

# DEVELOPMENT OF A TOOL FOR OPTIMIZED OPERATIONS OF A DANGEROUS GOODS AREA IN THE HARBOR OF HAMBURG



**Gabriele Lacchin**  
Software Developer



**Miguel Peñate Arrieta**  
Logistics & Design Thinking



**Harsh A Lokhande**  
Product Development



**Yehia Assi**  
Product owner



**Arpita Anand Prayag**  
Scrum Master

## PROBLEM STATEMENT

- Storage area relocation due to city development plans.
- Complex regulations.
- No automated compliance checks for dangerous goods.
- Inefficient space usage.

## OBJECTIVES

- **Automate Storage Management:** Develop an algorithm to manage the storage of dangerous goods in a warehouse.
- **Efficient Space Utilization:** Optimize the use of available storage space while adhering to safety regulations.

## WORK DONE

- **Successful Data Processing:** Functions developed for effective data reading and processing.
- **Accurate Matching:** Regex-based matching for substances with priority levels implemented.
- **GUI Functionality:** Interactive, functional GUI created for managing storage.
- **Safe Storage:** Safety constraints adhered to with action logs for container storage.

## FUTURE SCOPE

- **Advanced algorithms** - better space optimization.
- **Real-Time Monitoring** - Creating a database for keeping track of the stored containers.
- **Integration** - with warehouse management systems for seamless operation.

## CONCLUSION

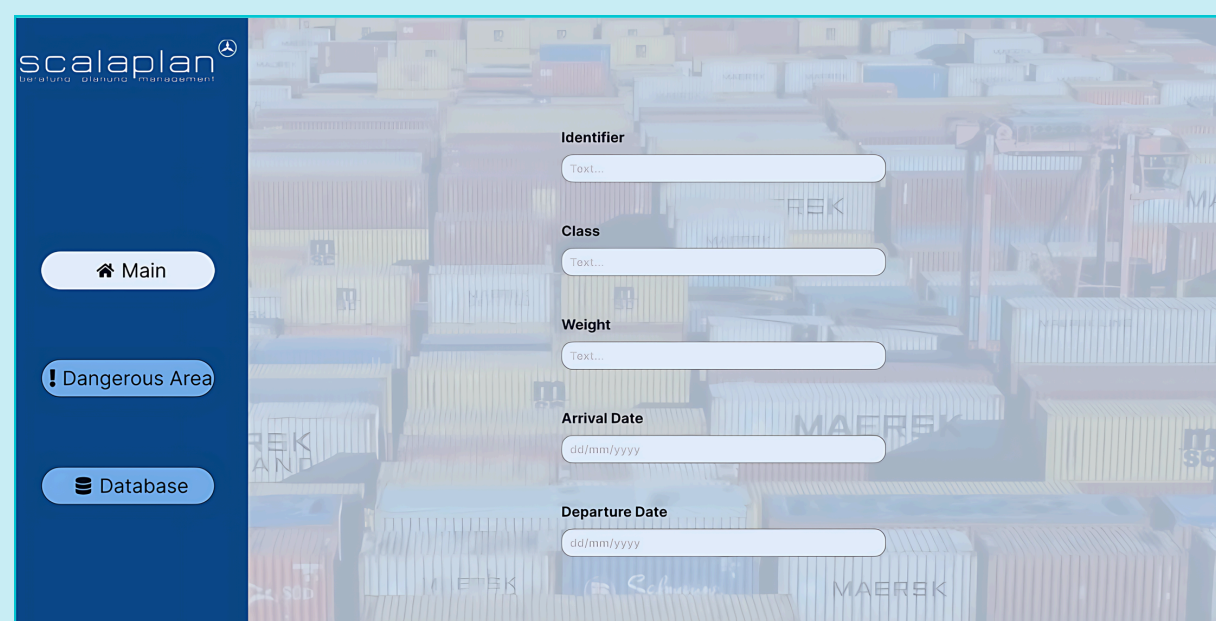
- Developed a tool for the placement of dangerous goods containers instantly.
- Automated regulatory compliance checks, reducing risks.

Load Container Data		Store Containers		Remove Container	
Num(umschlag_id)	Weight	Class	WGK Level		
1.0	282.0	2.1	2.0		
2.0	249.0	5.1	3.0		
3.0	71.0	2.2	3.0		
4.0	124.0	2.1	1.0		
5.0	101.0	2.1	1.0		
6.0	116.0	4.3	3.0		

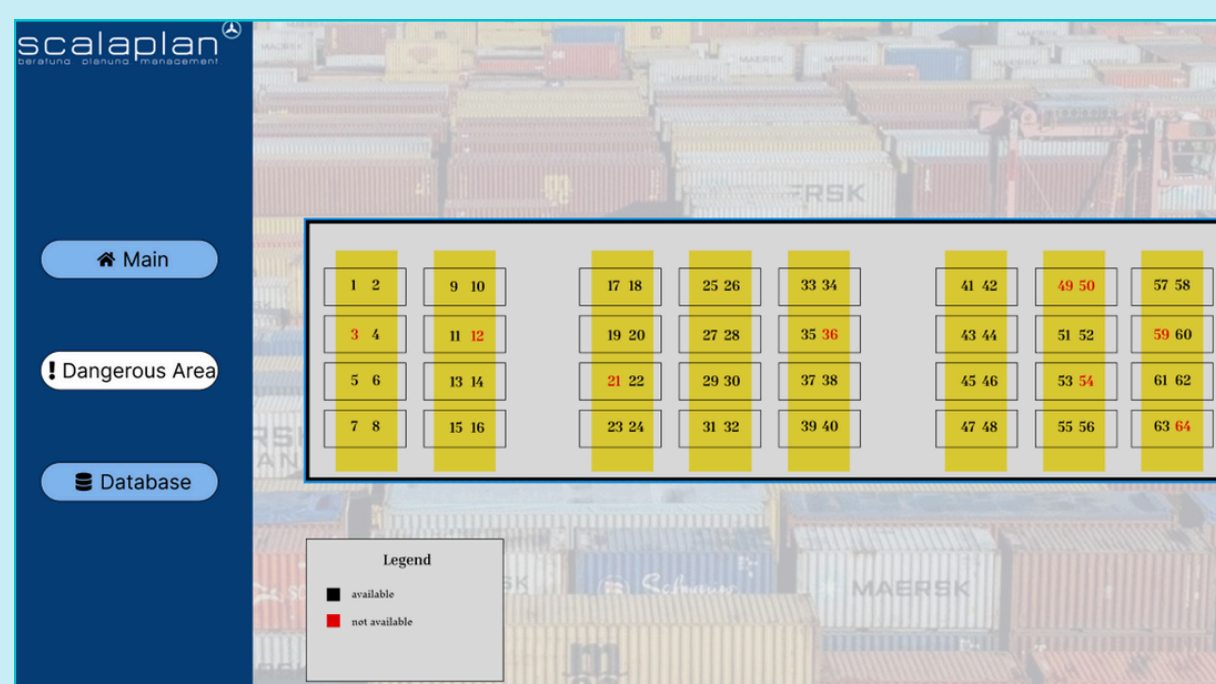
  

Log	Subarea States
Container 1.0 stored in A(1,1)	Subarea A:
Container 2.0 stored in A(2,1)	(1,1): 1.0, 4.0 - 2.1 - 406.0kg
Container 3.0 stored in B(1,1)	(1,2): Empty
Container 4.0 stored in A(1,1)	(1,3): Empty
Container 5.0 stored in B(2,1)	(1,4): Empty
Container 6.0 stored in B(3,1)	(2,1): 2.0 - 5.1 - 249.0kg
	(2,2): Empty
	(2,3): Empty
	(2,4): Empty

Dangerous Goods Storage Management



Planned Interface



Identifier	Class	Weight [to.]	Arrival Date	Departure Date
1116255	9	2,230	02/01/2020	14/01/2020
1117462	9	2,840	06/01/2020	17/01/2020
1117422	9	12,968	06/01/2020	17/01/2020
1117461	4.1	1,573	08/01/2020	13/01/2020
1117423	6.1	2,645	10/01/2020	15/01/2020
1117457	4.1	3,240	13/01/2020	16/01/2020
1118141	2.2	8,800	13/01/2020	05/02/2020
1120274	3	0,864	13/01/2020	14/01/2020
1119014	9	1,710	13/01/2020	14/01/2020
1119522	4.1	4,095	14/01/2020	17/01/2020
1117985	6.1	0,004	14/01/2020	18/01/2020
1117985	8	3,619	14/01/2020	15/01/2020
1119688	4.1	2,085	20/01/2020	22/01/2020
1119667	3	0,009	20/01/2020	27/01/2020
1119662	2	3,212	21/01/2020	27/01/2020
1119627	4.1	3,234	21/01/2020	25/01/2020
1110667	6.1	0,786	21/01/2020	22/01/2020
1118667	2	1,123	22/01/2020	23/01/2020
1118667	9	2,232	23/01/2020	26/01/2020
1119645	8	0,132	23/01/2020	26/01/2020
1119633	2	0,111	23/01/2020	02/02/2020
1119644	2	1,121	23/01/2020	01/02/2020

## METHODOLOGY

- **Data Input** → Read and process input from CSV and Excel files.
- **Matching Substances** → Use regex to match substances with WGK levels.
- **Storage Rules** → Set class distances and weight limits for storage.
- **GUI Development** → Build a user-friendly GUI with Tkinter.
- **Container Placement** → Place containers based on constraints and space.

