Comprehensive Exercise Report

Team <<X>> of Section <<000>>

Robertas Gaigalas – 221ADB186

Oleksii Zakharov – 221ADB091

Sofiia Azhniuk – 221ADB207

Andrii Bazanov – 221ADB086

Andrii Kuznetsov – 221ADB102

Veronika Stasiuk – 221ADB092

Oleksandr Tokarchuk – 221ADB090

Requirements/Analysis

Responsible person: Sofiia Azhniuk

Week 2

Journal

Short description of the project:
The project is a production tracking system designed to improve workflow visibility and efficiency. It allows real-time monitoring of the production process at the steel frame structure production factory.

Project`s requirements:

- project management system with unique project IDs, client information, responsible persons, and project timeline
- Component tracking system for individual parts within projects (beams, pipes, nuts, bolts, etc.)
  - Each component must have specifications (type, dimensions, weight)
  - Support for hierarchical structure (projects → assemblies → parts)
- Container management system
  - Ability to track which components are packed in which containers
  - CMR documentation support
- Barcode scanning functionality for part status tracking
- Client management system (company details, VAT numbers)
- Contact management for client representatives
- Shipping address management
- Worker information tracking (ID, name, role)
- Employee card system for attendance tracking

Questions for the client:

1. How many people will use that application?
   a. Around 30 people - a few engineers, an administrator who checks for attendance, and about 20 workers in the production facility.
2. Do you already have a similar system now?
   a. No, we currently have many Excel sheets. Everything is in different places and it is hard to manage how the production goes, so we believe that the proposed system will improve our workflow.
3. How do you track the progress of the production now?
   a. We mostly rely on our head foreman who keeps track of everything. He tries to maintain an overall picture of what's happening in production, but sometimes it's hard to answer in a short period.
4. What is better for you to use: a scanner or a phone?
   a. We can’t ask workers to use their phones because if they get damaged during work, we will be liable for all damages. If it’s possible to use dedicated scanners, it would be great as they are more reliable in the industrial environment and designed specifically for this purpose.

Topics of the projects we are unfamiliar with:

- Barcode implementation
    - Zebra software for barcode scanning https://www.zebra.com/us/en/software/scanner-software.html
    - Theory https://www.geeksforgeeks.org/what-is-a-barcode-reader/
    - Theory https://www.geeksforgeeks.org/difference-between-barcode-and-qr-code/

- PostgreSQL Database
    - Reference: https://www.postgresql.org/docs/

- Backend Integration
    - Focus on connecting PostgreSQL with Node.js/Express backend https://www.youtube.com/watch?v=sDPw2Yp4JwE
    - Reference https://www.youtube.com/watch?v=SfoSVdQJd6w
    - How to do it reference https://medium.com/@pawel66556655/creating-a-vanilla-node-js-rest-app-with-postgresql-aa1a66fcb0b7

The users of this software

- Engineers: Technical staff who need to create and manage projects, track progress, and generate reports.
- Production workers: Factory floor staff who need to scan barcodes and update production status through a simple interface
- Management: Factory director who needs to monitor overall production performance and assign permissions to other workers
- Logistics staff: Transportation person responsible for delivery management and shipping documentation

How each user would interact with the software

- Engineers: create projects, monitor project progress, add technical specifications, update project statuses, generate technical reports, manage container assignments, generate CMR documentation
- Production workers: use barcode scanning to update component status
- Management: View production statistics, monitor overall factory performance, access progress reports, analyze workflow efficiency
- Logistics staff: Manage container packing lists, create shipping documentation (CMR), track deliveries, update shipping statuses

What features must the software have? What should the users be able to do?
Software should have several components, which are divided into two groups.

Projects details:

- Projects: Monitor overall projects, timelines, and status
- Components: Track individual parts and assemblies
- Operations: Production workflow and task management
- Containers: Packing lists and shipping units

Administrative details:

- Clients: Client company information and details
- Contacts: Contact persons and representatives
- Delivery Addresses: Shipping locations and details
- Workers: Employee information and roles
- Cards: Employee ID cards for time tracking

Other notes:
The project should be customizable and have only needed functions.

Software Requirements

Detailed description of the project

This project aims to develop a production tracking system for UAB "Baltic Construction Solutions Lietuva", a metal construction manufacturing company. The system will digitize their current manual tracking process, replacing Excel sheets and paper-based workflow with a modern web-based solution. The main goal is to improve production visibility and efficiency by implementing barcode tracking for components and providing real-time status updates throughout the manufacturing process. It will serve approximately 30 users, including engineers, production workers, and administrative staff, utilizing dedicated barcode scanners for production tracking.

Software Requirements Specification (SRS):

- Functional requirements
    - Project creation and management
    - Component tracking with barcodes
    - Status updates and monitoring
    - Report generation or chart generation
    - User authentication
- Non-functional requirements
    - Less than 3 seconds for the response time
    - Role-based authentication
    - Backup and recovery
- External requirements
    - Barcode scanner integration
    - Database connectivity
    - Mobile device compatibility
    - Export to PDF/Excel
- System features
    - Multi-User support
    - Real-time updates
    - Data backup
    - Role-based access

System Requirements Specification (SysRS):

- System capabilities
    - Support for 25+ concurrent users
    - Real-time data processing
    - Multi-device support
- Interfaces
    - Web interface
    - Mobile interface
    - Scanner interface
- Performance
    - Response times
    - System availability
    - Data processing speed

- Security
    - Access control
    - Data encryption
    - Audit logging
- Hardware expectations
    - Server requirements
    - Network Infrastructure
    - Scanner compatibility
- System acceptance criteria
    - User acceptance testing
    - Performance benchmarks
    - Security Validation

Black-Box Testing

Instructions: Week 4

Journal

Remember: Black box tests should only be based on your requirements and should work independent of design.

The following prompts are meant to aid your thought process as you complete the black box testing portion of this exercise. Please review your list of requirements and respond to each of the prompts below. Feel free to add additional notes.

- What does input for the software look like (e.g., what type of data, how many pieces of data)?
  - <<Insert answer>>
- What does output for the software look like (e.g., what type of data, how many pieces of data)?
  - <<Insert answer>>
- What equivalence classes can the input be broken into?
  - <<Insert answer>>
- What boundary values exist for the input?
  - <<Insert answer>>
- Are there other cases that must be tested to test all requirements?
  - <<Insert answer>>
- Other notes:
  - <<Insert notes>>

Black-box Test Cases

Use your notes from above to complete the black-box test plan section of the formal documentation by writing black box test cases (other than actual results since no program currently exists). Remember to test each equivalence class, boundary value, and requirement.

Test ID	**Description**	**Expected Results**	**Actual Results**

Design

Instructions: Week 6

Journal

Remember: You still will not be writing code at this point in the process.

The following prompts are meant to aid your thought process as you complete the design portion of this exercise. Please respond to each of the prompts below and feel free to add additional notes.

- List the nouns from your requirements/analysis documentation.
    - <<Insert answer>>
- Which nouns potentially may represent a class in your design?
    - <<Insert answer>>
- Which nouns potentially may represent attributes/fields in your design? Also list the class each attribute/field would be a part of.
    - <<Insert answer>>
- Now that you have a list of possible classes, consider different design options (***lists of classes and attributes***) along with the pros and cons of each. We often do not come up with the best design on our first attempt. Also consider whether any needed classes are missing. These two design options should not be GUI vs. non-GUI; instead you need to include the classes and attributes for each design. Reminder: Each design must include at least two classes that define object types.
    - <<List at least two design options with pros and cons of each>>
- Which design do you plan to use? Explain why you have chosen this design.
- List the verbs from your requirements/analysis documentation.
    - <<Insert answer>>
- Which verbs potentially may represent a method in your design? Also list the class each method would be part of.
    - <<Insert answer>>
- Other notes:
    - <<Insert notes>>

Software Design

<<Use your notes from above to complete this section of the formal documentation by planning the classes, methods, and fields that will used in the software. Your design should include UML class diagrams along with method headers. ***Prior to starting the formal documentation, you should show your answers to the above prompts to your instructor.***>>

Implementation

Instructions: Week 8

Journal

The following prompts are meant to aid your thought process as you complete the implementation portion of this exercise. Please respond to each of the prompt below and feel free to add additional notes.

- What programming concepts from the course will you need to implement your design? Briefly explain how each will be used during implementation.
    - <<Insert answer>>
- Other notes:
    - <<Insert notes>>

Implementation Details

<<Use your notes from above to write code and complete this section of the formal documentation with a README for the user that explains how he/she will interact with the system.>>

Testing

Instructions: Week 10

Journal

The following prompts are meant to aid your thought process as you complete the testing portion of this exercise. Please respond to each of the prompts below and feel free to add additional notes.

- Have you changed any requirements since you completed the black box test plan? If so, list changes below and update your black-box test plan appropriately.
  - <<Insert answer>>
- List the classes of your implementation. For each class, list equivalence classes, boundary values, and paths through code that you should test.
  - <<Insert class>>
    - <<Insert needed tests>>
  - <<Insert class and tests for each class>>
- Other notes:
  - <<Insert notes>>

Testing Details

<<Use your notes from above to write your test programs and complete this section of the formal documentation by creating a list of your test programs along with descriptions of what they are testing. You will also complete the black-box test plan by running the program and filling in the Actual Results column.>>

Presentation

Instructions:Week 12

Preparation

The following prompts are meant to aid your thought process as you complete the presentation portion of this exercise. It is recommended that you examine the previous sections of the journal and your reflections as you work on the presentation as it is likely that you have already answered some of the following prompts elsewhere. Please respond to each of the prompts below and feel free to add additional notes.

- Give a brief description of your final project
    - <<Insert answer>>
- Describe your requirement assumptions/additions.
    - <<Insert answer>>
- Describe your design options and decision. How did you weigh the pros and cons of the different designs to make your decision?
    - <<Insert answer>>
- How did the extension affect your design?
    - <<Insert answer>>
- Describe your tests (e.g., what you tested, equivalence classes).
    - <<Insert answer>>
- What lessons did you learn from the comprehensive exercise (i.e., programming concepts, software process)?
    - <<Insert answer>>
- What functionalities are you going to demo?
    - <<Insert answer>>
- Who is going to speak about each portion of your presentation? (Recall: Each group will have ten minutes to present their work; minimum length of group presentation is seven minutes. Each student must present for at least two minutes of the presentation.)
    - <<Insert answer>>
- Other notes:
    - <<Insert notes>>

<<Use your notes from above to complete create your slides and plan your presentation and demo.>>