Integrative Project – Project Assignment
LEI - 2021/2022 – 1st Semester, 2nd Year
v1.1, October 25th

Abstract

A cargo shipping company requires a software system to handle their logistics. This company operates through land and sea, across different continents and has several warehouses spread along the world.

Table of Contents

1 Integrative Project's Goals and Context

In this project, students should be able to apply concepts of analysis, modelling and object-oriented programming to develop a system that supports the management of a logistics company.

In compliance with good practices learned and applied during the first part of the semester, namely on the course units of Applied Physics (FSIAP), Computer Architecture (ARQCP), Data Structures (ESINF) and Databases (BDDAD), and Laboratory/Project III (LAPR3) an iterative and incremental development process is used. An agile methodology based on Scrum must be applied to manage each team's work during each three-week sprint.

For teams where students are enrolled in all subjects, the system will be composed by two applications, one developed in Java, and another developed in C/Assembly.

To increase software's maintainability and best practices, Object-Oriented (OO) software design must be adopted, and implementation should follow a Test-Driven Development (TDD) approach.

Technical documentation must be produced during the project by using Javadoc documentation and the Readme.md file in your repository.

2 Problem Description

A Cargo shipping company requires a software system to handle their logistics. This company operates through land and sea, across different continents and has several warehouses spread along the world.

2.1 Containers

Due to the number of containers being transported, the company now requires some software that allows them to track the fastest routes from one point in the world to another.

Containers can be transported by land or sea. Every ship has the containers according to a three-axis displacement (as shown in Figure 1), always starting on the same corner of the ship, the corner between Port and Stern. This is point (0,0,0) for coordinates (x,y,z). For example, on Figure 1 container 1) is positioned at (0, 2, 2); container 2) is positioned at (1,0,6) and container 3) is positioned at (1,1,7). Ships may have containers below that axis using a negative number for the "z" axis.

Figure 1 – Three-axis displacement on a ship[1]

As shown in Figure 2 containers have certain characteristics[2] such as:

- Container Identification.
- Container Payload, Tare and Gross.
- Container dimensions (width, height, length) by using the ISO Code[3].

[1] https://lineragency.greencarrier.com/wp-content/uploads/2016/06/strictcontrols.jpg
[2] https://container-xchange.com/blog/container-markings-what-do-they-mean/
[3] https://www.containercontainer.com/iso6346/

- For the time being let us assume that all containers share the same length.

- Can or not be refrigerated (not shown on the Figure).
  - If it is refrigerated, at what temperature it should be kept.

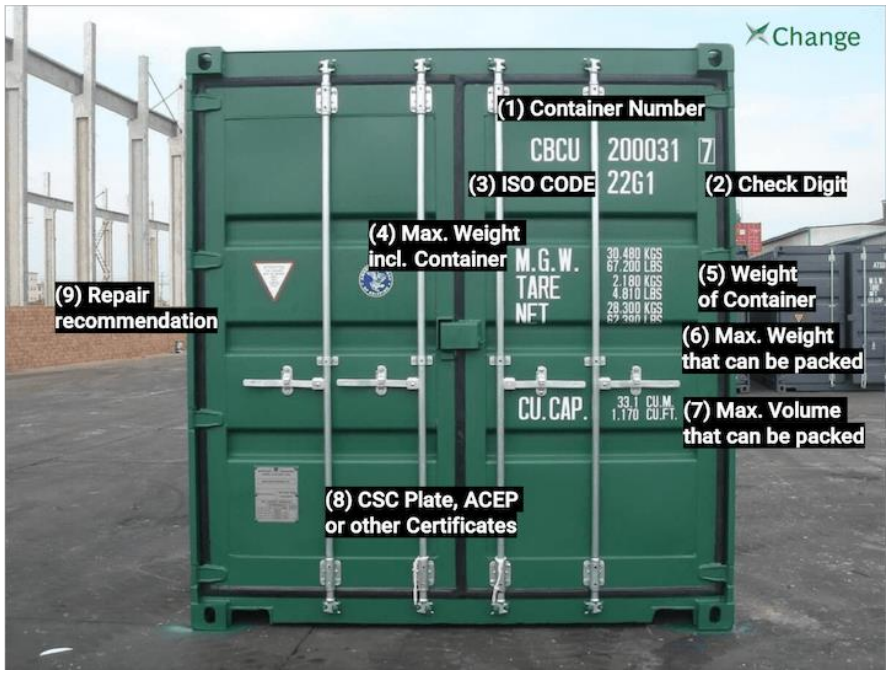

Figure 2 - Container characteristics[4]

2.2 Cargo Manifest

Each time a ship docks at a port or a truck arrives at a location, an Unloading Cargo Manifest is delivered to the operations team at the port or warehouse and at the end of operations a Loading Cargo Manifest is delivered to the ship crew. The same is true for trucks. The Cargo Manifest only has the necessary information about the containers that are being loaded and unloaded.

Each Cargo Manifest has at least the following information:

- Container identification.
- Container position in the transport vehicle.
  - We will be assuming that truck transport can only carry one container at a time.
- Container gross weight.

Furthermore, some containers can be refrigerated and rely on an outside energy source to work. Ships can fire up energy producing units to supply the energy for those containers. Yet, to preserve fuel those are set to reduce the energy producing units to a bare minimal by default.

[4] https://container-xchange.com/wp-content/uploads/2019/01/container-markings-xchange-2.png

2.3 Ships

Ships are characterized by:

- MMSI: unique 9-digit ship identification code.
- Ship name.
- Ship identification according to the IMO identification number[5].
    - IMO: unique 7-digit international identification number, which remains unchanged after transferring the ship's registration to another country.
- Numbers of energy generators.
- Generator's power output.
    - Let us assume that all generators provide the same power output.
- Call sign: ship's unique callsign.
- Vessel type: ship type, numerically coded.
- Length: ship length, in meters.
- Width: ship width, in meters.
- Capacity: ship load capacity in m3.
    - Ship maximum capacity according to a container mapping (using a three-axis displacement).
- Draft: Vertical distance between the waterline and the bottom of the ship's hull, in meters. Varies with ship load and water density

Dynamic data fields relating to a ship's positioning data:

- Base Date Time: date/time of AIS message.
- Latitude: ship latitude (in degrees: [-90; 90], negative value represents South, 91 indicates 'not available').
- Longitude: ship longitude (in degrees: [-180; 180], negative value represents West, 181 indicates 'not available').
- Speed over ground (SOG).
- Course over ground (COG): direction relative to absolute North (in degrees: [0; 359]).
- Heading: ship heading (in degrees: [0; 359], 511 indicates 'not available').
- Position: ship code in tow.
- Transceiver Class: class to transceiver used when sending data.

[5] https://www.imo.org/en/OurWork/MSAS/Pages/IMO-identification-number-scheme.aspx

2.4 Ports and Warehouses

Every Port and Warehouse should have at least the following information:

- Identification.
- Name.
- Continent.
- Country.
- Location: latitude and longitude.

2.5 Roles

The system is to be used by the following people:

- Client – someone who wants to get a quote for a shipment from one place to another.
- Fleet manager – someone who controls the fleet.
- Traffic manager – someone who controls the shipping/truck position.
- Warehouse staff – someone who loads and unloads cargo from warehouses.
- Warehouse manager – someone who manages the existing warehouses.
- Port staff – someone who loads and unloads cargo from ships.
- Port manager – someone who manages the list of ports.
- Ship captain – someone who oversees ship operations.
- Ship chief electrical engineer – someone who controls the electrical part of a ship.
- Truck driver – someone who oversees truck operations.

2.6 Non-Functional Requirements

This section describes some of the non-functional requirements that must be taken into consideration when implementing your project.

All users must be registered in the system.

Business logic validation should take place when registering or updating data. The database is the main repository for the application and should always reflect an integrity state. All data should be persisted on a remote SGBD. Failure to comply with this will penalize the project grading.

To maximize interoperability with other existing and/or developing systems, the software main core should be written in Java. Some other software used for quick notifications should be developed in C/Assembly.

The class structure must be designed to allow easy application maintenance and addition of new features and following good OO design practices.

3 Minimum Viable Product

The goal of this assignment is to achieve a Minimum Viable Product in incremental steps. For this purpose, the work will be divided in three Sprints:

- Sprint 1 – October 25th to November 12th.
- Sprint 2 – November 15th to December 3rd.
- Sprint 3 – December 6th to December 22nd.
- Sprint 4 – January 3rd to January 21st.

For each Sprint, a description for a minimum viable product is provided to students. Students should follow the user stories and perform story mapping. At the end of each sprint, each team should be able to have all the requirements fulfilled.

Teams should be able to add the User Stories to their backlog, size them and assign them over to each team member.

3.1 Sprint 1

The maritime transport is provided by different ships. Ships send messages during voyages to the US Coast Guard. These messages contain the positional and displacement information of each ship at a given time. The file (ships.csv) contains static information about the ship details and dynamic data regarding the voyage [6].

- [US101] As a traffic manager, I which to import ships from a text file into a BST.
    - Acceptance criteria [ESINF]:
        - no data lost.
- [US102] As a traffic manager I which to search the details of a ship using any of its codes: MMSI, IMO or Call Sign.
    - Acceptance criteria [ESINF]:
        - correct use of OOP concepts.
- [US103] As a traffic manager I which to have the positional messages temporally organized and associated with each of the ships
    - Acceptance criteria [ESINF]:
        - efficient access of any position value(s) of a ship on a period or date.
- [US104] As a traffic manager I which to make a Summary of a ship's movements.
    - Acceptance criteria [ESINF]:

[6] https://www.youtube.com/watch?v=3M7mxBimT70

        - For a given ship return in an appropriate structure one of its codes (MMSI, IMO or Call Sign), Vessel Name, Start Base Date Time, End Base Date Time, Total Movement Time, Total Number of Movements, Max SOG, Mean SOG, Max COG, Mean COG, Departure Latitude, Departure Longitude, Arrival Latitude, Arrival Longitude, Travelled Distance (incremental sum of the distance between each positioning message) and Delta Distance (linear distance between the coordinates of the first and last move)[7].
- [US105] As a traffic manager I which to list for all ships the MMSI, the total number of movements, Travelled Distance and Delta Distance.
    - Acceptance criteria [ESINF]:
        - ordered by Travelled Distance and total number of movements (descending/ascending).
- [US106] Get the top-N ships with the most kilometres travelled and their average speed (MeanSOG).
    - Acceptance criteria [ESINF]:
        - in a period (initial/final Base Date Time) grouped by Vessel Type.
- [US107] Return pairs of ships with routes with close departure/arrival coordinates (no more than 5 Kms away) and with different Travelled Distance.
    - Acceptance criteria [ESINF]:
        - Sorted by the MMSI code of the 1st ship and in descending order of the Travelled Distance difference.
        - Do not consider ships with ~~the number of movements greater than 1000 and~~ Travelled Distance less than 10 kms.
- [US108] As Project Manager, I want the team to develop the data model required to support all the functionality and to fulfill the purpose of the system to develop. This data model is to be designed following a systematic data modeling methodology.
    - Acceptance Criteria [BDDAD]:
        - The result should include (1) the conceptual data model, (2) the logical data model according to the database technology to use, (3) the physical data model to be implemented at the selected DBMS, (4) a data dictionary describing the relevant details of the database elements and (5) a clear and concise justification supporting the selected database technology.

[7] http://www.movable-type.co.uk/scripts/latlong.html

        - It is possible to run a SQL script to create the database in a complete and consistent way without errors.
        - The data models generated at each one of the three abstraction levels (conceptual, logical, and physical) map only the meaningful concepts and characteristics in compliance with the corresponding level.
        - Each one of the data models respects the former one, i.e., the logical data model respects the conceptual data model, and the physical data model respects the logical data model. The conceptual data model is a valid view/representation of the UoD.
        - The notation used for each one of the data models is adequate, consistent, and following the specifications.
- [US109] As Project Manager, I want the team to draft an SQL script to test whether the database verifies all the data integrity restrictions that are required to fulfil the purpose of the system and the business constraints of the UoD.
    - Acceptance Criteria [BDDAD]:
        - There is a catalogue of data integrity restrictions grouped by type (Domain, Identity, Referential, Application) clearly stated.
        - For each data integrity restriction in the catalogue there is a set of SQL instructions that verify the restriction.
        - All SQL instruction in the data integrity verification script are accompanied by a comment that describes the expected result (Pass or Fail; in the latter a justification is given).
- [US110] As Project Manager, I want the team to define the naming conventions to apply when defining identifiers or writing SQL or PL/SQL code. The naming conventions may evolve as new database and programming objects are known. The naming conventions guide should be organized in a way to facilitate its maintenance.
    - Acceptance Criteria [BDDAD]:
        - There are naming conventions clearly stated to create databases and database objects. The minimum set includes tables, attributes, constraints, primary and foreign keys.
        - The naming conventions are available in a way that makes them easy to understand and complete in a continuous way.
- [US111] As Project Manager, I want the team to create a SQL script to load the database with a minimum set of data sufficient to carry out data integrity verification and functional testing. This script shall produce a bootstrap report providing the number of tuples/rows in each relation/table.

- Acceptance Criteria [BDDAD]:
  - The bootstrap SQL script runs and loads the database as expected with no errors
  - The bootstrap report is generated and correct, i.e., all tables are mentioned, and their cardinality is correct.

3.1.1 Non-Functional Requirements

To assure data integrity in the database there are restrictions that apply generally, regardless of any other considerations. Many of these arise naturally, such as, length, width, weight, and volume cannot be negative. Others, such as latitude and longitude, although being also natural, might not be so well known to non-experts in the field.

Besides these general restrictions, the UoD imposes a few others, such as, "IMO: unique 7-digit international identification number" or "ship latitude …91 indicates 'not available'". All the data integrity restrictions from the UoD should be considered and described when designing the database; some of these may be imposed at the data model, others require coding.

4 Technical Details

4.1 ARQCP

In Sprint 3, all needed data structures should be statically reserved in memory, considering the worst-case scenario. Functions developed in Assembly have no parameters. As such, all needed input should be given as global variables, either defined in C or Assembly.

In Sprint 4, all needed data structures should be dynamically reserved in memory, adapting the amount of reserved memory to the amount of data being processed. Functions developed in Assembly should have parameters. As such, global variables are no longer needed.

4.2 BDDAD

Students should create one database according to the tutorial that will be provided on Moodle. These databases have the following name "LAPR3-GXXX" where XXX should be replaced by your team number.

All the scripts for creating/updating your database schema should be exported to your project's git repository.

4.3 ESINF

n/a.

4.4 FSIAP

n/a.

4.5 LAPR3

More information is available on LAPR3 Moodle's web page[8].

[8] https://moodle.isep.ipp.pt/course/view.php?id=1681

5 Revision History

V0.5	Work in progress.
V1.0	Initial Release.
V1.1	Correction to Acceptance Criteria on US107.

Notes: new changes are underlined while removed content are strikethrough.