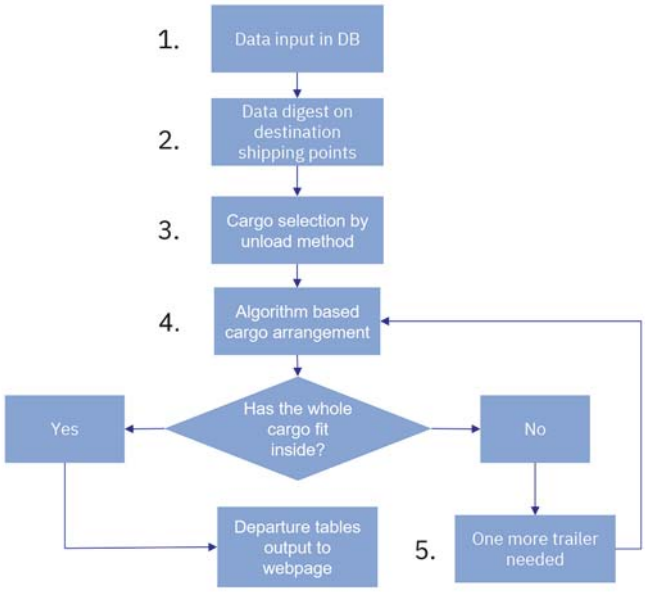


Fig. 2. Operation of application

Diagram construction is crucially important in project development, because designer needs to know how exactly he should connect all modules to make product as much holistic as it possible. For example, dividing algorithm into separate functions, known as the application is leading to ease in application development [9].

The 1st step in creating the algorithm was to form classes that store information about cargo. Within the Python algorithm, it is much more convenient to refer to cargo parameters if classes are used, for example, this variable will contain the weight of some cargo "link_to_cargo_object.weight". The ".weight" is a method of the class which is called Cargo within the project.

The designer has developed a function that allows you to get the parameters of cargos from the database and assign them to the methods of the class.

The 2nd step algorithm arranges packages. Creating a specific mathematical model of the cargo and the body would lead to slow operation of the algorithm because of the large number of packages to be arranged. This is the reason why there has been developed an algorithm based on the Cartesian coordinate system and simple logic, in which each rectangular cargo in two-dimensional space (top view) will have a corner, the coordinates of which will determine the occupied space in the body (Fig. 3).

* The new faculty of Russian university of transport – the Academy of high engineering school has given its students several projects to solve within project activity academic study. One of them has been suggested by one of the Academy's partners – AP GROUP logistics company.

Steps of the function, presented on the following picture:

1) The packages are divided into arrays by points of departure and arrival.

2) A query to the database is created, which results in all the cargo data already in the "cursor" variable.

3) The library "geopy" is used, to replace the points of departure and arrival by coordinates, simply in a way that "google maps" do, and the names of cities are stored in the array. This is done to avoid quiring "API geopy" for coordinates during each iteration, when measuring distances between two cities, since we need to measure the distance from each city to the other, i.e., it would be more complicated because of queries.

4) The loop "while" takes one city from the array and searches for all cities within a radius of 100 km, and they are all sent in an array of one shipment to the "result" array, and are also removed from the "cursor" array, since the shipment is generated.

5) The next function takes the resulting array of shipments and tries to arrange packages from one shipment. The shipments are sorted by unloading method and dimensions, first by unloading, then, if there are similar unloading methods, they are sorted by dimensions.

6) The first package is placed at the point (0, 0) of the body.

7) The function "weather_free" kicks in, it takes an already placed package as an initial and checks if there is space next to it for a new one. It also checks for possible staging restrictions, i.e. if something is to the right of the underlying cargo, it is still possible to put something that will fit between them.

```
1 egors +1
def get_route_inf():  # extract all information from DB to class Cargo by route
    start_time = time.time()
    cursor = db_auth()
    cursor.execute('select * from Test_table')
    cursor = list(cursor)

    towns = {}

    for i in cursor:
        towns[i[0]] = (i[3], i[4])
        i[3] = get_lat_lot(i[3])
        i[4] = get_lat_lot(i[4])

    result = []

    while len(cursor) > 0:
        cargo = []
        pop_list = []
        current = (cursor[0][3], cursor[0][4])

        cargo.append(Cargo(cursor[0][0], list(map(float, [cursor[0][7], cursor[0][8], cursor[0][9]])), cursor[0][1],
                           cursor[0][2], cursor[0][3], cursor[0][4], float(cursor[0][5]), cursor[0][12], cursor[0][14],
                           cursor[0][15], cursor[0][6], cursor[0][10]))
        cursor.pop(0)
        for i in cursor:
            if great_circle(i[3], current[0]).kilometers <= 100 and great_circle(i[4], current[1]).kilometers <= 100:
                for _ in range(i[6]):
                    cargo.append(Cargo(i[0], list(map(float, [i[7], i[8], i[9]])), i[1], i[2], i[3], i[4], float(i[5]),
                                       i[12], i[14], i[15], i[6], i[10]))
                pop_list.append(i)
        [cursor.pop(cursor.index(s)) for s in pop_list]

        result.append([s for s in cargo])

    for table in result:
        for i in table:
            i.route = towns[i.index][0]
            i.end_route = towns[i.index][1]

    total_time = time.time() - start_time
    print(total_time)
    return result
```

Fig. 3. Algorithm code

In the end, the information about the space around the current cargo (whether it is free and some restrictions) is written into variables and the cargo is placed anywhere with no semi-trailer area restrictions.

The algorithm is repeated until the body is full.

```
# define if there is an overlapping in any direction
free_right = whether_free(placed, 'right', placed_cargo, truck_body)
free_left = whether_free(placed, 'left', placed_cargo, truck_body)
free_top = whether_free(placed, 'top', placed_cargo, truck_body)
```

Fig. 4. Using functions

Technologies employed

In algorithm development the following programming languages has been used (Table 1):

Programming languages	Python modules	Other
• Python • SQL • HTML • CSS	• Flask • Pyodbc • SQLAlchemy • Flask-SSLify • Cryptography • Geopy	• HTTPS certification • Port forwarding • Domain registration • Web designing

Table 1. Applied technologies

UI and Backend development

A web-interface was developed for user interaction with the system which should achieve maximum convenience for user, who is working with it.

The core of the website is the Python Flask library, which allows you to create a "web" server. Using it you can link the visual part of the site with the logical part, the algorithm in this case [7].

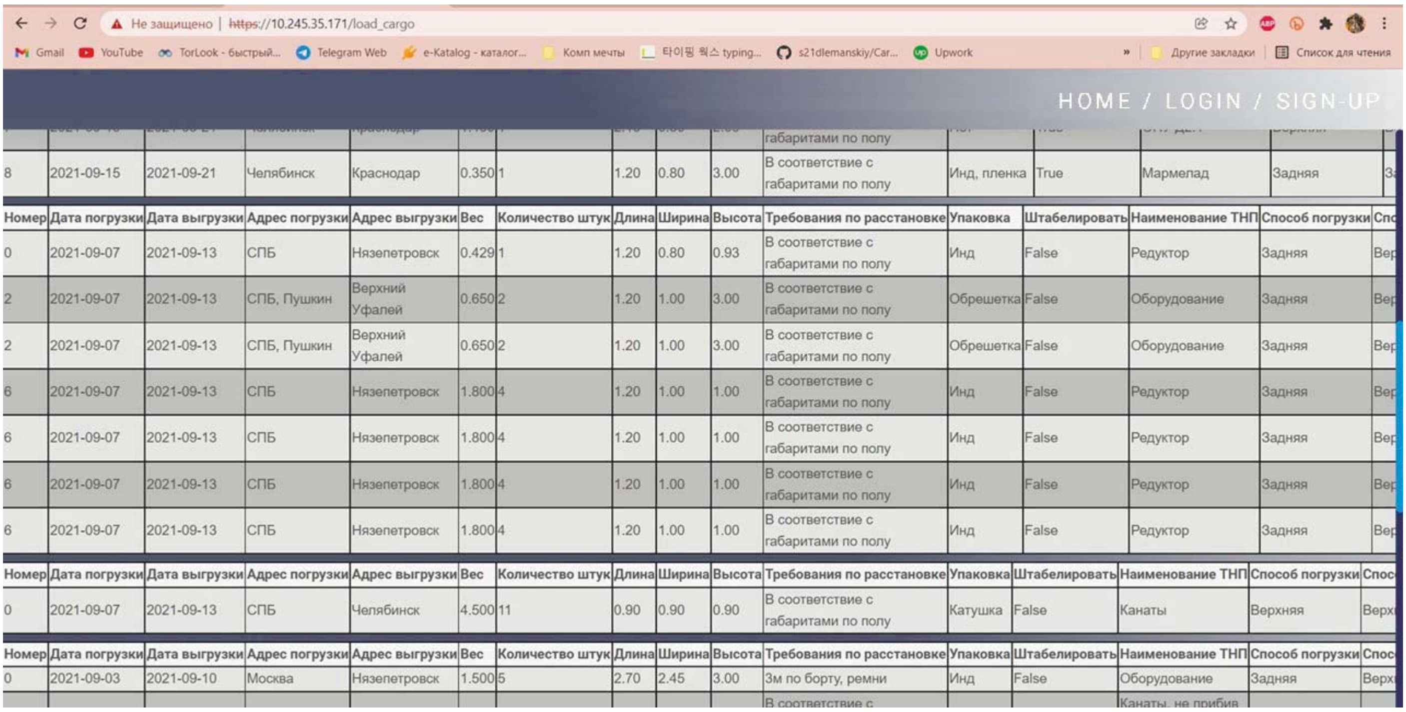
Developing a site – is primarily the creation of folders with different parts (page templates, different modules and main processing structures Flask).

The 3rd step is to create a master page template, it contains the basic HTML objects that are needed in the structure of the page and the objects that are seen when switching from page to page:

1. Navigation bar
2. Scroll bar

Actually, it is possible to use Python code in the HTML code to create generic objects several times or to pass variables to the screen. The plugin jinja2 is made for this; this code allows you to shape "notification" objects to a typical form and make several notifications at once if, for example, several parameters of the entered login and password are wrong. The code of the other pages uses the code of the main page through the same jinja2 plugin [4].

The last step of creating export page is to send variables. On the picture there is table headers which have been loaded in the variable "headings", in the "data" data and then they have been transferred to the HTML code, which already displays everything on the site.



8	2021-09-15	2021-09-21	Челябинск	Краснодар	0.350	1	1.20	0.80	3.00	В соответствие с габаритами по полу	Инд, пленка	True	Мармелад	Задняя	За
Номер	**Дата погрузки**	**Дата выгрузки**	**Адрес погрузки**	**Адрес выгрузки**	**Вес**	**Количество штук**	**Длина**	**Ширина**	**Высота**	**Требования по расстановке**	**Упаковка**	**Штабелировать**	**Наименование ТНП**	**Способ погрузки**	**Спо**
0	2021-09-07	2021-09-13	СПБ	Нязепетровск	0.429	1	1.20	0.80	0.93	В соответствие с габаритами по полу	Инд	False	Редуктор	Задняя	Вер
2	2021-09-07	2021-09-13	СПБ, Пушкин	Верхний Уфалей	0.650	2	1.20	1.00	3.00	В соответствие с габаритами по полу	Обрешетка	False	Оборудование	Задняя	Вер
2	2021-09-07	2021-09-13	СПБ, Пушкин	Верхний Уфалей	0.650	2	1.20	1.00	3.00	В соответствие с габаритами по полу	Обрешетка	False	Оборудование	Задняя	Вер
6	2021-09-07	2021-09-13	СПБ	Нязепетровск	1.800	4	1.20	1.00	1.00	В соответствие с габаритами по полу	Инд	False	Редуктор	Задняя	Вер
6	2021-09-07	2021-09-13	СПБ	Нязепетровск	1.800	4	1.20	1.00	1.00	В соответствие с габаритами по полу	Инд	False	Редуктор	Задняя	Вер
6	2021-09-07	2021-09-13	СПБ	Нязепетровск	1.800	4	1.20	1.00	1.00	В соответствие с габаритами по полу	Инд	False	Редуктор	Задняя	Вер
6	2021-09-07	2021-09-13	СПБ	Нязепетровск	1.800	4	1.20	1.00	1.00	В соответствие с габаритами по полу	Инд	False	Редуктор	Задняя	Вер
Номер	**Дата погрузки**	**Дата выгрузки**	**Адрес погрузки**	**Адрес выгрузки**	**Вес**	**Количество штук**	**Длина**	**Ширина**	**Высота**	**Требования по расстановке**	**Упаковка**	**Штабелировать**	**Наименование ТНП**	**Способ погрузки**	**Спос**
0	2021-09-07	2021-09-13	СПБ	Челябинск	4.500	11	0.90	0.90	0.90	В соответствие с габаритами по полу	Катушка	False	Канаты	Верхняя	Верх
Номер	**Дата погрузки**	**Дата выгрузки**	**Адрес погрузки**	**Адрес выгрузки**	**Вес**	**Количество штук**	**Длина**	**Ширина**	**Высота**	**Требования по расстановке**	**Упаковка**	**Штабелировать**	**Наименование ТНП**	**Способ погрузки**	**Спос**
0	2021-09-03	2021-09-10	Москва	Нязепетровск	1.500	5	2.70	2.45	3.00	3м по борту, ремни	Инд	False	Оборудование	Задняя	Верх

Table 2. Application table

```
def export():
    cursor = db_auth()
    cursor.execute('select * from Test_table')

    headings = ('Номер', 'Дата погрузки', 'Дата выгрузки', 'Адрес погрузки', 'Адрес выгрузки', 'Вес', 'Количество штук',
                'Длина', 'Ширина', 'Высота', 'Требования по расстановке',
                'Упаковка', 'Штабелировать', 'Наименование ТНП', 'Способ погрузки', 'Способ выгрузки',
                'Стоимость доставки, НДС')
    data = tuple([s for s in cursor])

    for i in range(len(data)):
        data[i][0] = i + 1
        data[i][1] = str(data[i][1])
        data[i][2] = str(data[i][2])
        if data[i][12]:
            data[i][12] = 'Да'
        else:
            data[i][12] = 'Нет'

    dates = dict.fromkeys(['Все'] + [f'{str(s[1])} - {str(s[2])}' for s in data])

    return render_template('export.html', user=current_user, headings=headings, data=data, dates=dates)
```

Fig. 5. Export function

Another challenging task was to implement the sort by date function on the data table page, it was created using Java Script code and embedded into the HTML code as a script. It was difficult to deal with a new programming language, but eventually everything has worked out, and now the function works perfectly.

The next step is designing; visual objects have been created, then they need to be shaped and styled. CSS code is responsible for styling. It is required to refer to HTML classes and change parameters: indentation, text style, color, coordinates and others.

The homepage style is also used on all other pages, it is referenced in the homepage HTML code, the repeating object – background.

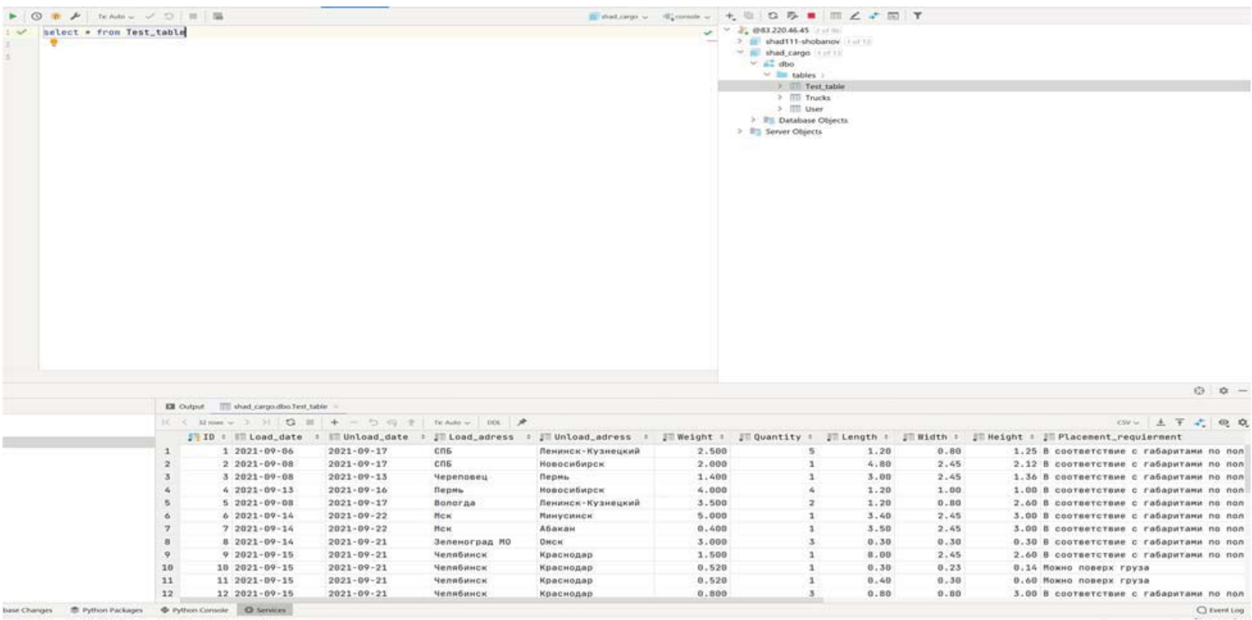
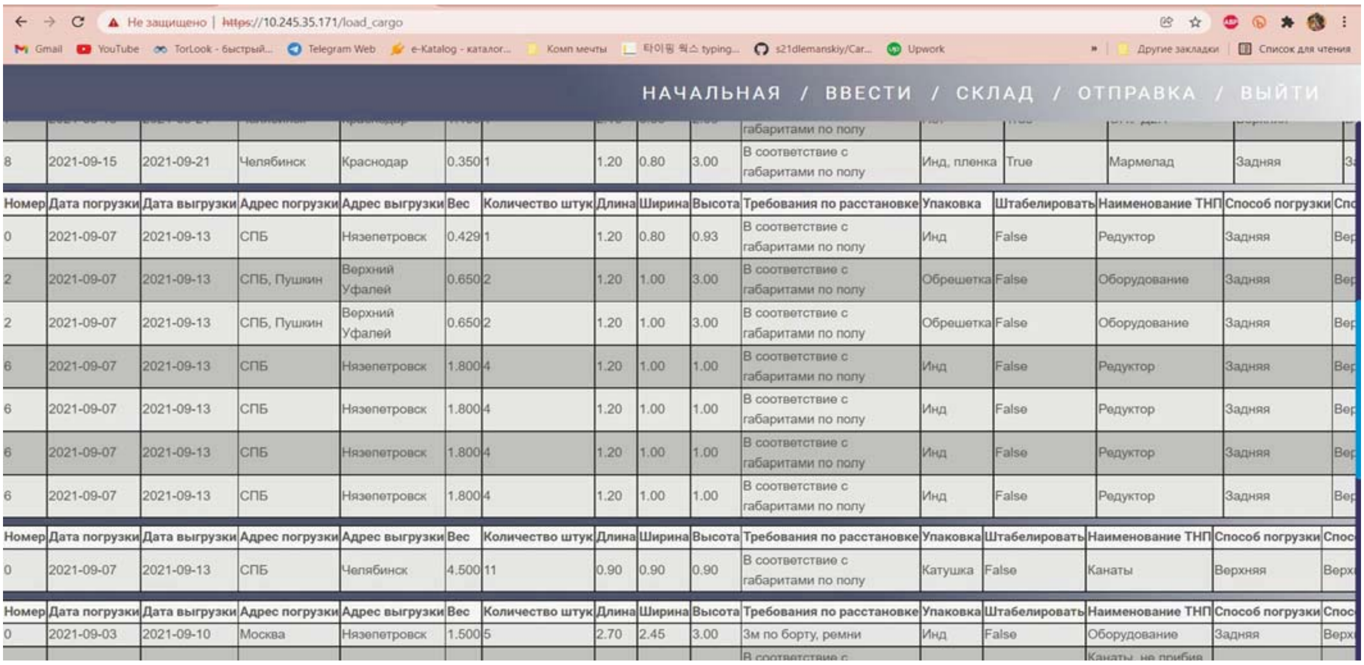


Fig. 6. Database usage



8	2021-09-15	2021-09-21	Челябинск	Краснодар	0.350	1	1.20	0.80	3.00	В соответствие с габаритами по полу	Инд, пленка	True	Мармелад	Задняя	З

Номер	Дата погрузки	Дата выгрузки	Адрес погрузки	Адрес выгрузки	Вес	Количество штук	Длина	Ширина	Высота	Требования по расстановке	Упаковка	Штабелировать	Наименование ТНП	Способ погрузки	Спо
0	2021-09-07	2021-09-13	СПБ	Нязепетровск	0.429	1	1.20	0.80	0.93	В соответствие с габаритами по полу	Инд	False	Редуктор	Задняя	Вер
2	2021-09-07	2021-09-13	СПБ, Пушкин	Верхний Уфалей	0.650	2	1.20	1.00	3.00	В соответствие с габаритами по полу	Обрешетка	False	Оборудование	Задняя	Вер
2	2021-09-07	2021-09-13	СПБ, Пушкин	Верхний Уфалей	0.650	2	1.20	1.00	3.00	В соответствие с габаритами по полу	Обрешетка	False	Оборудование	Задняя	Вер
6	2021-09-07	2021-09-13	СПБ	Нязепетровск	1.800	4	1.20	1.00	1.00	В соответствие с габаритами по полу	Инд	False	Редуктор	Задняя	Вер
6	2021-09-07	2021-09-13	СПБ	Нязепетровск	1.800	4	1.20	1.00	1.00	В соответствие с габаритами по полу	Инд	False	Редуктор	Задняя	Вер
6	2021-09-07	2021-09-13	СПБ	Нязепетровск	1.800	4	1.20	1.00	1.00	В соответствие с габаритами по полу	Инд	False	Редуктор	Задняя	Вер
6	2021-09-07	2021-09-13	СПБ	Нязепетровск	1.800	4	1.20	1.00	1.00	В соответствие с габаритами по полу	Инд	False	Редуктор	Задняя	Вер

Номер	Дата погрузки	Дата выгрузки	Адрес погрузки	Адрес выгрузки	Вес	Количество штук	Длина	Ширина	Высота	Требования по расстановке	Упаковка	Штабелировать	Наименование ТНП	Способ погрузки	Спос
0	2021-09-07	2021-09-13	СПБ	Челябинск	4.500	11	0.90	0.90	0.90	В соответствие с габаритами по полу	Катушка	False	Канаты	Верхняя	Верх

Номер	Дата погрузки	Дата выгрузки	Адрес погрузки	Адрес выгрузки	Вес	Количество штук	Длина	Ширина	Высота	Требования по расстановке	Упаковка	Штабелировать	Наименование ТНП	Способ погрузки	Спос
0	2021-09-03	2021-09-10	Москва	Нязепетровск	1.500	5	2.70	2.45	3.00	3м по борту, ремни	Инд	False	Оборудование	Задняя	Верх

Fig 7. Application output

Database development

Database requirements were low, so the designer needed to store information about cargo and a table with users and occasionally refer to them to display data in the application. It was decided to use the Microsoft SQL Management studio platform. The database queries were basic, copying the entire table or a specific column [6].

In PyCharm professional, where the code for this project has been written, it is possible to connect the database and set a specific dialect. The illustration shows how the connection looks in the program, as well as approximate type of queries that were used to retrieve information from the database.

Final result and company checking the product

During the last project presentation, the employer reviewed the results and project, and the company was satisfied with the fact that shipments created by the program matched the ones provided to designer by them. The illustration demonstrates the tables separated by an indentation and the dates of shipments in each shipment are the same, so the program has calculated everything correctly.

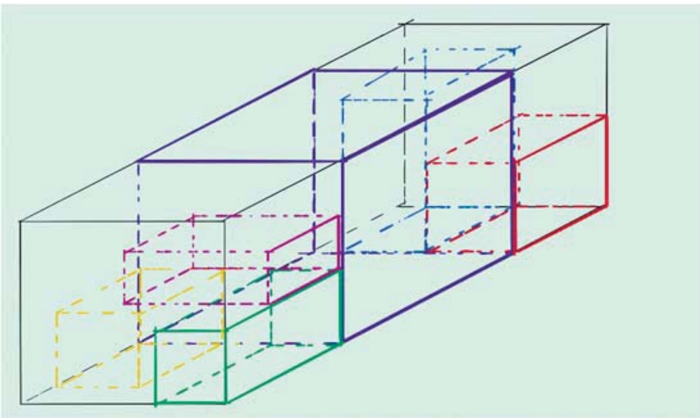


Fig. 8. Visualisation

Future development

Development of the project – is, first of all, improving the logic of the algorithm to achieve 100% correct operation, and also implementing the very function of visualization (Fig. 8).

Conclusion

The project has been approved by the company and can be used if with stability and reliability improvements, so it can operate within company's environment. The general idea was implemented and minimal required results were archived. So what is left – basically to implement this whole application to company based platform.

References

1. E.Yu. Baranova, A.A. Lugovec, A.R. Mel'nikov, M.A. Mel'nikova, V.M. Pokidyshev, K.A. Sylka, A.A. Funtusov. The methodical substantion of optimization of the system of transport and forwarding support for cargo delivery in intermodal. *Marine intellectual technologies*. 2017. No. 3-2 (37), pp. 193-202.
2. A.A. Belyakov, A.I. Shulepov. Cargo arrangement management system. *Amur State University Bulletin. Series: Physical and economical sciences*. 2021. No. 93, pp. 37-41.
3. A.A. Belyakov, A.I. Shulepov. Methodology of differentiated cargo arrangement strategy. *In collection: Mechatronics, automation and transport operation. IV pan-Russian research-to-practice conference*. 2022, pp. 178-182.
4. A.E. Gorev .The choice of optimal technology by delivering cost building materials. *Modern problems of science and eduction*. 2012. No. 4, pp. 106.
5. S.V. Mozgrin. Modular concept of the use of fitting platforms for expanding the range of transported goods. *Railroad research and development centre Bulletin*. 2018. Vol. 77. No. 3, pp. 177-181.
6. S.V. Mozgrin, G.E. Pisarevskij. Effectiveness of arrangement of the long cargo transportation on the fitting platforms with different length of freight storage. *Railroad research and development centre Bulletin*. 2019. Vol. 78. No. 4, pp. 203-209.

7. E.S. Musonov. Design and development of information system of automated formation and arrangement of pallets in cargo semitrailer. *In collection: Logistics – Eurasian bridge. XIV International research-to-practice conference*. Krasnoyarsk State Agrarian University. 2019, pp. 227-229.

8. I.A. Oleynikova, V.N. Sokolov. Complete of combined loads in transport companies. *Technical regulation in transport construction*. 2019. No. 2 (35), pp. 180-184.

9. V.A. Kornienko. Logistic approach in cargo transportation arrangement. In collection: professional English in use. *III pan-Russian research-to-practice conference of young scientists with international participation materials*. 2017, pp. 147-148.

10. Y.V. Nikitin. Static and tip-over stability analysis of two-chain suspension arrangements for large-scale cargo operations. *WMU Journal of Maritime Affairs*. 2014. Vol. 13. No. 1, pp. 101-126.

11. O. Pokusaev, D. Namiot. Anomalies in transport data. *In collection: IOP Conference Series: Earth and Environmental Science. Series "International Science and Technology Conference "Earth Science"*. Chapter 4. 2021, pp. 052015.

12. J. Vrabel, T. Skrucany, J.Koprna, L. Bartuska Movement analysis of the semitrailer with the tank-container at hard braking-the case study. *In collection: IOP Conference Series: Materials Science and Engineering*. 4. Сер. "IV International Conference of Computational Methods in Engineering Science, CMES 2019". 2019, pp. 12-25.

Sources

1. Flask web development, one drop at a time: https://flask.palletsprojects.com/en/2.2.x/

2. Python 3.10.8 documentation: https://docs.python.org/3/

3. HTML: HyperText Markup Language: https://developer.mozilla.org/en-US/docs/Web/HTML

4. CSS: Cascading Style Sheets: https://developer.mozilla.org/en-US/docs/Web/CSS

5. Openrouteservice: https://openrouteservice.org/

6. Welcome to GeoPy's documentation: https://geopy.readthedocs.io/en/stable/

ОСОБЕННОСТИ РАЗРАБОТКИ АЛГОРИТМА ВИЗУАЛИЗАЦИИ ОПТИМАЛЬНОЙ РАССТАНОВКИ ГРУЗА В ГРУЗОВОМ ПОЛУПРИЦЕПЕ

Исаева Сабина Маджидовна, *ФГАОУ ВО "Российский университет транспорта", Москва, Россия,* ***isaeva-sabina@yandex.ru***

Шобанов Егор Андреевич, *ФГАОУ ВО "Российский университет транспорта", Москва, Россия,* ***egorco.sa@gmail.com***

Аннотация

В статье рассматриваются особенности оптимального расположения сборных грузов в алгоритме полуприцепа. В настоящее время достаточно важно оптимизировать логистические процессы в связи с высоким спросом на этот вид услуг. Автор описывает процесс разработки и проектирования специального приложения, которое автоматически упорядочивает грузы в полуприцепе. В тексте приводится ценная информация, относящаяся к структуре полуприцепа и логистике. Подробно говорится о восьми шагах, объясняющих весь процесс разработки приложения. Большое внимание уделяется одному конкретному шагу, объясняющему работу приложения поэтапно. Такое приложение довольно удобно для пользователя, поэтому любой логист может использовать его без инструкций. Решение применимо в реальном рабочем процессе с некоторыми окончательными улучшениями в отношении совместимости с другими системами, а также это решение помогает ускорить рабочие процессы в логистических услугах, поэтому некоторые средства могут быть сэкономлены. Точная сэкономленная сумма может отличаться от компании к компании из-за индивидуальной внутренней структуры. Подводя итоги, можно сделать вывод, что решение достаточно многопрофильное, то есть оно может быть применимо к нескольким компаниям с некоторыми изменениями из-за конкретных запросов компании.

Ключевые слова: *питон, перевозка грузов, система декартовых координат, проектирование, груз, полуприцеп*

Литература

1. *Баранова Е.Ю., Луговец А.А., Мельников А.Р., Мельникова М.А., Покидышев В.М., Сылка К.А., Фунтусов А.А.* Методическое обоснование оптимизации системы транспортно-экспедиторского обеспечения доставки грузов в смешанном сообщении // Морские интеллектуальные технологии. 2017. № 3-2 (37). С. 193-202.

2. *Беляков А.А., Шулепов А.И.* Система управления размещением грузов // Вестник Амурского государственного университета. Серия: Естественные и экономические науки. 2021. № 93. С. 37-41.

3. *Беляков А.А., Шулепов А.И.* Методика дифференцированной стратегии размещения грузов // В сборнике: Мехатроника, автоматизация и управление на транспорте. Материалы IV Всероссийской научно-практической конференции. 2022. С. 178-182.

4. *Горев А.Э.* Выбор оптимальной технологии по критерию затрат доставки строительных грузов // Современные проблемы науки и образования. 2012. № 4. С. 106.
5. *Мозгрин С.В.* Модульная концепция использования фитинговых платформ для расширения номенклатуры перевозимых грузов // Вестник Научно-исследовательского института железнодорожного транспорта. 2018. Т. 77. № 3. С. 177-181.
6. *Мозгрин С.В., Писаревский Г.Е.* Эффективность организации перевозок длинномерных грузов на фитинговых платформах с различной длиной погрузочных площадок // Вестник Научно-исследовательского института железнодорожного транспорта. – 2019. Т. 78. № 4. С. 203-209.
7. *Мусонов Е.С.* Проектирование и разработка информационной системы автоматизированного формирования и расстановки паллет в грузовом полуприцепе // В сборнике: Логистика – евразийский мост. Материалы XIV Международной научно-практической конференции. Красноярский государственный аграрный университет. 2019. С. 227-229.
8. *Олейникова И.А., Соколов В.Н.* Комплектование сборных грузов в транспортных компаниях // Техническое регулирование в транспортном строительстве. 2019. № 2 (35). С. 180-184.
9. *Kornienko V.A.* Logistic approach in cargo transportation arrangement // В сборнике: professional English in use. материалы III Всероссийской научно-практической конференции молодых ученых на английском языке с международным участием. 2017. С. 147-148.
10. *Nikitin Y.V.* Static and tip-over stability analysis of two-chain suspension arrangements for large-scale cargo operations // WMU Journal of Maritime Affairs. 2014. Т. 13. № 1. С. 101-126.
11. *Pokusaev O., Namiot D.* Anomalies in transport data // В сборнике: IOP Conference Series: Earth and Environmental Science. Сер. "International Science and Technology Conference "Earth Science" - Chapter 4" 2021. С. 052015.
12. *Vrabel J., Skrucany T., Koprna J., Bartuska L.* Movement analysis of the semitrailer with the tank-container at hard braking-the case study // В сборнике: IOP Conference Series: Materials Science and Engineering. 4. Сер. "IV International Conference of Computational Methods in Engineering Science, CMES 2019". 2019. С. 12-25.

Источники

1. Flask web development, one drop at a time: https://flask.palletsprojects.com/en/2.2.x/
2. Python 3.10.8 documentation: https://docs.python.org/3/
3. HTML: HyperText Markup Language: https://developer.mozilla.org/en-US/docs/Web/HTML
4. CSS: Cascading Style Sheets: https://developer.mozilla.org/en-US/docs/Web/CSS
5. Openrouteservice: https://openrouteservice.org/
6. Welcome to GeoPy's documentation: https://geopy.readthedocs.io/en/stable/

Информация об авторах:

Исаева Сабина Маджидовна, *ФГАОУ ВО "Российский университет транспорта", Академия базовой подготовки, кафедра "Русский и иностранные языки", старший преподаватель, Москва, Россия*

Шобанов Егор Андреевич, *ФГАОУ ВО "Российский университет транспорта", Академия "Высшая инженерная школа", учебная группа ШАД-211, Москва, Россия*