ASSIGNMENT COVERSHEET

Student Name: Daniil Khmelnytskyi, Alexander Markov, Kirill Shulyak, Teo Bocev, Ahmed Khaled Ibrahem Ali Allam	
Class: Symbolic Computation	
Assignment: Clojure for Logistics	
Lecturer: Leonov Evgeny, Ahmet Vargelen	Semester: 2404
Due Date: 29.11.2024	Actual Submission Date: 29.11.2024

Evidence Produced (List separate items)	**Location (Choose one)**	
	X	Uploaded to the Learning Center (Moodle)
		Submitted to reception
Note: Email submissions to the lecturer are not valid.		

Student Declaration:	
I declare that the work contained in this assignment was researched and prepared by me, except where acknowledgement of sources is made. I understand that the college can and will test any work submitted by me for plagiarism. **Note:** The attachment of this statement on any electronically submitted assignments will be deemed to have the same authority as a signed statement	
Date: November 29, 2024	Student Signature: Daniil Khmelnytskyi, Alexander Markov, Kirill Shulyak, Teo Bocev, Ahmed Khaled Ibrahem Ali Allam

A separate feedback sheet will be returned to you after your work has been graded.
Refer to your Student Manual for the Appeals Procedure if you have concerns about the grading decision.

Student Comment (Optional)
Was the task clear? If not, how could it be improved?
Was there sufficient time to complete the task? If not, how much time should be allowed?
Did you need additional assistance with the assignment?
Was the lecturer able to help you?
Were there sufficient resources available?
How could the assignment be improved?

Happy Fruit's trucks optimal routes solution

Daniil Khmelnytskyi, Alexander Markov, Kirill Shulyak, Teo Bocev,
Ahmed Khaled Ibrahem Ali Allam

November 29, 2024

Contents

Abstract

This report will document the development process of Happy Fruit's new delivery system and logistics software. It provides an in-depth analysis into the team's decision making process from the initial requirements analysis, to the delivery process to the customer.

1 Introduction

In order to remain competitive in today's business environment, an efficient logistics plan is crucial. For a company like Happy Fruit, which distributes canned pineapples across major European cities, logistical efficiency is a key component in ensuring operations are conducted smoothly. Happy Fruit stores their product in warehouses of a limited capacity while maintaining a minimum stock level at each warehouse. The major challenge the business faces is being able to supply their stores where the product is sold, while maintaining minimum stock levels and keeping transportation costs as low as possible. This required a software solution providing routes that meet the business' previously stated constraints. The application was developed in Clojure by a team of developers, and it provides logistical efficiency while complying with warehouse stock policies. The report dives further into the application's features, technical details as well as the journey of it's development and all details relevant to the user.

2 Scoping

This section is dedicated for the list of the features and systems that will be implemented during the design of the optimal routes solution for the Happy Fruit's trucks, which of them are in scope of the current project and will be done, which are out of scope and will not.

2.1 Project Overview

This project implements a delivery route optimization system for Happy Fruit, a company specializing in canned pineapples distribution across European cities. The system calculates optimal delivery routes for trucks while maintaining required inventory levels at each warehouse location.

2.2 Features

The core functionality of this system revolves around efficient delivery route planning. Key features include:

- Route optimization for multiple trucks
- Warehouse inventory management
- Capacity constraint handling
- Real-time distance calculations

2.3 In scope list

The following list contains the items, which will be implemented during the process of the current project:

- **Database Development**: The database will store all the essential data, which can be accessed by different users like drivers, warehouse workers, and managers.
- **Program Code**: It will be responsible for processing user input, updating stock levels, assigning trucks for deliveries, and checking routes.
- **Technical Design and User Interface**: The design will cover how the database, program logic, and user interface work together. This includes a basic, easy-to-use interface where users can view stock levels, truck availability, and delivery routes.

- **Monitoring Stock Levels**: The system will check stock levels regularly. If a city's stock drops below a certain threshold, the system will trigger a resupply. This will help prevent stockouts in the different locations.
- **Improving Delivery Operations**: The goal is to make the delivery process more efficient. The system will help managers decide which locations need restocking first, so all shops have enough stock.

2.4 Out of Scope

The following list contains the items, which will NOT be implemented during the process of the current project:

- **Support and Maintenance After Delivery**: Once the system is built and handed over, we won't be providing ongoing support or troubleshooting. Any issues after deployment would need to be handled separately, as maintenance isn't part of this project.
- **Feedback Collection and Updates**: There will not be a formal way to collect user feedback or update the system based on feedback. This project is a prototype, so any future improvements would have to be done as separate updates.
- **Real-Time GPS Tracking**: There won't be live GPS tracking for trucks or drivers. The system will show stock and delivery data, but it won't include any live location tracking features.
- **Mobile App Development**: The system is meant to be used on a desktop within the company's network. There won't be a mobile app version or any customizations for mobile devices.
- **Multi-Language Support**: Everything in the UI will be in a single language (like English). We won't be adding support for multiple languages.

3 Analysis gathering

In this section, the detailed information about the approaches and technologies which will be used during the current project.

3.1 Design choices analysis

- **DataBase**
    - Implementation of the technical infrastructure for a DataBase functioning. The server will be rented from a third contractor company. Domestic server implementation requires a specialized personnel for a maintenance purposes.
    - For the database engine the PostgreSQL was chosen due to it's scalability, performance and support of document data.
    - Integration with the BackEnd of the application will be implemented via set of SQL functions and procedures.
- **Design and UI**
    - The application user interface will be implemented as a website with
    - Authentication for truck drivers, warehouses keepers and administrators will be implemented via accounting system with different access levels.
- **Program Code**
    - The main logic of the program will be written in Clojure, and this code will handle how the database and user interface talk to each other.
    - Language choice was based on two main Clojure benefits for this project. First one is that this language is data-oriented, which makes it an excellent tool for complex data workflows . Second one is that this language is based on immutable data structures, making consistency and state management easier.

3.2 User journeys

User journeys cover two main cases of system usage. When the system's work is being initiated by truck driver, if there is a need of manual restock session creation. And when it is being initiated by itself, the program will compare the initial value and minimum capacity of each warehouse every morning.

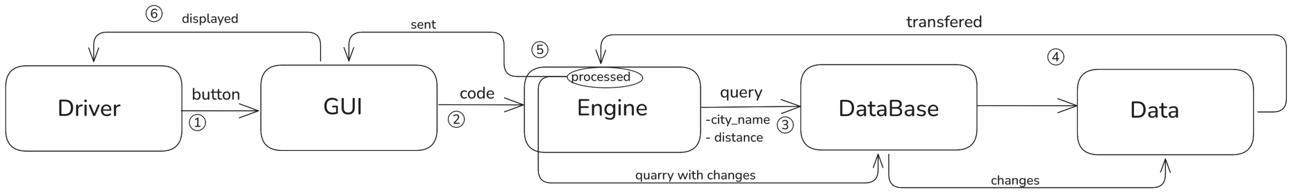

Figure 1: Graphical representation of the first user journey

1. **Driver** initiate the script by iterating with **GUI** via pushing "Create route" button on it

2. **GUI** interact with the **program engine** by code (HTTP requests)

3. **Engine** sends an SQL query to a **database** to take the information about distances between needed cities

4. Based on SQL query, **database** returns needed **data** and transfers it to the **program's engine**

5. After the **data** reached **Back-end**, the main script is being initiated. The result of the calculations is being sent to the **interface**

6. **Interface** displays results and become accessible to a **driver**

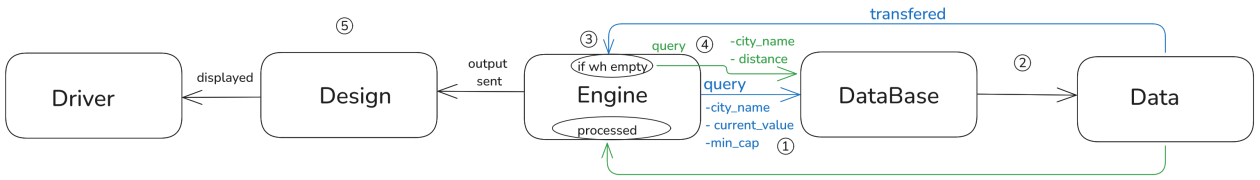

Figure 2: Graphical representation of the second user journey

1. (Blue path) **Program's engine** starts a daily supply check and sends an SQL query to an **database server**. It requests information about current status of the storage and the minimum capacity of each warehouse

2. (Blue path) Based on request, **database** returns a set of **data**. This data is being transferred to an **engine**

3. **Script** checks does the warehouse meet the conditions by comparing current and minimal capacity of it. If it does, script drops. If some warehouse does not fit the conditions, the green path starts

4. (Green path) **Program's engine** sends a new SQL query to a **database server**. The data is needed to initiate the main script for a warehouse(s), which did not fit the conditions
5. After the **data** reached **Back-end**, the main script is being initiated. The result of the calculations is being sent to the **interface**
6. **Interface** displays results and become accessible to a **driver**

4 Risks and disclaimers

4.1 Risks

- Due to the implementation of the database server deployment, company will be dependent by the server provider company
  - In case of crashes, data loses or other problems with database, additional costs or time is needed for malfunction
  - Lack of the control over the rented database, may lead to problems with securing server the way, is intended by the company.
- Users (truck drivers, warehouse keepers) may create hazardous cases for the system due to accidental mistakes or on-purpose
  - In case of the false input data, the amount of product which was delivered may be incorrect
- System does not contain info about traffic, route blockages etc.
  - In case of multiple traffic incidents on the route of the closest path, it can become not the optimal one, leading to delivery delays
  - In case of lack of the weather information, the dates and time of delivery and quality of delivery may change.
- The minor possibility of the database infiltration for the purpose of ransomware attack competitive advantage

4.2 Disclaimers

- In case of database vulnerability exploitation - delivery problems may occur.

- Without connection to GPS and internet services, system will not be able to work.
- System require infrastructure to work on per driver and warehouse keepers.

5 Methodology selection

5.1 The main choice

An Agile approach, especially Scrum, is most likely to work effectively for a project like this, which involves creating logistics software to improve delivery routes, resource allocation, and warehouse management. There are following reasons why Agile (Scrum) is efficient for this project:

- **Development by iteration:** The team is able to develop each component (such as the routing algorithm, inventory monitoring, and truck capacity management) before moving on thanks to Scrum's iterative structure, which makes it perfect for constructing complicated systems progressively.
- **Adaptability:** Scrum provides regular modifications based on comments since the project may change as Happy Fruit's unique requirements and difficulties are identified.
- **Flexibility and Cooperation:** The team may keep stakeholders and Happy Fruit's leadership updated and get suggestions to improve the tool by communicating with them frequently.
- **Risk prevention:** The team can immediately detect and resolve possible software problems, including optimizing for new routes or taking unexpected shifts in warehouse capacity, by working in short cycles, or sprints.

5.2 Possible problems

Scrum has numerous advantages for a project, but it can also have significant disadvantages, especially for a logistics-focused project with detailed needs. Here are several issues:

- **Team dedication hight level of self-management and team-dedication required.** Scrum mostly depends on the dedication and dependency of each team member. Productivity and progress may suffer if any team members lack the self-management abilities required by Scrum or have fewer experiences with Agile procedures.
- **Challenges in final outcome prediction.** Early in the process, it might be challenging to predict exactly how the finished result will be looking because Scrum focuses on gradual delivery and makes modifications depending on feedback. If Happy Fruit's management wants a clear picture of the finished product right away, this could be problematic.

- **Difficulty with technical dependence and sophisticated algorithms.** A short sprint cycle may not be suitable for implementing efficient routing algorithms and managing data relationships between trucks and warehouses, which may call for significant technical investigation and testing. Complex projects like these might additionally ask for constant modifications, which could disrupt the sprint timetable and compromise predictability.
- **Time and resources estimation is difficulty.** It can be difficult to estimate the time and resources needed for each feature or user story in Scrum, particularly if the team is utilizing complicated logistical methods or new technologies. This may result in features that aren't finished on time or sprints that go off schedule.
- **Team burnout risk.** Scrum's iterative process and regular deadlines can occasionally cause team burnout, especially if the group is under pressure to produce complicated features quickly.

5.3 Overall

By applying Scrum in this way, the team will be able to provide Happy Fruit with a scalable, adaptable solution that also permits continuous improvement in response to feedback and changing demands.

6 Plan

6.1 Phase 1 - Pre Development

Activity	**Date of Activity**	**Person/Team Responsible**	**Description**
Scheduling meetings	27.9.2024	Management team	Scheduling the meetings with our and Happy Fruit's management team.
Intro to our plan	30.9.2024	Full team	First meeting, explaining our vision for the project, requirement gathering from the clients.

Activity	**Date of Activity**	**Person/Team Responsible**	**Description**
Defining methodology	1.10.2024	Project manager	Providing Happy Fruit with info for the methodology being used for the project.
Presenting the scope	1.10.2024	Project manager	Going through a summary of the in and out-of-scope requirements.
Database requirement gathering	2.10.2024	Data engineer and analysts	Receiving more details on the data that needs to be stored, tracked, and updated in the system.
Company collaboration meeting	3.10.2024	Data engineer and analysts with project manager	Meeting with a 3rd company for subscription of a database, providing them with requirements for the structure of the database.
Database contract	4.10.2024	Project manager	Confirming the deal for the database, signing the contract.
Code preparation (1)	7.10.2024	Software developers	Presenting an insight into the data structures used for the system, and method of initialization.
Code preparation (2)	8.10.2024	Software developers	Covering the main functions, their purpose, and explaining the logic behind them along with the outcome of said functions.
Code preparation (3)	9.10.2024	Software developers	Summary of the code, questions, and critiques from the company, confirming the plan.

Activity	**Date of Activity**	**Person/Team Responsible**	**Description**
Technical design preparation (1)	10.10.2024	Software developers, Web developers/designers	Discussing the requirements and wishes of the company for the technical design and user interface.
Technical design preparation (2)	11.10.2024	Software developers, Web developers/designers	Brainstorming, coming up with a look for the design.
Technical design preparation (3)	14.10.2024	Software developers, Web developers/designers	Confirming the final concept for the design, presenting it to the client, receiving permission to start.
Team meeting	15.10.2024 - 16.10.2024	Full team	Finishing touches of the development stages, preparation for the presentation of Phase 1 to Happy Fruit.
Development summary to client	17.10.2024	Full team	Presenting a summary/recap of the plan for the development of the system, approval of Happy Fruit's management team.

6.2 Phase 4 Development

Sprint no.	Activity	Person/Team Responsible	Description
1	System development recap/overview	Full team	Setting the groundwork, going over each step and the way it will be achieved.
1	System development ideas	System architect	Brainstorming for ideas on how to construct the system, confirming the chosen plan.
1	Database diagram	Data engineer and analysts	Creating a schema of the desired database.
1	Database development (1)	Data engineer and analysts	Constructing the tables/entities that need to be tracked and updated using "postgreSQL", defining the relationships between them.
1	Database development (2)	Data engineer and analysts	Filling the tables with data.
- - - - - - - Demo run (1) - - - - - - -	Full team	1d	Re-visiting the system plan, testing the functionality of the database.
2	Code development (1)	Software developers	Creating the functions for: setting the best route at the moment.
2	Code development (2)	Software developers	Creating the functions for: keeping track of required deliveries, processing user input (drivers, warehouse employees, managers).
2	Code development (3)	Software developers	Creating the functions for: calculating the storage of warehouses and amount needed.

Sprint no.	**Activity**	**Person/Team Responsible**	**Description**
2	Code development (4)	Software developers	Finalizing the code, finishing touches, fixing syntax.
2	Database + code	Software developers and Data engineer	Integrating the database into the BackEnd of the app with SQL functions and procedures.
- - - - - - - Demo run (2) - - - - - - -	Full team	1d	Database revisiting, code test run.
3	UI development (1)	Web designers/developers	Brainstorming for look and feel of the website, creating moldboard.
3	UI development (2)	Web designers/developers, software developers	Designing the layout, implementing the brand identity of "Happy Fruit".
3	UI development (3)	Web designers/developers, software developers	Implementing authentication for truck drivers, warehouse keepers, and administrators.
3	Code + Website	Web designers/developers, software developers	Combining the code with the user interface.
- - - - - - - Demo run (3) - - - - - - -	Full team	1d	Testing the website, correcting errors.

7 Conclusion

The development of this program allows Happy Fruit to utilize a reliable and efficient application to schedule shipment deliveries to and from their warehouses and stores. The development team worked cohesively during development to make key design decisions and follow a collectively decided upon methodology. The technical report provides further details for the user Happy Fruit, and ensures that the application is well understood. This project meets the initial requirements for Happy

Fruit's current business needs, and lays a solid foundation for any future logistical requirements.