

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра Програмного забезпечення комп'ютерних систем

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
магістра

(назва освітньо-кваліфікаційного рівня)

студента	<i>Федорченко Владислава Сергійовича</i> (ПІБ)
академічної групи	<i>122м-21-2</i> (шифр)
спеціальності	<i>122 Комп'ютерні науки</i> (код і назва спеціальності)
освітньої програми	<i>Комп'ютерні науки</i>
на тему:	<i>Розробка програмного додатку у пакеті STEP 7 для роботи зі SCADA-системою на базі OPC-інтерфейсу</i>

В.С. Федорченко

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтин говою	інституційною	
розділів кваліфікаційної роботи				
спеціальний	<i>Проф. Слесарев В.В.</i>			

Рецензент	<i>Доц. Кожевніков А.В.</i>			
-----------	-----------------------------	--	--	--

Нормоконтролер	<i>Проф. Лактіонов І.С.</i>			
----------------	-----------------------------	--	--	--

Дніпро
2022

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

Завідувач кафедри

Програмного забезпечення комп'ютерних систем

(повна назва)

М.О. Алексєєв

(підпис)

(прізвище, ініціали)

« » ____

20 22 Року

ЗАВДАННЯ

**на виконання кваліфікаційної роботи
магістра**

спеціальності

122 Комп'ютерні науки

(код і назва спеціальності)

студенту

122М-21-2

(група)

Федорченко Владиславу Сергійовичу

(прізвище та ініціали)

Тема кваліфікаційної роботи

Розробка програмного додатку у пакеті STEP 7
для роботи зі SCADA-системою на базі OPC-
інтерфейсу.

1 ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Наказ ректора НТУ «Дніпровська політехніка» від 31.10.2022 р. № 1200-с

2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень – процес керування інформаційними потоками.

Мета роботи: керування технологічними процесами зерносховища на базі SCADA-системи.

3 ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Сучасний елеваторний комплекс або зерносховище має в своєму складі велику кількість обладнання і вимагає простого централізованого керування технологічними процесами. Це забезпечується застосуванням сучасних систем автоматизованого керування.

Розроблено функціональну схему автоматизації системи та схему інформаційно-матеріальних потоків. Складено алгоритм для керування виконавчими механізмами. Описано реалізацію SCADA-системи. Досліджено параметри регулятора кліматозабезпечення зерносховища. Обрано давачі та

виконуючі механізми, описана структура системи керування.

4 ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	Строки виконання робіт (початок – кінець)
Аналіз існуючих рішень та постановка задачі роботи	12.09.2022-30.09.2022
Аналіз засобів для роботи зі SCADA-системою на базі OPC-інтерфейсу	01.10.2022-31.10.2022
Розробка програмного забезпечення та дослідження ефективності запропонованих рішень.	01.11.2022-10.12.2022

Завдання видав

_____ (підпис)

Слесарєв В.В.

_____ (прізвище, ініціали)

Завдання прийняв до виконання

_____ (підпис)

Федорченко В.С.

_____ (прізвище, ініціали)

Дата видачі завдання: 10.09.2022 р.

Термін подання до ЕК 22.12.2022 р.

РЕФЕРАТ

Пояснювальна записка: 61 стор., 9 рис., 3 таблиці, 2 додатка, 18 джерел.

Об'єкт досліджень – процес керування інформаційними потоками.

Мета роботи: керування технологічними процесами зерносховища на базі SCADA-системи.

Сучасний елеваторний комплекс або зерносховище має в своєму складі велику кількість обладнання і вимагає простого централізованого керування технологічними процесами. Це забезпечується застосуванням сучасних систем автоматизованого керування.

Розроблено функціональну схему автоматизації системи та схему інформаційно-матеріальних потоків. Складено алгоритм для керування виконавчими механізмами. Описано реалізацію SCADA-системи. Досліджено параметри регулятора кліматозабезпечення зерносховища. Обрано датчики та виконуючі механізми, описана структура системи керування.

СПИСОК КЛЮЧОВИХ СЛІВ: БАЗА ДАНИХ, ТЕСТУВАННЯ, ФРЕЙМВОРК, SCADA-СИСТЕМА, ЛОГІЧНИЙ КОНТРОЛЕР.

ABSTRACT

Explanatory note: 61 pages, 9 figures, 3 tables, 2 appendices, 18 sources.

The object of research is the process of managing information flows.

The purpose of the work: management of technological processes of the grain storage on the basis of the SCADA system.

A modern elevator complex or granary contains a large amount of equipment and requires simple centralized management of technological processes. This is ensured by the use of modern automated control systems.

A functional scheme of system automation and a scheme of information and material flows have been developed. An algorithm for controlling executive mechanisms has been developed. The implementation of the SCADA system is described. The parameters of the grain storage climate control regulator were studied. Sensors and executing mechanisms are selected, the structure of the control system is described.

LIST OF KEYWORDS: DATABASE, TESTING, FRAMEWORK, SCADA SYSTEM, LOGIC CONTROLLER.

ЗМІСТ

РЕФЕРАТ.....	4
ABSTRACT.....	5
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ РОБОТИ.....	9
1.1. Технологічний процес зберігання зерна.....	9
1.2. Схема інформаційно-матеріальних потоків.....	18
1.3. Постановка задачі роботи.....	21
РОЗДІЛ 2. МЕТОДИ ТА ТЕХНОЛОГІЇ ВИРІШЕННЯ ЗАДАЧІ.....	23
2.1. Архітектура програмного забезпечення.....	23
2.2. Вибір програмних додатків	24
2.3. Вибір мов програмування	27
2.4. Опис архітектури мережі.....	31
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ЙОГО ЗАСТОСУВАННЯ.....	32
3.1. Вибір ПЛК та інтерфейсів.....	32
3.2. Програма керування процесом зберігання зерна	37
3.3 Створення бази даних	26
3.4. Реалізація та тестування програмного продукту	29
ВИСНОВКИ.....	43
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	45
Додаток А. Код програми.....	46
Додаток Б. Перелік файлів на диску.....	61

ВСТУП

Збереження та раціональне використання всього вирощеного врожаю, отримання максимальної кількості виробів із сировини сьогодні є одним із найважливіших державних завдань.

Продукти харчування, що виробляються із зерна злакових рослин (печений хліб, крупа, макаронні та інші вироби з борошна), є складовою частиною їжі людини. Зерна і насіння злакових рослин безперечно мають вплив на життя кожної людини. Аналіз споживання продовольства у світі показує, що приблизно 55% білків, 70% вуглеводів і 15% жирів доводиться на частку зерна і насіння. Крім того, вони є важливим концентрованим кормовим засобом і, в певній мірі, технічною сировиною.

Зберігання, що є фінальною фазою виробництва зерна, - це наука, яка вивчає особливості зерна і зернових мас в цілому як об'єктів зберігання, а також вплив фізичних, хімічних і біологічних факторів на стан зерна. Зберігання зерна і зернових продуктів вимагає грандіозної матеріально-технічної бази і спеціалістів, які володіють достатніми знаннями та досвідом в цій області [1].

Автоматизація виробничих процесів - основний напрямок, по якому в даний час просувається виробництво в усьому світі. Все, що раніше виконувала сама людина, її функції, не тільки фізичні, але і інтелектуальні, поступово переходять до техніки, яка сама виконує технологічні цикли і здійснює контроль за ними.

Впровадження новітніх інформаційних технологій та останніх розробок в сфері промислової автоматизації підприємств зберігання і переробки зерна має значний вплив на підвищення ефективності керування, зростання конкурентоспроможності підприємств в цілому.

Зберігання і переробка зерна - це складний, багатоступінчастий, енергоємний процес, який вимагає використання досконалих, надійних систем автоматизації зберігання і переробки зерна для досягнення високої ефективності роботи даної галузі.

Дана робота присвячена створенню SCADA-системи керування

технологічним процесом зберігання зерна в силосі. А саме підтриманню основних технологічних параметрів на необхідному для збереження якості зерна рівні при умові забезпечення енергоефективності. Також приділено необхідну увагу питанням охорони праці.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ РОБОТИ

1.1 Технологічний процес зберігання зерна.

Ефективне функціонування зерноскладів залежить від складних фізіологічних та біохімічних процесів, які відбуваються в зерні під час його зберігання, а також від чинників, що їх обумовлюють.

Враховуючи, що дозрівання зерна тривале у часі, в процесі зберігання в ньому відбувається явище післязбирального досягання. Воно протікає досить повільно і зазвичай супроводжуються посиленням дихання зерна і виділенням певної кількості вологи, яка повинна випаровуватися в навколишнє середовище. В іншому випадку зерно надмірно зволожується. Зберігання вологого зерна без належного провітрювання призводить до його швидкого псування [2].

Саме підвищені показники вологості є основною небезпекою для зерна, що зберігається, вони впливають на швидкість протікання фізіологічних процесів, які відбуваються у зерні під час його зберігання. Якщо зерно сухе, то і фізіологічні процеси відбуваються в ньому досить повільно та майже непомітно. У такому стані зерно перебуває в стані спокою. При різкому підвищенні вологості понад 14% в зерні починає посилюватися і поступово наростає процес дихання. Його можна порівняти з процесом горіння - зерно поглинає з навколишнього середовища кисень і виділяє вуглекислоту, вологу і теплову енергію. Якщо існує брак вільного кисню, то процес дихання відбувається за рахунок кисню, що створюється з вуглеводів зерна. При всьому цьому в зерні розпочинаються процеси, що максимально наближаються до спиртового бродіння. Як результат, погіршується якість зерна [3].

При різкому зниженні температурних показників всі життєдіяльні процеси в зерні уповільнюються, в протилежному випадку дихання зерна лише посилюється.

Підвищена вологість зерна і висока температура викликають теж

розвиток та прискорення процесів розмноження різноманітних мікроорганізмів. Частіше за все у таких умовах розвиваються цвілеві грибки і бактерії, для яких подібні умови вважаються сприятливими. Вони досить швидко псують зерно [4].

При проектуванні зерноскладів і при розробці окремих їх деталей слід враховувати теж фізичні властивості зерна: його високі показники гігроскопічності, низьку теплопровідність, слабку повітро- і газопроникність, сипкість, здатність самосортуватись при запуску в глибокі склади і самоущільнюватись під впливом своєї ж ваги.

Найбільш результативним заходом підготовки зерна до довготривалого зберігання є природний або штучний процес сушіння. Рекомендується проводити процедуру очищення зерна від органічних та неорганічних сторонніх включень (сміттєве насіння, бите зерно, пісок та інше).

Як видно з аналізу процесу зберігання зерна, цей процес нестаціонарний у часі. Він може залежати як від конструкції апарату для зберігання, так і від фізіологічного і біологічного стану сировини. Будь-якими процесами, що відбуваються в зерні, весь час необхідно керувати і контролювати їх. Через те, що всі параметри зерна змінні у часі, регулятори, що використовуються в системі керування повинні бути адаптивними. При цьому можуть бути використані наступні методи зберігання: аерація, вентилявання в декількох режимах, хімічне знезараження, а також консервування. Метод обирається відповідно стану і призначення зерна.

Аерація – пасивне або примусове (штучне) провітрювання силосу з зерном. Аерація необхідна для очищення повітря та оздоровлення зерна від продуктів розпаду і дихання зернової маси (діоксиду вуглецю, етилену, водяної пари). Спеціалісти не радять зберігати зерно насипом в наземних зерноскладах, які не оснащені активною вентиляцією. При аерації слід контролювати та керувати станом рівноважної вологості зерна. Особливо актуально це для периферійних шарів, де проникність повітря мінімальна [5].

Вентилювання – це процес продування повітря через зернову масу з метою охолодження або підсушування вологого зерна. Як наслідок,

сировина насичується киснем. У режимі підсушування вентиляції проводиться у тому випадку, коли фактична вологість зерна досягає показників вище рівноважних. Вентиляція необхідна для покращення якості сировини та прискорення післязбирального дозрівання.

Ця методика обробки зерна дає можливість запобігти ймовірності самозігрівання зерна, охолоджувати його до показників, що забезпечують найтриваліший термін зберігання.

Вентиляція звичайним не підігрітим та не охолодженим повітрям дає можливість змінювати температурні показники зерна шляхом пропускання його через зернову масу. Більше того, цей процес дещо підсушує зерно. А інтенсивність підсушування залежить від потужності вентилятора.

У міру вентиляції зернового насипу проходить переміщення зони охолодження (навесні – теплої зони). Напрямок руху даної зони залежить перш за все від режиму функціонування вентилятора. Якщо він направляє повітря в сховище, то зона охолодження (нагрівання) рухається вгору. Якщо ж вентилятор висмоктує повітря зі сховища, то ця зона розпочинає рухатися згори до низу.

Вентиляція вважається більш ефективною, ніж перелопачування і пересипання зерна з одного бункера в інший. Окрім вищевказаного, вона не призводить до додаткового травмування сировини, внаслідок чого сповільнюється процес появи цвілі. При вентиляції головне правильно зафіксувати температурні показники і вологість повітря як зовні, так і всередині силосу. Відхилення від найкращих режимів вентиляції може привести до введення в зерно дуже вологого повітря [6].

Охолодження представляє собою прийом інтенсивного вентиляції, який збільшує стійкість і довговічність зерна. Воно також пригнічує будь яку життєдіяльність в сховищі, а також робить неможливим розмноження і розвиток зернових шкідників та навіть патогенних мікроорганізмів. Реалізується охолодження за допомогою використання устаткування для активного вентиляції в складах, повітряних силосних системах або шляхом пропуску сировини через охолоджувальні шахти сушарок. Найбільш ефективно охолоджувати зерно спеціальним підготованим охолодженим

повітрям. Але в силу високої фінансової вартості даний спосіб рекомендується використовувати лише для цінних партій зерна продовольчого і насінневого призначення [7].

Для того, щоб мінімізувати втрати зерна під час довготривалого зберігання у сховищі, необхідно вміти правильно підбирати тип установки. Усі сховища для зберігання зерна умовно можна розділити на три типи: склад, бетонний силос і металевий силос. Кожен із них відрізняється функціональною спрямованістю і технологічністю.

Вимоги, яким повинні відповідати зерносховища

Необхідні якості зерно при тривалому зберіганні може зберегти лише в правильно влаштованих зерноскладах, вимоги до яких залежать від вищеописаних властивостей зерна. Зерносклади будь-якого з існуючих типів зазвичай конструюють неопалювальними, без горищних перекриттів. У кожному зерноскладі мають бути ліквідовані будь-які причини, що можуть викликають хвороби зерна: вони мають бути сухими, чистими, добре вентильовані, та захищеними від потрапляння гризунів, птахів, комах та інших шкідників зерна. Також слід звернути увагу на те, щоб силос був ретельно захищеним від проникнення в нього атмосферних опадів, поверхневих та ґрунтових вод [8].

Внутрішнє планування зерносховищ, будова, форма і величина ємностей для зберігання зернової сировини (засіків, бункерів, відсіків або силосів), їх розміщення в сховищах повинні забезпечувати вільний вхід до зерна, що дозволить у будь-який момент спостерігати за його станом і можливістю внутрішньоскладської обробки зерна в момент зберігання. У зерноскладах мають бути можливі очищення, огляд і дезінфекція окремих частин конструкції, внутрішнього устаткування та механізму в цілому.

Технологічні процеси, пов'язані з використанням зерносховищ (завантаження, вивантаження, обробка, пересипання зерна та ін.) повинні бути повністю автоматизовані та механізовані із використанням як стаціонарних, так і пересувних агрегатів і з максимальним застосуванням принципу самопливу зерна.

Механічне обладнання зерноскладів

Для механізації завантаження, розвантаження, пересипання і підробітку зерна в зерноскладах використовуються наступні стаціонарні та пересувні агрегати:

- 1) стрічкові норії (нерідко називають елеваторами або самотягами) для вертикального підйому сировини;
- 2) стрічкові стаціонарні конвеєри для переміщення зерна в горизонтальному напрямку або під незначним кутом;
- 3) пересувні конвеєри для вантажно-розвантажувальних процедур переважно при зберіганні зерна на підлозі;
- 4) самоподавачі для завантаження конвеєрів при підлоговому завантаженні зерна;
- 5) гвинтові конвеєри, або шнеки для переміщення зерна на невеликій відстані;
- 6) самопливні зернопроводи для переміщення зерна зверху вниз під впливом сили тяжіння;
- 7) зерноочисні механізми і сепаратори, необхідні для очищення зерна від сторонніх включень органічного і неорганічного типу.

Розглянемо детальніше кожен з типів зерносховищ:

1. Склад

Склади для зберігання зерна представляють собою одноповерхові приміщення з повністю горизонтальними або ж похилими підлогами. Стіни можуть бути виконані із цегли, каменю або ж залізобетону. Зерно в складах такого типу зберігається насипом на підлозі або ж в засіках. Різні методи зберігання зерна, розміри господарств і різновиди культур визначили появу великої кількості видів та розмірів зерносховищ. Зернові склади бувають механізованими та немеханізованими.

Немеханізовані варіанти складів будують лише з горизонтальними підлогами. Переміщення і відпуск зерна в даних складах здійснюють із використанням пересувних і самохідних агрегатів.

Механізовані склади можуть бути побудовані як з горизонтальними, так

і з похилими підлогами. Ці склади оснащують верхніми (завантажувальними) і нижніми (розвантажувальними) стаціонарними стрічковими транспортерами і норіями. Встановлюється устаткування зазвичай в торцях складів.

Зерно в таких складах зазвичай зберігають насипом. У спеціальній тарі, в якості якої зазвичай виступають мішки, зберігають тільки окремі партії зерна насінневого призначення, а також насіння з тендітною структурою оболонок.

Щоб раціональніше експлуатувати зерносклади і здешевити процедуру зберігання зерна, слід найбільш повно використовувати його об'єм, максимально збільшуючи висоту зернового насипу. Гранично допустима висота насипу в складах такого типу визначається не характеристиками складу, а станом зерна, його цільовим призначенням, технічним станом і видом складу. Крім того може бути врахований термін зберігання і пора року, коли зерно буде знаходитися у складі [9].

Продовольче і фуражне зерно, сухе чи середньої сухості, що пройшло стадію післязбирального дозрівання і все ретельно очищене від домішок, можна зберігати в підлогових зерноскладах шаром до 5 м, а в підлогових зерноскладах з похилими підлогами, обладнаних механізмами, що забезпечують механізовану завантаження, розвантаження та обробку зерна, - шаром до 10 м.

Сухе насіннєве зерно вологістю не вище 14% з метою збереження схожості насіння між собою та енергії їх проростання зберігають зазвичай розсипом на підлозі або в бункерах і засіках. Висота насипу може складати 2-3 метри, також зерно може зберігатися в тарі (мішках) штабелями. Висота має дорівнювати не більше 8 рядів, причому значно менша висота насипу чи штабеля при всіх інших рівних умовах можуть бути призначені для насіння олійних культур, а також для проса, рису, гороху і кукурудзи. Це пояснюється меншими показниками стійкості при зберіганні, аніж у пшениці чи у жита. Якщо зерносклад оснащений активною вентиляцією, висота насипу насінневого зерна може бути збільшена до 5 м.

Зерно, показник вологості якого складає понад 16-17%, та яке ще не пройшло післязбирального дозрівання і, отже, має знижену стійкість, в залежності від пори року і температури можна зберігати насипом незначної висоти, що не перевищує 1-1,5 м.

На показники висоти насипу може впливати також передбачуваний строк зберігання зерна. Якщо зерно закладають для нетривалого зберігання, то висоту насипу можна дещо збільшити. При тривалому ж зберіганні слід застосовувати значно нижчу висоту насипу. Висоту насипу зерна можна варіювати теж в залежності від пори року. Її можна збільшувати на холодний період та знижувати з настанням перших високих температурних показників.

Елеватор - це найбільш досконалий тип механізованого сховища. Він призначається перш за все для зберігання сухого товарного зерна зі встановленою вологістю не вище 14-15%. Зерно в елеваторах зберігають в силосах, які розташовані поблизу один від одного. Всі трудомісткі процеси в елеваторах - прийом зерна, його зважування, завантаження та вивантаження, внутрішнє пересування, очищення від домішок, сортування і т.д.- повністю механізовані й автоматизовані в елеваторах.

Елеватор означає власне підйомник, так як основним механізмом в такого типу зерносховищах є елеватор-підйомник. Саме тому ця назва розповсюдилася і на всю споруду. Крім норій, застосовуваних для вертикального підйому зерна, елеватори оснащені конвеєрами і шнеками, що необхідні для горизонтального переміщення зерна, трубами для переміщення зерна самопливом зверху вниз і зерноочисними машинами і агрегатами [10].

Зерно, показник вологості якого складає понад 16-17%, та яке ще не пройшло післязбирального дозрівання і, отже, має знижену стійкість, в залежності від пори року і температури можна зберігати насипом незначної висоти, що не перевищує 1-1,5 м.

На показники висоти насипу може впливати також передбачуваний строк зберігання зерна. Якщо зерно закладають для нетривалого зберігання, то висоту насипу можна дещо збільшити. При тривалому ж зберіганні слід застосовувати значно нижчу висоту насипу. Висоту насипу зерна можна

варіювати теж в залежності від пори року. Її можна збільшувати на холодний період та знижувати з настанням перших високих температурних показників.

Елеватор - це найбільш досконалий тип механізованого сховища. Він призначається перш за все для зберігання сухого товарного зерна зі встановленою вологістю не вище 14-15%. Зерно в елеваторах зберігають в силосах, які розташовані поблизу один від одного. Всі трудомісні процеси в елеваторах - прийом зерна, його зважування, завантаження та вивантаження, внутрішнє пересування, очищення від домішок, сортування і т.д.- повністю механізовані й автоматизовані в елеваторах.

Головні будівлі та споруди елеватора

Сучасний елеватор складається із комплексу споруд, зв'язаних спільними виробничими процесами. Головними з них є приймання, зважування, зберігання, відпуск зерна. Серед спеціальних функцій виділяють очищення, сушіння та сортування зерна.

Будинки й споруди елеватора за функціональними властивостями можна умовно розділити на: виробничі, необхідні для приймання, зберігання, підробітки та відпуску зерна та іншої зернової продукції; допоміжні та обслуговуючі процес виробництва; невиробничі.

До головних виробничих будівель та споруд елеваторів можна віднести: робочу будівлю; силосні корпуси з конвеєрними галереями; споруди призначені для розвантаження зерна з залізничного, автомобільного і водного видів транспорту і навантаження сировини на ці ж види транспорту; устаткування для сушіння зерна; агрегати для зберігання і навантаження відходів зернового виробництва на автомобільний та залізничний транспорт.

У робочій споруді елеватора зазвичай розміщують машини і механізми, необхідні для підйому зерна (норії), зважування, очищення, а також прилади для переміщення і розподілу зерна. Робочу будівлю вважають основною в системі елеватора, навколо якого згруповуються і з яким пов'язують всі інші його виробничі споруди.

Силосний корпус - це власне зерносховище, що може складатися з різної кількості силосів.

До складу елеватора можуть входити й інші допоміжні виробничі будівлі та споруди, такі, як спеціальні будівлі та споруди для якісного очищення і сортування зерна, камера для швидкого збору пилу, цех для відходів, склади для підлогового зберігання зерна і т.д.

До допоміжних обслуговуючих виробництво будівель і споруд елеватора слід віднести: силову станцію, склади для зберігання палива, ремонтні майстерні, пожежне депо, лабораторія і т. д. В якості невиробничих будівель і споруд виступають - їдальня, побутові пристрої, адміністративний корпус та ін.

Комплекс будівель і споруд елеватора може бути різним і залежить від типу елеватора, виконуваних функцій і обсягу робіт [8].

Незначна маса конструкцій сталевих силосів дає можливість різко зменшити транспортні витрати, скоротити витрати на будівельні машини, особливо при будівництві в глибинних районах. Також така характеристика дає можливість зводити елеватори на будівельних ділянках з незначною несучою здатністю ґрунтів.

Конструкції сталевих силосів реально виготовити повністю в умовах заводу, доставляти на будівельний майданчик укрупненими елементами, що дозволить вести швидкий монтаж їх і скоротити строк зведення [10].

Величини сталевих силосів можуть бути найрізноманітнішими. Є безліч різних прийомів виготовлення і монтажу сталевих стінок силосів.

Головні переваги металевих силосів:

- 1) Швидкий процес монтажу;
- 2) Значно менший фундамент;
- 3) Стандартне виконання;
- 4) Можливість переміщення в інше місце;
- 5) Гнучкість в плані адаптації та різних модифікацій;
- 6) Конструкція дна має конічну або плоску форму;
- 7) Можливість застосування зачисних шнеків.

Зберігання зерна в металевих силосах елеваторів прийнято вважати самим модернізованим та індустріальним.

1.2 Схема інформаційно-матеріальних потоків

На основі аналізу технологічного процесу зберігання зерна в силосі розробимо схему інформаційно-матеріальних потоків (рис. 1.3) – спосіб наочного уявлення маршрутів, потоків управлінської інформації між складовими системи керування.

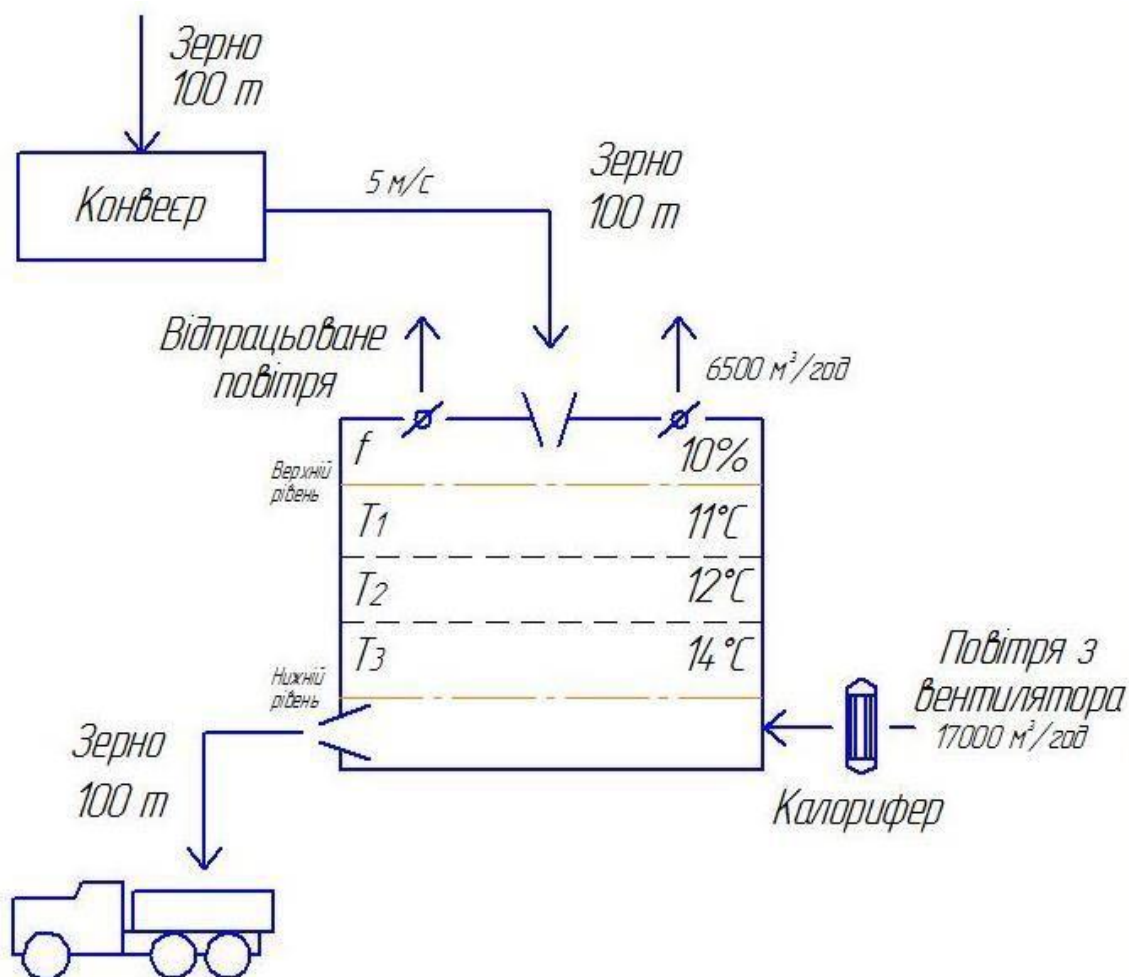


Рис. 1.1. Схема інформаційно-матеріальних потоків

Для реалізації завантаження та зерна в системі передбачений конвеєр. Зерно через засувку потрапляє на конвеєр та далі в силос через завантажувальний отвір в даху. Необхідно і достатньо мати уявлення про нижній і про верхній рівні зерна, для того щоб точно знати коли завершувати завантаження або вивантаження зерна. Загрузка зерна припиняється після спрацювання датчиків верхнього рівня. Вивантаження зерна відбувається з вивантажувального отвору та триває до спрацювання нижніх датчиків рівня. До засобів контролю роботи конвеєра входять датчі швидкості та сходу стрічки, а також пристрій плавного пуску та зупинки

двигуна. Важливою задачею є також контроль потоку матеріалу, а саме своєчасне реагування у випадках коли відсутній потік через заштибовку, підпір, забивання або просту відсутність зерна, що подається.

Одне з найважливіших завдань при зберіганні зерна є підтримання температури зерна та вологості повітря в заданому діапазоні. Перш за все це означає безперервний контроль за процесом самозігрівання зерна і зміною температури зернової маси в силосах і ємностях для зберігання. Сертифіковані системи контролю температури зерна в силосах на базі термопідвісок з цифровими давачами температури забезпечують точний і безперервний контроль за станом зерна. Не менш важливим є завдання контролю вологості всередині зерносковища [12].

Своєчасне регулювання температури зерна досягається застосуванням активного вентилявання. Активне вентилявання заключається в примусовому продуванні атмосферного повітря через нерухомий насип зерна. Цей технологічний прийом заснований на використанні сквапності зернової маси: багаточисленні між зернові простори утворюють повітропровідну систему, роблячи її проникною для повітря, яке має змогу вільно переміщуватись. Потік повітря здійснює вплив на температуру і вологість зерна, змінює газовий склад повітря міжзернових просторів, тобто впливає на ті фактори, які в першу чергу відповідають за рівень життєдіяльності всіх живих компонентів зернової маси, а отже і її збереження. Зниження інтенсивності біологічних процесів псування зерна, тобто консервація – головний технологічний ефект активного вентилявання [13].

При активному вентиляванні тільки певна кількість повітря забезпечує достатньо швидке охолодження. Якщо повітря недостатньо, зерно охолоджується повільно, а найвіддаленіші від місця надходження повітря ділянки зернового насипу часто зволожуються. В цих зонах через довготривале знаходження зерна при підвищеній температурі і вологості активно розвиваються мікроорганізми, і можливе виникнення процесу самозігрівання. Отже, вентилявання необхідно проводити з такою інтенсивністю, щоб виключити розвиток небажаних процесів та охолодити

зерно набагато раніше, ніж воно може почати псуватись.

Активне вентилявання необхідно проводити в строгій відповідності до установлених для кожної культури режимів обробки. Під режимом активного вентилявання розуміють оптимальне поєднання основних параметрів обробки зерна повітряним потоком. До таких параметрів можна віднести: питому подачу повітря, тривалість та періодичність вентилявання, висоту зернового насипу [14].

Активне вентилявання зерна можна проводити не завжди. При вологості зерна нижче 20% в процесі вентилявання в окремих випадках можливе його незначне зволоження. Щоб цього не допустити, необхідно регулярно контролювати рівень відносної вологи повітря і за цим показником визначати величину рівноважної вологості зерна, до якої воно буде прямувати при вентиляванні. Таким чином, для визначення можливості вентилявання необхідно регулярно визначати відносну вологість повітря.

Процес сушіння зерна в нерухомому насипу активним вентиляванням представляє собою найпростіший спосіб конвективного сушіння, який широко використовується в сільському господарстві. Проте при сушці зерна підігрітим повітрям проявляється вагомий негативний фактор, пов'язаний з різким пересушуванням зерна в нижній і середній частинах насипу. Важливим технологічним завданням підвищення ефективності сушіння зерна в нерухомому насипі при використанні повітря з високою температурою закладається в тому, щоб не допустити різкого пересушування зерна. Іншим негативним фактором є те, що при вентиляванні зерна підігрітим повітрям утворюється небажана волога в повітрі. В цьому і полягає основна проблема кліматозабезпечення – одночасне підтримання всіх важливих параметрів зерна на необхідному рівні. Для досягнення результату необхідно реалізувати взаємопов'язане регулювання параметрів [15].

Для отримання рівномірного по вологості зерна першу половину часу сушіння проводять гарячим повітрям, а далі проводять обробку лише холодним повітрям. На другому етапі обробки холодне повітря починає зволожувати до оптимального рівня вологості раніше пересушену нижню частину насипу зерна, при цьому агент сушіння стає більш сухим, що

забезпечує досушування верхньої частини насипу [16].

Витяжні вентилятори, розміщені в даху силоса забезпечують відведення відпрацьованого повітря назовні. Найкращий варіант вентиляювання зерна при працюючих одночасно основному та дахових вентиляторах. Завдяки цьому вологе повітря буде легко видалятися поки зерно вентиляється. Для найкращого режиму роботи, витяжні вентилятори повинні працювати одночасно з основним та після його відключення залишатись працюючими певний час для того, щоб бути впевненим, що все вологе повітря видалено.

1.3. Постановка задачі роботи

Сучасний елеваторний комплекс або зерносховище має в своєму складі велику кількість обладнання і вимагає простого централізованого керування технологічними процесами. Це забезпечується застосуванням сучасних систем автоматизованого керування [13].

Проаналізувавши схему інформаційно-матеріальних потоків виділимо основні функціональні задачі керування, необхідні для забезпечення матеріальних потоків:

1. Завантаження та вивантаження зерна.
2. Підтримання відповідного рівня температури зерна.
3. Забезпечення прийнятної вологості повітря.

Із основних функціональних завдань можна виділити значну кількість підзадач:

1. Визначення нижнього і верхнього рівня зерна в силосі.
2. Керування роботою двигуна конвеєра.
3. Контроль положення стрічки конвеєра.
4. Контроль швидкості конвеєра.
5. Контроль потоку матеріалу, що надходить на конвеєр.
6. Керування засувом подачі зерна.
7. Контроль обертів двигуна основного вентилятора системи вентиляювання.
8. Регулювання витрати повітря основного вентилятора в залежності від заданих параметрів.

9. Керування роботою калорифера.
10. Контроль положення шиберних засувок витяжних вентиляторів.
11. Керування витяжними вентиляторам.

РОЗДІЛ 2

МЕТОДИ ТА ТЕХНОЛОГІЇ ВИРІШЕННЯ ЗАДАЧІ

2.1. Архітектура програмного забезпечення

Додаток створений як веб-застосунок – розподілений застосунок, в якому клієнтом виступає браузер, а сервером – веб-сервер. Так як клієнти можуть підключатись до системи з мобільних пристроїв та з своїх домашніх комп'ютерів або ноутбуків, потужність яких може бути незадовільною для виконання складних обчислень та обробки великої кількості даних, а в страхуванні по іншому не виходить, було прийняте рішення розробки «тонкого клієнта». Це означає що всі обчислення зосереджуються на веб-сервері.

В якості веб серверу було обрано «Amazon Web Services» з їх віртуальною машиною EC2.

Обчислювальний хмарний сервіс Amazon EC2 – це веб-сервіс, що надає безпечні масштабовані обчислювальні ресурси в хмарі. Він допомагає розробникам, полегшуючи проведення обчислень в хмарі в масштабі всього Інтернету. Простий веб-інтерфейс сервісу Amazon EC2 дозволяє отримати доступ до обчислювальних ресурсів і налаштувати їх з мінімальними зусиллями. Він надає користувачам повний контроль над обчислювальними ресурсами, а також перевірену обчислювальну середу Amazon для роботи.

Використовуючи AWS питання безпеки також делегуються до Amazon. Крім того, при реєстрації Amazon надає безкоштовне використання одного серверу протягом одного року.

На серверах EC2 встановлена операційна система Linux, що дуже схожа на образ CentOS7.

Варто розуміти що підключення до серверу відбувається завдяки протоколу SSH, тому всі зміни та налаштування відбуваються за допомогою терміналу та програмного забезпечення PuTTY. Інтерфейс програми PuTTY зображений на рисунку 2.1.

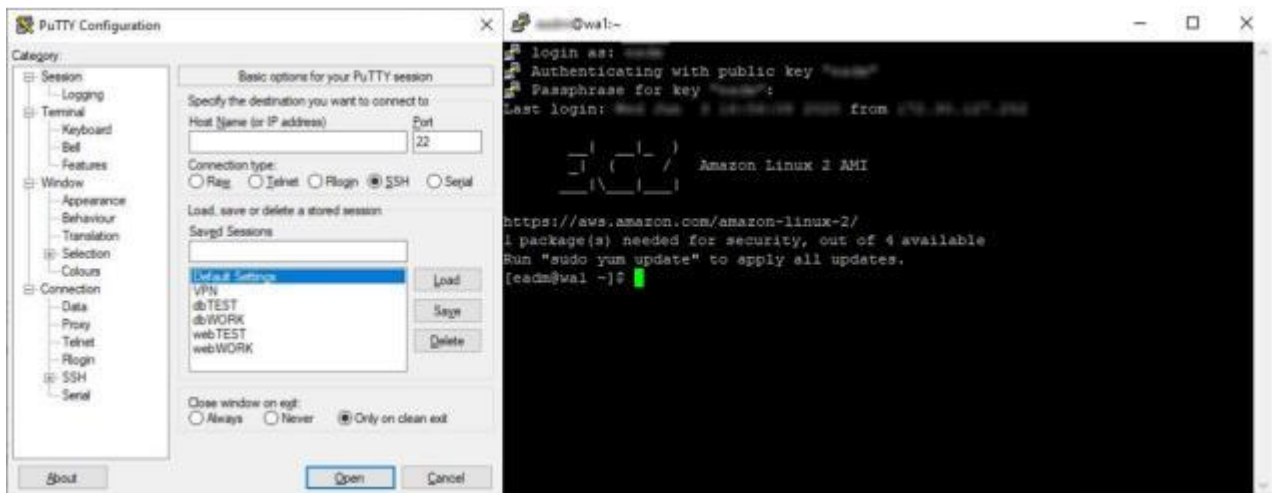


Рис. 2.1. Інтерфейс програми PuTTY

2.2. Вибір програмних додатків

Сьогодні більшість програмних продуктів є платними, однак існує достатньо програм з відкритим сирцевим кодом. Це дає можливість отримувати нові знання, поліпшуват вже створени продукт, співпрацювати з розробниками. Безпека – також важливий фактор, адже можна переглянути код та бути впевненим що всі дані користувача будуть в безпеці. Проте найголовніше в програмах з відкритим сирцевим кодом – це те, що не потрібно витратити кошти на ліцензії.

VPN мережа створена для підвищення рівня безпеки, адже всі клієнти розмежовані і не мають доступ до комп'ютерів іншої мережі. Однак це і можливість об'єднати їх в одній мережі. Загальна схема використання зображена на рисунку 3.2. Для створення VPN мережі використовується OpenVPN Server. OpenVPN – вільна реалізація технології віртуальної приватної мережі (VPN) з відкритим початковим кодом для створення шифрованих з'єднань між двома клієнтськими машинами або забезпечення роботи централізованого VPN-сервера для одночасної роботи декількох клієнтів. OpenVPN дозволяє встановлювати з'єднання між комп'ютерами, що перебувають за NAT-екраном, без необхідності зміни їхніх налаштувань. Для забезпечення безпеки керуючого каналу і потоку даних OpenVPN використовує бібліотеку OpenSSL. Завдяки цьому залучено весь набір алгоритмів шифрування, доступних у цій бібліотеці. Також може

використовуватися пакетна авторизація HMAS для забезпечення вищого рівня безпеки.

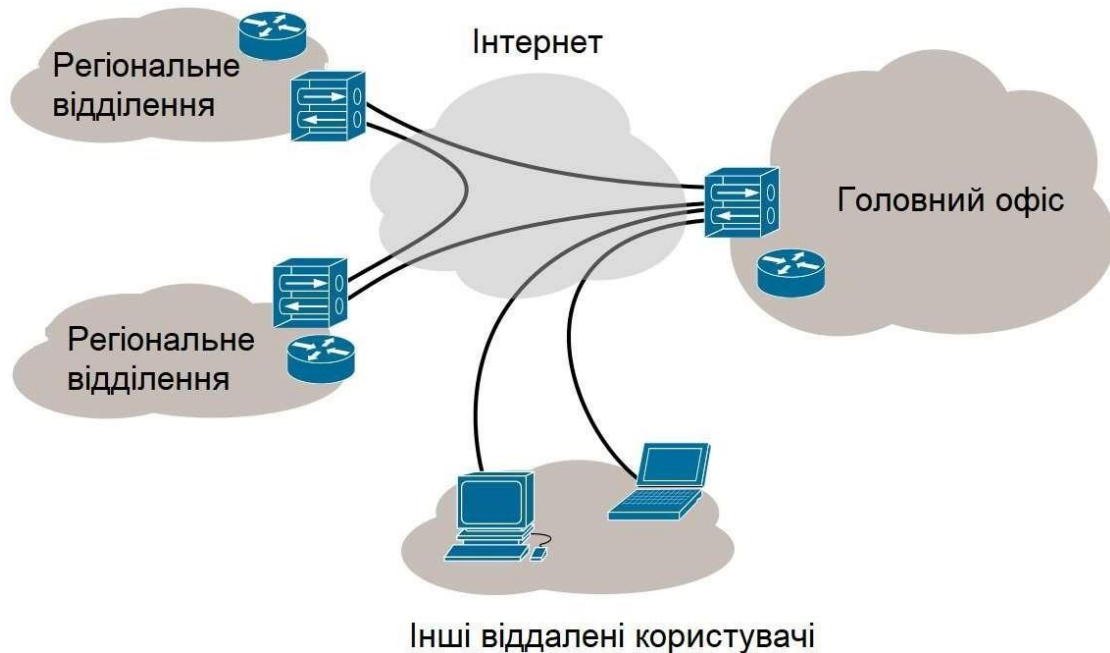


Рис. 2.2. Загальна схема використання VPN

В якості веб серверу був обраний Apache 2.4.41. Apache HTTP-сервер – відкритий веб-сервер Інтернет для UNIX-подібних, Microsoft Windows, Novell NetWare та інших операційних систем. Apache розроблюється та підтримується спільнотою розробників відкритого програмного забезпечення під керівництвом Apache Software Foundation. Станом на червень 2013 року Apache був встановлений на 53,34% серверів (358 974 045 одиниць). Для порівняння: на другому місці Microsoft IIS - їхня частка становить 17,22% серверів (115 920 681 одиниць).

Web-сервер Apache є самостійним, некомерційним, вільно розповсюджуваним продуктом. Продукт підтримує безліч можливостей, багато з яких реалізовані як скомпільовані модулі, які розширюють основні функціональні можливості. Існують інтерфейси для підтримки мов програмування Perl, Python, Tcl і PHP. Функції віртуального хостингу дозволяють одній інсталяції Apache обслуговувати різні веб-сайти. На цей веб сервер досить легко поставити SSL сертифікат безпеки для того, щоб

клієнт обмінювався даними з сервером по захищеному 443 порту.

В якості СУБД було обрано MariaDB 5.5.64. MariaDB – реляційна система керування базами даних, створена на початку 2016 року як відгалуження MySQL. MariaDB поширюється під вільною та відкритою ліцензією GNU GPL. Поштовхом для створення MariaDB стала непевність спільноти розробників та користувачів невизначеною долею і ліцензією MySQL після її придбання Oracle. Проект підтримує високий ступінь сумісності з MySQL, включаючи "вставку" замінних можливостей бінарним еквівалентом бібліотек і відповідністю з наявним API і командами MySQL. Навіть перша стабільна версія СУБД отримала номер 5.2, ніби продовжуючи лінійку MySQL.

Для полегшення роботи було встановлено програмне забезпечення phpMyAdmin. Воно дає можливість швидко та легко змінювати дані і БД не використовуючи термінал та SQL команди. Інтерфейс програми зображений на рисунку 2.3.

Для генерації PDF-документів використовується безкоштовна бібліотека з відкритим кодом TCPDF. TCPDF – це єдина бібліотека на основі PHP, яка включає повну підтримку Unicode UTF-8. Зазначена бібліотека є однією з найбільш використовуваних бібліотек PHP у світі, завдяки тому, що вона вже включена до найпопулярніших CMS додатків.

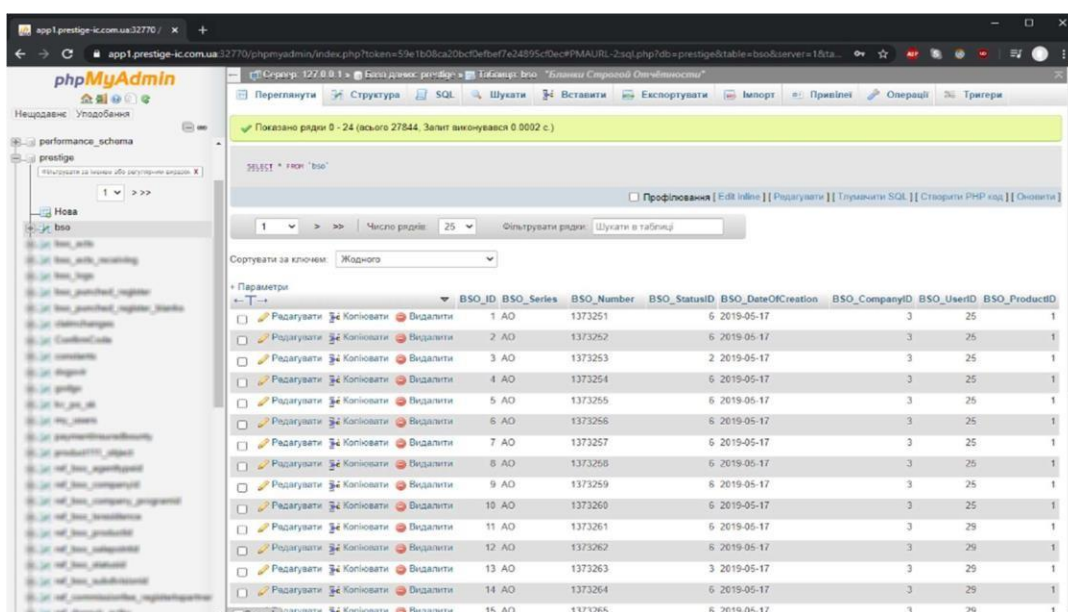


Рис. 2.3. Інтерфейс програми phpMyAdmin

2.3. Вибір мов програмування

В якості мови програмування було обрано JavaScript з бібліотекою jQuery та PHP.

Таке рішення обумовлене тим, що JavaScript є єдиною динамічною, об'єктно-орієнтованою мовою програмування, скрипти якої виконуються на стороні клієнта (пристрої кінцевого користувача). Вона найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки. JavaScript класифікують як прототипну, скриптову мову програмування з динамічною типізацією. Особливістю JavaScript є те, що вона має C-подібний синтаксис.

jQuery – популярна JavaScript-бібліотека з відкритим кодом. Вона була представлена у січні 2016 року. Згідно з дослідженнями організації W3Techs, JQuery використовується понад половиною найвідвідуваніших сайтів. JQuery є найпопулярнішою бібліотекою JavaScript, яка посилено використовується на сьогоднішній день. JQuery є вільним програмним забезпеченням під ліцензією MIT. Синтаксис jQuery розроблений, щоб зробити орієнтування у навігації зручнішим завдяки вибору елементів DOM, створенню анімації, обробки подій, і розробки AJAX-застосунків. Основне завдання jQuery – це надавати розробнику легкий та гнучкий інструментарій кросбраузерної адресації DOM об'єктів за допомогою CSS та XPath селекторів.

Для написання також використовувалась мова розмітки HTML. Це мова тегів, якою пишуться гіпертекстові документи для мережі Інтернет. Веб-браузери отримують HTML-документи з веб-сервера або з локальної пам'яті і передають документи в мультимедійні веб-сторінки.

В свою чергу на стороні серверу використана скритована мова PHP. Вона є однією з найпоширеніших мов, що використовуються у сфері веб-розробок. PHP підтримується переважною більшістю хостинг-провайдерів. PHP – проект відкритого програмного забезпечення. PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. На відміну від

скриптово мови JavaScript користувач не бачить PHP-коду, тому що браузер отримує готовий HTML-код. Це є перевагою з точки зору безпеки, але погіршує інтерактивність сторінок. Але надана можливість використовувати мову PHP для генерування JavaScript-кодів, які виконуються вже на стороні клієнта.

Для оформлення сторінок використовувалась мова CSS – спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду. CSS є основною технологією всесвітньої павутини, поряд із HTML та JavaScript. Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Всі дані передаються або формами HTML, або JSON чи XML файлами.

За допомогою форм HTML можна передавати дані POST та GET запитами POST – один з багатьох методів запити, що підтримуються протоколом передачі даних HTTP, який використовується у всесвітній мережі Інтернет. Метод POST призначений для запити, при якому веб-сервер приймає дані збережені в тілі повідомлення, для зберігання. Метод часто використовується для завантаження файлу або передачі заповненої веб-форми. На відміну від POST, метод GET призначений для отримання інформації від сервера. В рамках GET-запитів деяких даних можуть бути передані в рядку запити URI, який вказує, наприклад, умови пошуку, діапазони дати, або іншу інформацію, що визначає запит. В рамках методу POST-запити довільну кількість даних будь-якого типу може бути відправлено на сервер в тексті листа запити.

JSON – це текстовий формат обміну даними між комп'ютерами. JSON базується на тексті, може бути прочитаним людиною. Формат дає змогу описувати об'єкти та інші структури даних. Цей формат використовується переважно для передачі структурованої інформації через мережу.

XML – запропонований консорціумом World Wide Web Consortium стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками, зокрема, через Інтернет. XML-документ

складається із текстових знаків, і придатний до читання людиною.

JSON знайшов своє головне призначення в написанні веб-програм, а саме при використанні технології AJAX. JSON, що використовується в AJAX, виступає як заміна XML під час асинхронної передачі структурованої інформації між клієнтом та сервером. При цьому перевагою JSON перед XML є те, що він дозволяє використовувати складні структури в атрибутах, займає менше місця і прямо інтерпретується за допомогою JavaScript в об'єкти.

AJAX – підхід до побудови користувацьких інтерфейсів веб-застосунків, за яких вебсторінка, не перезавантажуючись, у фоновому режимі надсилає запити на сервер і сама звідти довантажує потрібні користувачу дані. AJAX – один з компонентів концепції DHTML. Процес роботи AJAX зображений на рисунку 2.4.

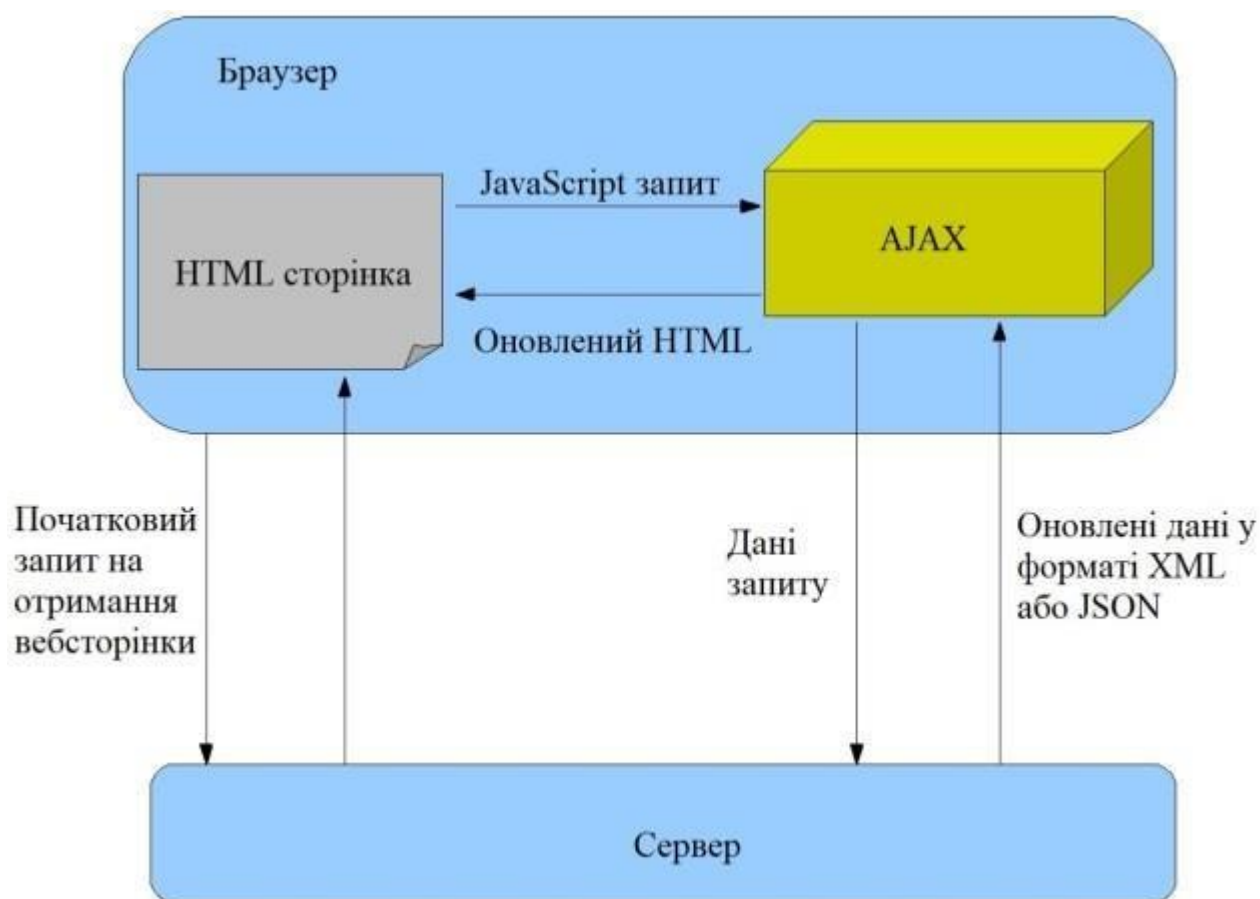


Рис. 2.4. Схема роботи AJAX

Порівняємо дві моделі: класичну та з використанням AJAX

Класична модель веб-застосунку:

1. Користувач заходить на вебсторінку і натискає на який-небудь її

елемент.

2. Браузер надсилає запит серверу.

У відповідь сервер генерує повністю нову вебсторінку і відправляє її браузеру і т. д.

Модель AJAX:

- 1) Користувач заходить на вебсторінку і натискає на який-небудь її елемент.
- 2) Браузер відправляє відповідний запит на сервер.
- 3) Сервер віддає тільки ту частину документа, яка змінилася.

2.4. Опис архітектури мережі

Загальна архітектура мережі зображена на рисунку 2.5. IP адреси не відображаються через вимоги безпеки.

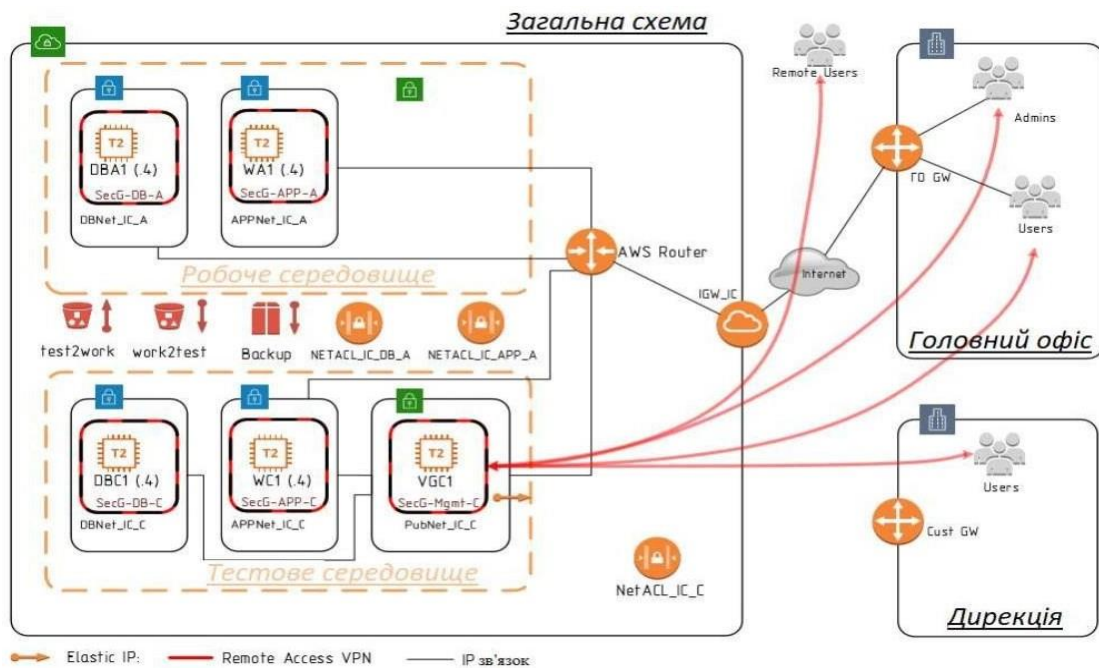


Рис. 2.5. Загальна схема інфраструктури

На схемі присутні два середовища – тестове та робоче. Це зроблено для того, щоб клієнти працювали на стабільній версії програми. Обмін файлами між середовищами відбувається за допомогою Amazon S3 – сервіс-сховище даних, один з сервісів AWS. Сервіс надає можливість для зберігання й отримання будь-якого обсягу даних, у будь-який час з будь-якої точки мережі, тобто так званий файловий хостинг. За допомогою Amazon S3 досягається висока масштабованість, надійність, висока швидкість недорог

інфраструктур зберігання даних. При виконанні скрипту «work2 test» копіюється база даних та програмний код, а при виконанні «test2work» – лише програмний код, для унеможливлення втрати даних, що були внесені в БД.

В кожному середовищі розміщені дві віртуальні машини Amazon EC2. Це web-сервер (WA1, WC1) та сервер бази даних (DBA1, DBC1). І тестовому середовищі також розміщений VPN сервер. На віртуальній машині WA1 налаштований планувальник задає створення резервних копій програми та бази даних кожні дві години. Ці копії зберігаються протягом двох днів, після чого видаляються. Це дає можливість швидко відновити базу даних у випадку перебоїв на сервері. На кожному web-сервері встановлено сертифікати безпеки.

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ЙОГО ЗАСТОСУВАННЯ

3.1 Вибір ПЛК та інтерфейсів

Як і будь-яка сучасна АСУ ТП, наша система має трирівневу структуру. Нижній рівень – рівень обладнання (входів / виходів – Input / Output-level). Це рівень датчиків, пристроїв, які контролюють керовані параметри, а також пристроїв, які впливають на ці параметри процесу, для приведення їх у відповідність із завданням. На цьому рівні здійснюється узгодження сигналів датчиків з входами пристрою керування, а команд, що виробляються з виконавчими пристроями.

Середній рівень – рівень керування обладнанням (Control level). Це рівень контролерів. ПЛК постійно отримує дані з контрольно-вимірювального устаткування та датчиків про існуючий стан технологічного процесу і видає команди керування, відповідно до запрограмованого алгоритмом керування, на виконавчі механізми.

Верхній рівень – рівень промислового сервера, мережевого устаткування, рівень операторських і диспетчерських станцій. Саме на даному рівні відбувається контроль процесу виробництва, а саме забезпечується зв'язок з нижніми рівнями, де здійснюється збір інформації, візуалізація та диспетчеризація (моніторинг) ходу процесу. Це рівень HMI, SCADA. На цьому рівні задіяна людина, тобто оператор (диспетчер). Вона здійснює локальний контроль технологічного обладнання через так званий людино-машинний інтерфейс (HMI – Human Machine Interface). До нього відносяться: монітори, графічні панелі, які встановлюються локально на пультах керування і шафах автоматики. Для здійснення контролю за розподіленою системою машин, механізмів і агрегатів застосовується SCADA (Supervisory Control And Data Acquisition – диспетчерське керування і збір даних) система. Ця система являє собою програмне забезпечення, яке налаштовується і встановлюється на диспетчерських комп'ютерах. Вона забезпечує збір, архівацію, візуалізацію, найважливіших даних від ПЛК.

При отриманні даних система самостійно порівнює їх із заданими значеннями керованих параметрів і при відхиленні від завдання повідомляє оператора за допомогою тривоги, дозволяючи йому вжити необхідних заходів. При цьому система записує все, що відбувається, включаючи дії оператора, забезпечуючи контроль дій оператора в разі аварії або іншої нештатної ситуації. Таким чином, забезпечується персональна відповідальність керівника оператора [23].

Апаратною платформою АСУ ТП обрано програмований контролер S7-300 виробництва Siemens. Розроблений на базі модульної, що розширюється, архітектури для задач керування в теперішньому часі. Системи, створені на базі даного ПЛК можуть змінюватися від найпростіших одиночних систем керування (до 448 точок вводу / виводу) до складних розподілених мереж (до 64 000 точок вводу / виводу).

Контролер Siemens в першу чергу має модульну структуру. Контролер у своєму складі повинен містити процесорний модуль та блок живлення. Окрім вищесказаного існує можливість установки близько 10 інтерфейсних модулів, які обираються відповідно до правил об'єкта керування. Будь-який інтерфейсний модуль повинен виступати в якості закінченого пристрою. Крім узгодження інтерфейсів зв'язку і трансформації протоколів дає можливість резервувати канали зв'язку в різноманітних комбінаціях (але слід враховувати певні обмеження: резервування каналу вимагає наявності схожих протоколів зв'язку).

Будь-який апаратний інтерфейсний модуль контролера можна вважати інтелектуальним приладом (тобто він містить у своєму складі процесор) і дає можливість перетворювати інформацію підтримуваного протоколу в вид, очевидний для відповідного драйверу модуля. Драйвери ж поміщають отримані дані в загальну для всього контролера спеціальну базу даних, яка називається «база сигналів». Подання інформації в цій базі однаково. Виконуюча система розподіляє інформацію по різним драйверам і програмам відповідно до вищевказаних правил перетворення. Інформація, що прибуває на вхід драйверів, направляється до наступних апаратних модулів, звідки і передається назовні відповідно до вказаного протоколу.

Програмовані контролери Siemens SIMATIC S7-300 відзначаються модульною конструкцією і містять у собі наступні елементи:

- Модуль центрального процесора (CPU). В залежності від складності запиту в контролерах можуть використовуватися різні типи центральних процесорів, які різняться між собою, в першу чергу, продуктивністю, об'ємом пам'яті, наявністю або відсутністю вбудованих входів-виходів і типових функцій, кількістю і видом вбудованих комунікаційних інтерфейсів та ін.

- Блоки живлення (PS) – відповідають за живлення контролера від мережі змінного струму напругою від 120В до 230 В або від джерела постійного струму різними показниками напруги (24/48/60/110 В).

- Сигнальні модулі (SM) – необхідні для введення та виведення дискретних або аналогових сигналів з різними електричними та часовими характеристиками.

- Комунікаційні модулі (CP) дають можливість підключитися до мереж PROFIBUS, Industrial Ethernet, AS-Interface або встановити зв'язок по PtP (point to point) інтерфейсу.

- Функціональні модулі (FM) необхідні для самостійного вирішення запитів автоматичного регулювання, позиціонування та обробки сигналів. Функціональні модулі у своєму складі містять вбудований мікропроцесор і реалізують покладені на них функції навіть якщо центральний процесор програмованого логічного контролера відмовився від виконання даної функції [23].

Нами було обрано ПЛК з процесором 317-2 PN / DP, характеристики якого наведені в табл. 3.1.

Таблиця 3.1. Технічні характеристики модуля процесора 317-2 PN / DP

Робоча пам'ять		1 Мб
Завантажена пам'ять MMC		64Кб-8Мб
Час виконання операцій, мкс.	Логічних	0,05
	З фіксованою точкою	0,2
	З плаваючою точкою	1
Кількість таймерів/лічильників		512/512
Кількість каналів введення/виведення, аналогових/дискретних		65536/4096
Вбудовані інтерфейси		MPI/DP+PROFINET
Кількість активних комунікаційних з'єднань		32
Габаритні розміри, мм		160,125,130

Також Siemens контролери підтримують додаткові інтерфейсні модулі, призначені для додавання наступних інтерфейсів: Modbus, Modbus Plus, TCP / IP Ethernet та інших. Встановлення мережі Modbus Plus дає можливість встановлювати контролери Siemens в якості рівноправних вузлів в індустріальній мережі на основі RS - 485. Програмовані контролери Siemens SIMATIC S7-300 дозволяють застосовувати в своєму складі не більше 32 сигнальних і функціональних модулів, а також комунікаційних процесорів.

В конфігурації контролера задаються норми маршрутизації даних з модулів-джерел в модулі-приймачі. Задаються, в першу чергу, наступні правила: протокольні адреси сигналів та клас даних. Окрім вищевказаного, в конфігурації визначаються характеристики ініціалізації програмних і апаратних модулів, таких, як швидкість функціонування в каналах для дротових цифрових мереж та таблиці опитування.

Комунікаційні Можливості

- 1) прийом та передача даних по інтерфейсах RS-232, RS-485/422, Ethernet, V.23, стик С1-ТЧ;
- 2) підтримка протоколів: Ethernet TCP / IP, Modbus (RTU, TCP), ГОСТ Р МЕК 870-5101, ГОСТ Р МЕК 608070-5-104;
- 3) наявність веб-інтерфейсу;
- 4) підтримка бездротової передачі інформації на базі стандарту GSM,

Глобалстар;

- 5) сполучність з існуючими системами на основі ПЛК, мережами та засобами операторського інтерфейсу.

Функції, що можуть бути реалізовані:

- 1) трансформація певного протоколу в інший або декілька інших;
- 2) перерозподіл даних кількох потоків;
- 3) резервування каналів зв'язку;
- 4) надійна робота навіть на низькошвидкісних каналах зв'язку;
- 5) естафетний характер передачі даних;
- 6) робота з SCADA-системою на базі OPC-інтерфейсу;
- 7) співставлення інтерфейсів зв'язку;
- 8) незалежна робота в режимі без обслуговування.

Апаратні особливості:

- 1) модульна система з можливістю змін інтерфейсів, що підключаються;
- 2) резервування джерела живлення та процесорного модуля;
- 3) так звана «гаряча» заміна модулів;
- 4) значна міцність за рахунок реалізації в металевому корпусі.

Програмні особливості:

- a) середовище програмування Simatic Step 7;
- b) відповідає стандарту ІЕС 61131-3;
- c) підтримка декількох (5) мов програмування;
- d) призначена для користувача розробка і налагодження прикладних програм;
- e) моніторинг і редагування в режимі реального часу.

Переваги:

- ✓ достатньо широкий вибір інтерфейсних модулів;
- ✓ підтримка стандартних промислових протоколів і інтерфейсів;
- ✓ відкритість архітектури програмного забезпечення;
- ✓ висока швидкість обробки та передачі даних;
- ✓ гнучка апаратна платформа.

Модулі введення дискретних сигналів SM 321 забезпечують перетворення вхідних дискретних сигналів контролера в внутрішні логічні сигнали. До входів модулів мають можливість підключатися контактні або ж безконтактні давачі BERO.

Модулі SM 321 можуть працювати в системах локального введення-виведення всіх модифікацій програмованих контролерів S7-300, а також в станціях розподіленого вводу-виводу ET 200M.

Виберемо модуль SM 321-1BH02, який містить 32 входи.

Модулі виводу дискретних сигналів SM 322 призначені для перетворення внутрішніх логічних сигналів контролера в його вихідні дискретні сигнали. До виходів модулів можуть підключатися виконавчі пристрої або їх комутаційні апарати.

Виберемо модуль SM 322-1BH01, який містить 16 виходів.

3.2 Програма керування процесом зберігання зерна

Для розробки програмного забезпечення (ПО) сучасних програмованих логічних контролерів (ПЛК), що мають вбудовану операційну систему, можна використовувати як традиційні інструментальні засоби (компілятори мов СІ, Паскаль, і т.д.), так і спеціалізовані мовні засоби.

Програмування на мові низького рівня (Асемблер) та мовами високого рівня вимагає програміста високої кваліфікації, додаткових знань особливостей операційної системи і апаратних засобів. Переносимість програм на іншу апаратного-програмну платформу погана [24].

Програми, написані на спеціалізованих мовах, мають повну переносимість на інші процесори (за наявності системи виконання спеціалізованої мови), більш наближені до особливостей систем автоматизації і не вимагають додаткових знань від розробника. STEP 7 – це базовий пакет програм, що включає в свій склад весь спектр інструментальних засобів, необхідних для конфігурації апаратури і промислових мереж, налаштування параметрів, програмування, діагностики та обслуговування систем керування, побудованих на основі програмованих контролерів SIMATIC S7-300 / S7-400 / C7 / WinAC.

Відмінною особливістю пакета STEP 7 є можливість розробки комплексних проектів автоматизації, що базуються на використанні безлічі програмованих контролерів, промислових комп'ютерів, пристроїв і систем людино-машинного інтерфейсу, пристроїв розподіленого введення-виведення, мережових структур промислового зв'язку. Обмеження на розробку таких проектів накладаються тільки функціональними можливостями програматорів або комп'ютерів. При необхідності STEP 7 може доповнюватися інструментальними засобами проектування, що значно спрощують розробку складних проектів.

У STEP 7 закладена методологія структурного програмування, що дозволяє користувачеві представити автоматизований процес в найбільш легкій і зрозумілій формі. Стандартом MEK 61131-3 визначається п'ять мов: три графічних (SFC, FBD, LD) і два текстових (ST, IL). Всі ці мови програмування інтегровані в єдину інструментальну середу і працюють з єдиними об'єктами даних.

SFC – графічна мова послідовних функціональних схем. Керуюча програма представляється послідовністю кроків, поділюваних переходами. Мова добре пристосована для програмування задач логічного керування.

LD – графічна мова релейної логіки (Ladder Diagram). Мова LD застосовується для опису логічних виразів і для вирішення завдань у вигляді релейно-контактних схем автоматики.

ST – мова структурованого тексту (Structured Text). Це мова високого рівня, по синтаксису схожа на Паскаль і застосовується для програмування складних логічних і обчислювальних процедур, які важко описати графічними мовами.

FBD – графічна мова функціональних блокових діаграм, за допомогою якої програма представляється у вигляді різних функціональних блоків (арифметичних, тригонометричних, регуляторів, мультиплексорів і т.д.) Мова зручна для програмування задач, наприклад, обчислювального характеру, рішення яких може бути представлено функціональною схемою.

IL – мова інструкцій (Список команд) – мова низького рівня, схожа на

Асемблер і є високоефективною для невеликих програм або для оптимізації окремих частин складних програм при наявності вимог високої швидкодії, економії пам'яті [25].

SCADA – процес збору інформації реального часу з віддалених точок для обробки, аналізу і можливого керування віддаленими об'єктами. Вимоги обробки реального часу обумовлюються необхідністю доставки всіх необхідних подій і даних на центральний інтерфейс оператора (диспетчера). Прообразом сучасної системи SCADA на ранніх стадіях розвитку АСУ є системи телеметрії і сигналізації. Всі сучасні SCADA системи включають 3 основних структурних компонента:

- RTU (Remote Terminal Unit) – віддалений термінальний пристрій (нижній рівень АСУТП: промислові комп'ютери, ПЛК);
- MTU (Master Terminal Unit) – диспетчерський пункт управління (верхній рівень);
- CS (Communication System) – комунікаційна система [26].

Основні завдання, які вирішуються SCADA-системою:

1. Забезпечення обміну даними в реальному часі з об'єктом – промисловими контролерами, термінальними пристроями;
2. Обробка даних в реальному часі (масштабування змінних);
3. Візуалізація на моніторах і терміналах ходу технологічного процесу в зручній для людини формі;
4. Забезпечення зберігання технологічної інформації в базі даних реального часу;
5. Ведення контролю технологічних параметрів;
6. Реалізація аварійної сигналізації і протоколу тривожних подій;
7. Генерація звітів про хід перебігу технологічних процесів;
8. Надання даних зовнішнім системам рівня управління підприємством.

Основні вимоги до SCADA систем:

1. Надійність системи (технологічність і функціональність);
2. Безпека керування;
3. Точність обробки і представлення даних;
4. Простота розширення системи;

Вимоги безпеки і надійності керування в SCADA включають:

1. Ніяка одинична відмова обладнання не повинна викликати помилкового вихідного впливу на об'єкт керування;
2. Ніяка одинична помилка оператора не повинна викликати видачу помилкового вихідного впливу на об'єкт керування;
3. Всі операції по керуванню повинні бути інтуїтивно зрозумілими і зручними для оператора.

Вибір SCADA системи можна проводити відповідно до характеристик, зазначеним в табл. 3.2.

Аналізуючи дані наведені в табл. 3.3 можна виділити систему Trace Mode V6, як найдешевшу з розглянутих. Однак у функції виконавчого модуля не входить архівування даних, що визначає вибір Master SCADA V3.1.

Мнемосхема системи автоматизованого керування технологічним процесом зберігання зерна приведено на рис. 3.1.

Екран термометрії SCADA системи автоматизації технологічного процесу зберігання зерна в силосі приведено на рис. 3.2.

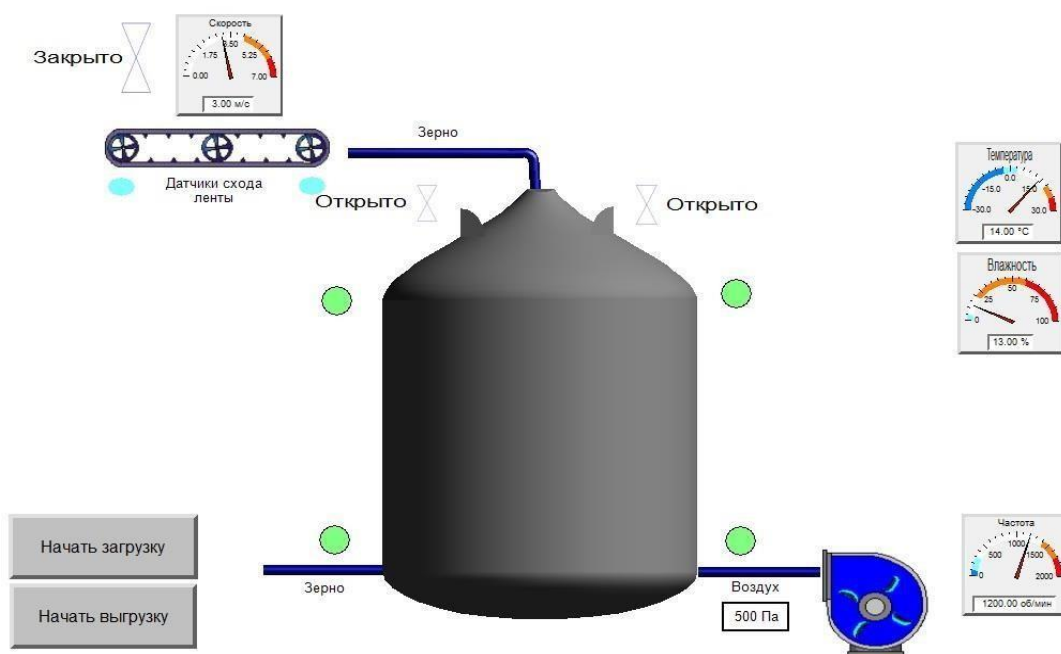


Рис. 3.1. Мнемосхема системи автоматизованого керування технологічним процесом зберігання зерна

на якість зберігання зерна, та в залежності від зміни цих параметрів вмикати чи вимикати виконуючі механізми, змінювати їх режими роботи.

Отже, це не є ідеальним варіантом. Значно більшої ефективності можна досягти шляхом автоматизації процесу зберігання зерна. Цей варіант найбільш коштовний, оскільки потребує додаткових витрат на купівлю ПЛК, давачів, налагодження інтерфейсів зв'язку та створення SCADA системи. Проте в цьому випадку керування процесом зберігання здійснюється за заданим законом. Перше, що можна з цього отримати – це звільнення людини від завдань збору показань технологічних параметрів та зміни керуючого впливу, а отже і звільнення від можливості людської помилки.

ВИСНОВКИ

Завдання автоматизації комплексу по зберіганню зерна на сьогоднішній день є досить актуальним. Сучасні засоби АСУ ТП зернопереробних підприємств забезпечують зниження втрат при зберіганні і переробці зерна, заощадження енергоресурсів зернопереробних підприємств, елеваторів, мінімізацію впливу людського фактора, ризиків виникнення аварійних ситуацій роботи автоматизованих технологічних комплексів по зберіганню і переробці зерна. Останні розробки в області АСУ ТП зернопереробної галузі дають можливість автоматично прогнозувати процес самозігрівання зерна, що є дуже необхідним в забезпеченні підвищення ефективності зберігання зерна.

В результаті виконання роботи були досягнуті наступні результати:

1. Проведено конструктивно-технологічний аналіз процесу зберігання зерна в силосі;
2. Розроблено схему інформаційно-матеріальних потоків;
3. Проаналізовано функціональні задачі керування;
4. Розроблено структурну схему та функціональну схему автоматизації системи автоматизованого керування процесом зберігання зерна в силосі;
5. Реалізовано вибір давачів та виконуючих механізмів;
6. Досліджено параметри регулятора кліматозабезпечення зерносховища;
7. Реалізовано SCADA-систему.

В результаті дослідження параметрів регулятора кліматозабезпечення зерносховища було досягнуто покращення показників зберігання зерна та підвищення енергоефективності процесу.

Отримані результати можуть бути використані для створення аналогічних систем, або модернізації існуючих.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Активне вентилування та сушіння зерна [Текст] : навч. посіб. / О. І. Гапонюк, М. В. Остапчук, Г. М. Станкевич, І. І. Гапонюк. – Одеса : ВМВ, 2014. – 326 с.
2. Dorigo M. Optimization, learning and natural algorithms: Ph.D. Thesis/ Politecnio de Milano, Italy. Milan, 1992. 25 p.
3. Dorigo M., Maniezzo V., Colorni A. The ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics*. 1996. Part B. № 1(26). P. 29–41.
4. Dorigo M., Stutzle T. Ant Colony Optimization. Cambridge: MIT Press, 2004. 35 p.
5. Dorigo M., Di Caro G., Gambardella L.M. Ant algorithms for distributed discrete optimization. *Artificial Life*. 1999. № 5. P. 137–172.
6. Dorigo M., Gambardella L.M. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*. 1997. № 1. P. 53–66.
7. Stutzle T., Hoos H.H. Max-min ant system. *Future Generation Computer Systems*. 2000. № 16. P. 889–914.
8. Gambardella L.M., Taillard E.D., Agazzi G. MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows, in: D. Corne, M. Dorigo, F. Glover (Eds.), *New Ideas in Optimization*, London: McGraw-Hill. 1999. P. 63–76.
9. Richards J.A. Remote Sensing Digital Image Analysis. An Introduction. Berlin: Springer, 2013. 503 p.
10. Qihao Weng. Remote Sensing and GIS Integration. New York: McGraw- Hill Professional, 2009. 416 p.
11. Технологічне обладнання зернопереробних та олійних виробництв / О.В. Дацишин, А.І. Ткачук, О.В. Гвоздєв; ред.. О.В. Дацишина. Навчальний посібник. – Вінниця: Нова Книга, 2013. – 488с.
12. Сільськогосподарські будівлі та споруди [Електронний ресурс] – Режим доступу: <http://bibliograph.com.ua/spravochnik-44/16.htm>

13. Пузік Л. М. Технологія зберігання і переробки зерна. / Л. М. Пузік, В. К. Пузік. – Х.: Точка, 2013. – 311 с.
14. Занько М. Правильний контроль зерна під час зберігання / М. Занько // Пропозиція. – 2015. – С. 104 – 107.
15. Gewinner J, Harnisch R, Muck O. Manual on the Preservation of Post-Harvest Grain Losses. – GTZ, Eschborn, 1996. – 294 p.
16. Maier DE. Preservation of grain with aeration. In: Grain Drying in Asia. Champ BR, Highley E, Johnson GI eds. ACIAR Publication 71, 1996. – 397p.
17. Подпратов Г. І. , Рожко В.І., Скалецька Л.Ф. Технологія зберігання та переробки продукції рослинництва: підручник. – К.: Аграрна освіта, 2014. – 393 с.
18. McFarlane JA. Storage methods in relation to post-harvest losses in cereals. – Insect Sci Appl, 1988. – 854 p.

Додаток А

Лістинг програми

Текст підпрограми для авторизації в системі

```
<?php
    session_start(); // Початок
    сессії $username =
    $_POST["form_username"]
    ;
    $_SESSION['username'] =
    $_POST["form_username"];
    $_SESSION['userpassword'] =
    $_POST["form_password"];
    $_SESSION['dbip'] = "127.0.0.1";
    $_SESSION['dbname'] = "prestige";
    if($link = mysqli_connect($_SESSION['dbip'], $_SESSION['username'],
$_SESSION['userpassword'], $_SESSION['dbname'])){ // спроба підключення до БД
        $r = mysqli_fetch_array(mysqli_query($link, "SELECT `id`, `ierarhic`,
`subdivisionid` from `my_users` where `username` = '". $username. "'")); // збереження
необхідних даних
        $_SESSION['userid'] =
        $r[0];
        $_SESSION['userierarhic'
        ] = $r[1];
        $_SESSION['usersd'] =
        $r[2];
        mysqli_close($link);
        $_SESSION['connected']
        = true;
    }
    {
        $_SESSION = array();
        session_destroy();
        session_start();
        $_SESSION['connecte
        d'] = false;
        $_SESSION['result'] =
        "error";
    }
    header ('Location: index.php'); // перенаправлення на
    потрібну сторінку exit();
?>
```

Текст підпрограми для відображення сторінки створення бланку

```
<?php
    error_reporting(0);
    session_start();
    if($_SESSION['connected']
    == false){
        header ('Location: /prestige/Login.php'); // перенаправлення на
        потрібну сторінку exit();
    }
    e
    l
    s
    e
    {
```

```

        $userid = $_SESSION['userid'];
        $link = mysqli_connect($_SESSION['dbip'],
$_SESSION['username'],
$_SESSION['userpassword'],$_SESSION['dbname']);
        mysqli_set_charset($link, "utf8");
        $SD = mysqli_fetch_array(mysqli_query($link, "SELECT `subdivisionid`
from `my_users` where `id` = ".$userid))[0];
        if(mysqli_fetch_array(mysqli_query($link, "SELECT `AllowAddBlanks`
from `roles` where `UID` = ".$userid.""))[0] == 0){
            header ('Location: /prestige/index.php'); // перенаправление
на нужную страницу
        }
        exit(); }
    }
?>
<html>
<!DOC
TYPE
html>
<html
lang="ru"
u">
<head>
<link rel="shortcut icon" href="/prestige/Icon.ico"
type="image/x-icon"> <link href="/prestige/css/main.css"
rel="stylesheet" />
<link href="/prestige/css/css.css" rel="stylesheet" />
<link href="/prestige/css/defaultpage.css"
rel="stylesheet" /> <meta charset="utf-8">
<title>Створення
бланку</title>
<script>
    function tryToCreate(){
        var AllFine = true;
        if(document.getElementById('form_company').value == 0 &&
AllFine == true){ AllFine = false;
            alert("Перевірте введені
данні(компанія)"); }
        if(document.getElementById('form_product').value == 0 &&
AllFine == true){ AllFine = false;
            alert("Перевірте введені данні(продукт)");
        }
    }
</td> </tr>
<tr class = "trinput"> <td>
<select id="form_company" name="form_company" form="my_form"> <option value
= 0 disabled selected>Виберіть компанію</option> <?php
$result = mysqli_query($link, 'SELECT CompanyID, CompanyName FROM
ref_bso_companyid'); while($row = mysqli_fetch_array($result)){
echo '<option value = "'.$row[0]."'>'.$row[1].</option>'; }
>
<select id="form_product" name="form_product" form="my_form"> <option value = 0
disabled selected>Виберіть продукт</option> <?php
$result = mysqli_query($link, 'SELECT ID, productid, simpleDescription FROM
ref_bso_productid where blank = 1 order by `productid`');

```



```

while($row = mysqli_fetch_array($result)){
echo '<option value = "'. $row[0]."'>'. $row[2]. '</option>'; }
?>
</td> </tr>
<tr class = "trtitle"> <td>
Дата створення бланку </td>
<td>
Користувач </td>
</tr>
<tr class = "trinput"> <td>
<input type="date" value="<?php echo date("Y-m-d");?>" id="form_date"
name="form_date" form="my_form">
<select id = "form_user" name = "form_user" form="my_form"> <?php
$result = mysqli_query($link, 'SELECT id, names FROM my_users where `id` = 1');
while($row = mysqli_fetch_array($result)){
echo '<option value = "'. $row[0]."'>'. $row[1]. '</option>'; }
<tr class = "trtitle"> <td>
Серія бланку </td>
</tr>
<tr class = "trinput"> <td>
<input type="text" id="form_seried" name="form_seried" pattern="^[0-9A-Яа-
яЁё]+$" <?PHP if($_SESSION['seried']){ echo (' value = "'.$_SESSION['seried'].'" ');
$_SESSION['seried'] = null;}?> form="my_form"> </td>
</tr>
<tr class = "trtitle"> <td>
Мінімальне значення номера </td>
<td>
Максимальне значення номера </td>
</tr>
<tr class = "trinput">
<td>
<input type="text" id="form_maxnumber" name="form_maxnumber" pattern="[0-
9]+$" <?PHP if($_SESSION['num2']){ echo (' value = "'.$_SESSION['num2'].'" ');
$_SESSION['num2'] = null;}?> required form="my_form">
</td> </tr>
<tr class = "trtitle"> <td>
Коментар </td>
</tr>
<tr class = "trinput">
<td colspan = '2'>
<input type = "text" <?php if($_SESSION['com']){echo'value =
"'.$_SESSION['com'].'"; $_SESSION['com'] = NULL;}?> id="form_comment"
name="form_comment" form="my_form">
</td> </tr>
<tr class =
"trin
put"
>
<td
>
<input type="submit" id = "confirm" form="my_form" hidden>
<input type="button" class = "submitbutton" value="СТВОРИТИ" onclick =
"tryToCreate()"> </td>>
displ('Blanks');

```

```

document.getElementById('CreateBlankMenu').style.font
Weight = 600;
document.getElementById('CreateBlankMenu').innerHT
ML = "&#149;
"+document.getElementById('CreateBlankMe
nu').innerHTML; </script>
<

```

Текст підпрограми для створення бланку в системі

```

<?php
error_reporting(0);
session_start();
if($_SESSION['connected']
== false){
    header ('Location: /prestige/Login.php'); // перенаправление на
    нужную страницу exit();
    $userid = $_SESSION['userid'];
    $link = mysqli_connect($_SESSION['dbip'],
$_SESSION['username'],
$_SESSION['userpassword'],$_SESSION['dbname']);
    mysqli_set_charset($link, "utf8");
    $SD = mysqli_fetch_array(mysqli_query($link, "SELECT `subdivisionid`
from `my_users` where `id` = ".$userid))[0];
    mb_internal_encodin
g("UTF-8"); }

function
expectionfound($thisl
ink){ echo
mysqli_error($thislin
k);
mysqli_rollback($thi
slink); die();
}
$form_seried = $_POST["form_seried"];
if($form_seried){ $form_seried = mb_strtoupper( $form_seried, 'UTF-8');}

mysqli_autocommit($link,FALSE);
$num = strlen((string)$_POST["form_minnumber"]);
for($i = $_POST["form_minnumber"]; $i <= $_POST["form_maxnumber"]; $i++){
    $sql = 'INSERT INTO `bso`(`BSO_ID`,`BSO_Series`,
`BSO_Number`,`BSO_StatusID`,`BSO_DateOfCreation`,`BSO_CompanyID`,
`BSO_UserID`,`BSO_ProductID`)
VALUES (NULL, "".$form_seried.", "".$str_pad($i, $num, '0',
STR_PAD_LEFT).", "1", "".$_POST["form_date"].", "".$_POST["form_company"].",
"".$_POST["form_user"].", "".$_POST["form_product"].");
    mysqli_query($link, $sql) or expectionfound($link);
    $sql2 = 'UPDATE bso_logs SET `comment` =
"".$_POST["form_comment"].", `DateChanged` = "".$_POST["form_date"]." ORDER BY id
DESC LIMIT 1';
    mysqli_query($link, $sql2) or
expectionfound($link); }

```

Текст підпрограми для звіту по бланкам на екран

```
<?php
error_reporting(0);
session_start();
if($_SESSION['connected']
== false){
    header ('Location: /prestige/Login.php'); // перенаправлення на
    необхідну сторінку exit();
    $userid = $_SESSION['userid'];
    $link = mysqli_connect($_SESSION['dbip'], $_SESSION['username'],
    $_SESSION['userpassword'],$_SESSION['dbname']);
    mysqli_set_charset($link, "utf8"); $SP = $_SESSION['usersd'];
    $Ierarhic = $_SESSION['userierarhic']; }
?>
<!DOCTYPE html> <html lang="ru"> <head>
<meta charset="utf-8"> <title>Вигрузка бланків</title> <style>
body{
margin: 0px; }
#output { margin:0px; padding:0px;
font-family: "Trebuchet MS", Arial, Helvetica, sans-serif; font-size: 12px;
border-collapse: collapse; width: 100%;
}
#output td, #customers
th { border-left: 1px
solid #ddd; border-
top: 1px solid #ddd;
border-bottom: 1px
solid #ddd; padding:
8px;
}
#output tr:nth-child(odd){background-color:
AliceBlue;} #output tr:hover {background-
color: #b8e1ec;}
#output th {
padding-
top: 12px;
padding-
bottom:
12px; text-
align: left;
background-color:
LightSeaGreen; color:
white;
}
#output th:hover{
cursor:
pointer; }
date_default_timezone_set('Europe/Kiev');
$form_date1 = new
DateTime($_POST['form_date1']);
$form_date2 = new
DateTime($_POST['form_date2']);
$form_date2->add(new
```

```

DateInterval('P1D'));
if($_POST["form_series"]){
    $form_series = $_POST["form_series"];
    $form_series = mb_strtoupper( $form_series,
'UTF-8' ); }

```

Текст підпрограми для звіту по бланкам в CSV

<?php

```

error_reporting(0);
session_start();
if($_SESSION['connected']
== false){
    header ('Location: /prestige/Login.php'); // перенаправление на
    нужную страницу exit();
    $userid = $_SESSION['userid'];
    $link = mysqli_connect($_SESSION['dbip'],
$_SESSION['username'],
$_SESSION['userpassword'],$_SESSION['dbname']);
    mysqli_set_charset($link, "utf8");
    $SSD = mysqli_fetch_array(mysqli_query($link, "SELECT `subdivisionid`
from `my_users` where `id` = ".$userid))[0];
    }
header("Content-type: text/csv");
header("Content-Disposition: attachment; filename=Blanks ".date("Y-m-d
h:m:s").".csv"); header("Pragma: no-cache");
header("Expires: 0");
mb_internal_encoding("UTF-
8");
date_default_timezone_set('Euro
pe/Kiev');
    $form_date1 = new
    DateTime($_POST['form_date1']);
    $form_date2 = new
    DateTime($_POST['form_date2']);
    $form_date2->add(new
    DateInterval('P1D'));
    if($_POST["form_series"]){
        $form_series = $_POST["form_series"];
        $form_series = mb_strtoupper( $form_series,
'UTF-8' ); }
    $form_number1 =
    $_POST["form_number1"];
    $form_number2 =
    $_POST["form_number2"];
    $form_company =
    $_POST["form_company"];
    $form_status =
    $_POST["form_status"];
    $_SESSION['form_user'] =
    $_POST["form_user"];
    $form_salepoint =
    $_POST["form_salepoint"];
    $form_showdate =
    $_POST["form_showdate"];

```

```

$form_showseries =
$_POST["form_showseries"];
$form_shownumber =
$_POST["form_shownumber"];
$form_showproduct =
$_POST["form_showproduct"];
$form_showcompany =
$_POST["form_showcompany"];
$form_showstatus =
$_POST["form_showstatus"];
$form_showsubdivision =
$_POST["form_showsubdivision"];
$form_showsalepoint =
$_POST["form_showsalepoint"];
$form_showuser =
$_POST["form_showuser"];
$form_showcontract =
$_POST["form_showcontract"];
$form_showcomment =
$_POST["form_showcomment"];
// Формирование запроса, выбор
что выводить $query = "SELECT
";
$firstfilter = true;
if($form_showdate){ if($firstfilter) { $firstfilter = false; } else { $query =
$query.'; } $query = $query.(select
DATE_FORMAT(BSO_DateOfCreation,"%d/%m/%y"), (select
DATE_FORMAT(bso_logs.`DateChanged`,"%d/%m/%y") from bso_logs where
bso_logs.`bso_id` = `bso`.`BSO_ID` order by `DateChanged` DESC limit 1); }

```

Текст підпрограми для знаходження даних за ППН

```

<?php
session
_start();
$userid = $_SESSION['userid'];
$mysqli = new mysqli($_SESSION['dbip'],
$_SESSION['username'],
$_SESSION['userpassword'],$_SESSION['dbname']);
    if($mysqli-
    >connect_error) {
        exit('Could not
        connect');
    }
$mysqli->set_charset("utf8");
$sql = "SELECT count(*), `FullName`, `PhoneNumber`, `BDay`, `Adress`, `Documentid`,
`Seria`, `Number`, `DateGivement`, `WhoGave`, `ActualTo`, `DateZapysu`, `ZapysNumber`,
`DocumentNumber`, `ResidenceID`, `E-Mail`, `ClientTypeID`, `BankInfo` FROM
ref_dogovir_person_persons WHERE `tin` != '9999999999' and `tin` = ?";
$stmt = $mysqli-
>prepare($sql); $stmt-
>bind_param("s",
$_GET['q']); $stmt-
>execute();
$stmt->store_result();
$stmt->bind_result($count, $n, $p, $b, $a, $did, $ser, $num, $DG, $WG, $AT, $DZ, $ZN,

```

```

$DN, $r, $e, $c, $bank);
$stmt-
>fetch(
);
$stmt-
>close
();
$xml = new SimpleXMLElement('<xml/>'); $persson = $xml->addChild('person');
$persson->addChild('count', $count); $persson->addChild('FullName', $n); $persson-
>addChild('PhoneNumber', $p); $persson->addChild('BDay', $b); $persson-
>addChild('Adress', $a); $persson->addChild('DocumentId', $did); $persson-
>addChild('Ser', $ser); $persson->addChild('Num', $num); $persson->addChild('DG',
$DG); $persson->addChild('WG', $WG); $persson->addChild('AT', $AT);
$persson->addChild('DZ', $DZ); $persson->addChild('ZN', $ZN); $persson-
>addChild('DN', $DN); $persson->addChild('ResidenceID', $r); $persson-
>addChild('E-Mail', $e); $persson->addChild('ClientType', $c); $persson-
>addChild('BankInfo', $bank

```

<?php

```

session_start(); // Початок
сесії $username =
$_POST["form_username"]
;
$_SESSION['username'] =
$_POST["form_username"];
$_SESSION['userpassword'] =
$_POST["form_password"];
$_SESSION['dbip'] = "127.0.0.1";
$_SESSION['dbname'] = "prestige";
if($link = mysqli_connect($_SESSION['dbip'], $_SESSION['username'],
$_SESSION['userpassword'], $_SESSION['dbname'])) { // спроба підключення до БД
    $r = mysqli_fetch_array(mysqli_query($link, "SELECT `id`, `ierarhic`,
`subdivisionid` from `my_users` where `username` = '". $username. "'")); // збереження
необхідних даних
    $_SESSION['userid'] =
    $r[0];
    $_SESSION['userierarhic'
    ] = $r[1];
    $_SESSION['usersd'] =
    $r[2];
    mysqli_close($link);
    $_SESSION['connected']
    = true;
}
{
    $_SESSION = array();
    session_destroy();
    session_start();
    $_SESSION['connecte
    d'] = false;
    $_SESSION['result'] =

```

```

        "error";
    }
    header ('Location: index.php'); // перенаправление на
    нужную страницу exit();
?>

```

Текст підпрограми для відображення сторінки створення бланку

```

<?php
    error_reporting(0);
    session_start();
    if($_SESSION['connected']
    == false){
        header ('Location: /prestige/Login.php'); // перенаправление на
        нужную страницу exit();
    }
    e
    l
    s
    e
    {
        $userid = $_SESSION['userid'];
        $link = mysqli_connect($_SESSION['dbip'],
$_SESSION['username'],
$_SESSION['userpassword'],$_SESSION['dbname']);
        mysqli_set_charset($link, "utf8");
        $SD = mysqli_fetch_array(mysqli_query($link, "SELECT `subdivisionid`
from `my_users` where `id` = ".$userid))[0];
        if(mysqli_fetch_array(mysqli_query($link, "SELECT `AllowAddBlanks`
from `roles` where `UID` = ".$userid.""))[0] == 0){
            header ('Location: /prestige/index.php'); // перенаправление
на нужную страницу
            e
            xit(); }
        }
    }
?>
<html>
<!DOC
TYPE
html>
<html
lang="r
u">
<head>
<link rel="shortcut icon" href="/prestige/Icon.ico"
type="image/x-icon"> <link href="/prestige/css/main.css"
rel="stylesheet" />
<link href="/prestige/css/css.css" rel="stylesheet" />
<link href="/prestige/css/defaultpage.css"
rel="stylesheet" /> <meta charset="utf-8">
<title>Створення
бланку</title>
<script>
    function tryToCreate(){
        var AllFine = true;
        if(document.getElementById('form_company').value == 0 &&

```

```

        AllFine == true){ AllFine = false;
        alert("Перевірте введені
данні(компанія)"); }
if(document.getElementById('form_product').value == 0 &&
AllFine == true){ AllFine = false;
alert("Перевірте введені данні(продукт)");

</td> </tr>
<tr class = "trinput"> <td>
<select id="form_company" name="form_company" form="my_form"> <option value
= 0 disabled selected>Виберіть компанію</option> <?php
$result = mysqli_query($link, 'SELECT CompanyID, CompanyName FROM
ref_bso_companyid'); while($row = mysqli_fetch_array($result)){
echo '<option value = "'.$row[0]."'>'.$row[1].</option>'; }
>
<select id="form_product" name="form_product" form="my_form"> <option value = 0
disabled selected>Виберіть продукт</option> <?php
$result = mysqli_query($link, 'SELECT ID, productid, simpleDescription FROM
ref_bso_productid where blank = 1 order by `productid`');
while($row = mysqli_fetch_array($result)){
echo '<option value = "'.$row[0]."'>'.$row[2].</option>'; }
?>
</td> </tr>
<tr class = "trtitle"> <td>
Дата створення бланку </td>
<td>
Користувач </td>
</tr>
<tr class = "trinput"> <td>
<input type="date" value="<?php echo date("Y-m-d");?>" id="form_date"
name="form_date" form="my_form">
<select id = "form_user" name = "form_user" form="my_form"> <?php
$result = mysqli_query($link, 'SELECT id, names FROM my_users where `id` = 1');
while($row = mysqli_fetch_array($result)){
echo '<option value = "'.$row[0]."'>'.$row[1].</option>'; }
<tr class = "trtitle"> <td>
Серія бланку </td>
</tr>
<tr class = "trinput"> <td>
<input type="text" id="form_seried" name="form_seried" pattern="^[0-9A-Яа-
яЁё]+$" <?PHP if($_SESSION['seried']){ echo (' value = "'.$_SESSION['seried']."' ');
$_SESSION['seried'] = null;}?> form="my_form"> </td>
</tr>
<tr class = "trtitle"> <td>
Мінімальне значення номера </td>
<td>
Максимальне значення номера </td>
</tr>
<tr class = "trinput">
<td>
<input type="text" id="form_maxnumber" name="form_maxnumber" pattern="[0-
9]+$" <?PHP if($_SESSION['num2']){ echo (' value = "'.$_SESSION['num2']."' ');
$_SESSION['num2'] = null;}?> required form="my_form">

```



```

</td> </tr>
<tr class = "trtitle"> <td>
Коментар </td>
</tr>
<tr class = "trinput">
<td colspan = '2'>
<input type = "text" <?php if($_SESSION['com']){echo'value =
"'.$_SESSION['com'].'"; $_SESSION['com'] = NULL;}?> id="form_comment"
name="form_comment" form="my_form">
</td> </tr>
<tr class =
"trin
put"
>
<td
>
<input type="submit" id = "confirm" form="my_form" hidden>
<input type="button" class = "submitbutton" value="СТВОРИТИ" onclick =
"tryToCreate()"> </td>>
displ('Blanks');
document.getElementById('CreateBlankMenu').style.font
Weight = 600;
document.getElementById('CreateBlankMenu').innerHT
ML = "&#149;
"+document.getElementById('CreateBlankMe
nu').innerHTML; </script>
<
<?php
session_start(); // Початок
сеції $username =
$_POST["form_username"]
;
$_SESSION['username'] =
$_POST["form_username"];
$_SESSION['userpassword'] =
$_POST["form_password"];
$_SESSION['dbip'] = "127.0.0.1";
$_SESSION['dbname'] = "prestige";
if($link = mysqli_connect($_SESSION['dbip'], $_SESSION['username'],
$_SESSION['userpassword'], $_SESSION['dbname'])) { // спроба підключення до БД
    $r = mysqli_fetch_array(mysqli_query($link, "SELECT `id`, `ierarhic`,
`subdivisionid` from `my_users` where `username` = '".$_SESSION['username']."'")); // збереження
необхідних даних
    $_SESSION['userid'] =
    $r[0];
    $_SESSION['userierarhic'
    ] = $r[1];
    $_SESSION['usersd'] =
    $r[2];
    mysqli_close($link);
    $_SESSION['connected']
    = true;
}
{
    $_SESSION = array();

```

```

        session_destroy();
        session_start();
        $_SESSION['connected'] = false;
        $_SESSION['result'] =
            "error";
    }
    header ('Location: index.php'); // перенаправление на
    нужную страницу exit();
?>

```

Текст підпрограми для відображення сторінки створення бланку

```

<?php
    error_reporting(0);
    session_start();
    if($_SESSION['connected']
    == false){
        header ('Location: /prestige/Login.php'); // перенаправление на
        нужную страницу exit();
    }
    else {
        $userid = $_SESSION['userid'];
        $link = mysqli_connect($_SESSION['dbip'],
        $_SESSION['username'],
        $_SESSION['userpassword'],$_SESSION['dbname']);
        mysqli_set_charset($link, "utf8");
        $SD = mysqli_fetch_array(mysqli_query($link, "SELECT `subdivisionid`
        from `my_users` where `id` = ".$userid))[0];
        if(mysqli_fetch_array(mysqli_query($link, "SELECT `AllowAddBlanks`
        from `roles` where `UID` = ".$userid.""))[0] == 0){
            header ('Location: /prestige/index.php'); // перенаправление
            на нужную страницу
            exit(); }
        }
?>
<html>
<!DOCTYPE
html>
<html
lang="ru"
>
<head>
<link rel="shortcut icon" href="/prestige/Icon.ico"
type="image/x-icon"> <link href="/prestige/css/main.css"
rel="stylesheet" />
<link href="/prestige/css/css.css" rel="stylesheet" />
<link href="/prestige/css/defaultpage.css"
rel="stylesheet" /> <meta charset="utf-8">
<title>Створення

```

бланку</title>

<script>

```
function tryToCreate(){
    var AllFine = true;
    if(document.getElementById('form_company').value == 0 &&
        AllFine == true){ AllFine = false;
        alert("Перевірте введені
данні(компанія)"); }
    if(document.getElementById('form_product').value == 0 &&
        AllFine == true){ AllFine = false;
        alert("Перевірте введені данні(продукт)");
```

</td> </tr>

<tr class = "trinput"> <td>

```
<select id="form_company" name="form_company" form="my_form"> <option value
= 0 disabled selected>Виберіть компанію</option> <?php
$result = mysqli_query($link, 'SELECT CompanyID, CompanyName FROM
ref_bso_companyid'); while($row = mysqli_fetch_array($result)){
echo '<option value = "'. $row[0]."'>'. $row[1]. '</option>'; }
>
```

```
<select id="form_product" name="form_product" form="my_form"> <option value = 0
disabled selected>Виберіть продукт</option> <?php
$result = mysqli_query($link, 'SELECT ID, productid, simpleDescription FROM
ref_bso_productid where blank = 1 order by `productid`');
while($row = mysqli_fetch_array($result)){
echo '<option value = "'. $row[0]."'>'. $row[2]. '</option>'; }
?>
```

</td> </tr>

<tr class = "trtitle"> <td>

Дата створення бланку </td>

<td>

Користувач </td>

</tr>

<tr class = "trinput"> <td>

```
<input type="date" value="<?php echo date("Y-m-d");?>" id="form_date"
name="form_date" form="my_form">
```

```
<select id = "form_user" name = "form_user" form="my_form"> <?php
$result = mysqli_query($link, 'SELECT id, names FROM my_users where `id` = 1');
while($row = mysqli_fetch_array($result)){
echo '<option value = "'. $row[0]."'>'. $row[1]. '</option>'; }
<tr class = "trtitle"> <td>
```

Серія бланку </td>

</tr>

<tr class = "trinput"> <td>

```
<input type="text" id="form_seried" name="form_seried" pattern="^[0-9A-Яа-
яЁё]+ $" <?PHP if($_SESSION['seried']){ echo (' value = "'. $_SESSION['seried']. "' );
$_SESSION['seried'] = null;}?> form="my_form"> </td>
```

</tr>

<tr class = "trtitle"> <td>

Мінімальне значення номера </td>

<td>

Максимальне значення номера </td>

</tr>

<tr class = "trinput">

```

<td>
<input type="text" id="form_maxnumber" name="form_maxnumber" pattern="[0-9]+$" <?PHP if($_SESSION['num2']){ echo (' value = "'. $_SESSION['num2']. "' '); $_SESSION['num2'] = null;}?> required form="my_form">
</td> </tr>
<tr class = "trtitle"> <td>
Комментар </td>
</tr>
<tr class = "trinput">
<td colspan = '2'>
<input type = "text" <?php if($_SESSION['com']){echo'value = "'. $_SESSION['com']. "'"; $_SESSION['com'] = NULL;}?> id="form_comment" name="form_comment" form="my_form">
</td> </tr>
<tr class =
"trin
put"
>
<td
>
<input type="submit" id = "confirm" form="my_form" hidden>
<input type="button" class = "submitbutton" value="СТВОРИТИ" onclick =
"tryToCreate()"> </td>>
displ('Blanks');
document.getElementById('CreateBlankMenu').style.font
Weight = 600;
document.getElementById('CreateBlankMenu').innerHT
ML = "&#149;
"+document.getElementById('CreateBlankMe
nu').innerHTML; </script>
<

```

ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ

Ім'я файла	Опис
Пояснювальні документи	
Кваліфікаційна робота.doc	Пояснювальна записка. Документ Word.
Кваліфікаційна робота.pdf	Пояснювальна в форматі PDF
Програма	
Program.rar	Архів. Містить коди програми і откомпільовану програму
Презентація	
Презентація.ppt	Презентація