

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Комп'ютеризована система візуалізації розкладу дня для дитини з аутизмом

Виконав(ла): студент(ка) IV курсу, групи СІ-41

спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

Олійник М.Р.
(прізвище та ініціали)

(підпис)

Керівник Осухівська Г.М.

(підпис)

(прізвище та ініціали)

Нормоконтроль Тиш Є.В.

(підпис)

(прізвище та ініціали)

Завідувач кафедри Осухівська Г.М.

(підпис)

(прізвище та ініціали)

Рецензент Дмитроца Л.П.

(підпис)

(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

« ____ » _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

студенту Олійнику Максиму Романовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютеризована система візуалізації розкладу дня для дитини з аутизмом

Керівник роботи Кандидат технічних наук, доцент кафедри КС Осухівська Галина Михайлівна
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « ____ » _____ 2024 року № _____

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи Технічне завдання

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ 1. Аналіз технічного завдання

2. Проектна частина

3. Практична частина

4. Безпека життєдіяльності, основи охорони праці

Висновки. Список використаних джерел

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Блок-схема

2. Схема архітектурна

3. Схема підключення

4. Схема результатів роботи

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорона праці			

7. Дата видачі завдання 11.06.2024

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Розробка технічного завдання	01.02-03.02.24	Виконано
2	Аналіз технічного завдання	05.02-11.02.24	Виконано
3	Аналіз та обґрунтування можливих рішень	25.04-04.05.24	Виконано
4	Розробка структурної схеми	05.05-18.05.24	Виконано
5	Розробка схеми	03.06-09.06.24	Виконано
6	Розробка програмного забезпечення для проектованої системи	07.06-10.06.24	Виконано
7	Безпека життєдіяльності, основи охорони праці	17.06.24	Виконано
8	Оформлення кваліфікаційної роботи	19.06.24	Виконано
9	Попередній захист кваліфікаційної роботи	14.06.24	Виконано
10	Захист кваліфікаційної роботи	28.06.24	Виконано

Студент

_____ (підпис)

Олійник М.Р.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Осухівська Г.М.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Комп'ютеризована система візуалізації розкладу для дитини з аутизмом
// Кваліфікаційна робота бакалавра // Олійник Максим Романович //
Тернопільський національний технічний університет імені Івана Пулюя,
факультет комп'ютерно-інформаційних систем і програмної інженерії,
кафедра комп'ютерних систем та мереж, група СІ-41 // Тернопіль, 2024 // с.
– 52, рис. – 27, табл. – 3, аркушів А1 – 4, бібліогр. – 20.

Ключові слова: КОМП'ЮТЕРИЗОВАНА СИСТЕМА, RASBERRY PI,
РОЗКЛАД, ПЛАТА, ДОДАТОК, ПРОГРАМА.

Кваліфікаційна робота присвячена розробці системи, для візуалізації розкладу для дитини з аутизмом.

В результаті огляду та аналізу сучасних комп'ютеризованих засобів для підтримки дітей з аутизмом виявлено, що важливим аспектом є створення інтуїтивно зрозумілих і візуально привабливих розкладів. Розроблено структурну схему системи, яка включає в себе зручний користувацький інтерфейс, інтеграцію з мобільними пристроями для доступу батьків, а також можливість адаптації під індивідуальні потреби дитини. Описано алгоритм роботи системи, який включає в себе можливість налаштування розкладу, додавання візуальних підказок і нагадувань для дитини. Всі ці функції спрямовані на поліпшення адаптації дитини для щоденних завдань та зниження стресу, пов'язаного зі зміною діяльності.

ANNOTATION

Computer-aided Schedule Visualization System for a Child with Autism//
Bachelor's Qualification Work // Maksym Romanovych Oliinyk // Ternopil Ivan
Puluj National Technical University, Faculty of Computer Information Systems and
Software Engineering, Department of Computer Systems and Networks, Group CI-
41 // Ternopil, 2024 // pp. – 52, figs. – 27, tables. – 3, A1 sheets – 4, bibliography. –
20.

Keywords: COMPUTERIZED SYSTEM, RASPBERRY PI, SCHEDULE,
BOARD, APPLICATION, PROGRAM.

Qualification work is devoted to the development of a system for visualizing
a schedule for a child with autism..

As a result of the review and analysis of modern computerized tools to support
children with autism, it has been found an important aspect is the creation of intuitive
and visually appealing schedules. A structural diagram of the system has been
developed, which includes a user-friendly interface, integration with mobile devices
for parental access, and the ability to adapt to the individual needs of the child. The
system operation algorithm is described, which includes the ability to set up
schedules, add visual cues and reminders for the child. All these functions are aimed
at improving the child's adaptation to daily tasks and reducing stress associated with
changes in activities

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ.....	7
1.1 Аналіз вимог до комп'ютеризованої системи.....	9
1.2 Аналіз можливих рішень поставленого завдання.....	10
1.3 Огляд джерел і аналогів комп'ютеризованих систем.....	14
РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА.....	18
2.1 Розробка узагальненої структури комп'ютеризованої системи.....	18
2.2 Обґрунтування вибору апаратного забезпечення проектованого комп'ютерного засобу.....	20
2.3 Обґрунтування вибору програмного забезпечення КСРДДА.....	23
2.4 Проектування комп'ютерного засобу.....	28
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ.....	34
3.1 Реалізація програмного забезпечення.....	34
3.2 Апаратна реалізація.....	40
3.3 Тестування.....	42
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ.....	46
4.1 Долікарська допомога при отруєннях.....	46
4.2 Заходи щодо захисту установки від короткого замикання.....	48
ВИСНОВКИ.....	50
СПИСКИ ВИКОРИСТАННИХ ДЖЕРЕЛ.....	51
ДОДАТОК А. Технічне завдання.....	53
ДОДАТОК Б. Лістинг програми.....	58

						КС КРБ 123.124.000.000 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Олійник М.Р.			<i>Комп'ютеризована система візуалізації розкладу дня для дитини з аутизмом</i>	Літ.	Арк.	Акрушів
Перевір.		Осухівська Г.М.					5	1
Реценз.		Дмитроца Л.П.				<i>ТНТУ, каф. ФІС, гр. СІ-41</i>		
Н. контр.		Тиш Є.В.						
Затверд.		Осухівська Г.М.						

ВСТУП

Сьогодні, в епоху швидкого розвитку цифрових технологій, застосування інноваційних рішень дозволяє покращити якість життя людей з різними потребами. Особливу увагу заслуговують технології, що сприяють соціальній інтеграції та полегшенню повсякденного життя людей з особливими потребами, зокрема дітей з аутизмом.

Діти з аутизмом часто стикаються з труднощами в організації свого розпорядку дня та виконанню цілком звичних дій, що може призводити до підвищеного рівня стресу та тривоги. Традиційні методи, такі як роздруковані розклади чи паперові нотатки [1-3], часто є недостатньо гнучкими та не забезпечують можливість миттєвого оновлення інформації. В цьому контексті, розробка комп'ютеризованої системи візуалізації розкладу набуває особливої актуальності. Використання такої системи дозволяє не лише забезпечити чітке та зрозуміле представлення інформації, але й оперативно вносити зміни до розкладу, що є надзвичайно важливим для підтримки стабільного та передбачуваного середовища для дітей з аутизмом.

Ця кваліфікаційна робота бакалавра присвячена розробці комп'ютеризованої системи візуалізації розкладу для дитини з аутизмом. Її метою є створення інтерактивного, зручного та адаптивного інструменту, який допоможе здійснити поведінкову корекцію дітей з такими розладами, сприяючи зменшенню рівня їх тривожності і вразливості, підвищуючи рівень їх самостійності.

					КС КРБ 123.124.000.000 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Аналіз вимог до комп'ютеризованої системи

У кваліфікаційній роботі розробляється комп'ютеризована система розкладу дня для дитини з аутизмом (КСРДДА). Метою розробки системи є створення інтерактивного, зручного та адаптивного інструменту, який допоможе дитині з особливими потребами легше орієнтуватись у своєму розпорядку дня, зменшити рівень тривожності та підвищити рівень самостійності, а батькам – надає інструмент для додаткового контролю за виконанням побутових дій дитини.

КСРДДА повинна функціонувати у автономному режимі, будучи незалежною від зовнішніх фізичних дій і керуватись виключно через веб-протокол HTTP. Головними вимогами до розроблюваної системи є: візуалізація актуального завдання на визначений час, можливість віддаленого керування системою батьками, зручний інтерфейс, безпечність у використанні та надійність.

КСРДДА повинна надавати точну та своєчасну інформацію про розклад дня та наступні заплановані завдання. Оскільки, відповідно до досліджень неврологів, що займаються корекцією поведінки дітей з аутизмом, нормований графік дня, із послідовними задачами – є критично важливим, тому що чітка структура дня допомагає їм орієнтуватися у своїй діяльності та знижує рівень тривожності [1].

Інформація про поточне та заплановане завдання для виконання дитиною повинне відображатися на екрані у зрозумілому форматі з використанням відповідних іконок та текстових підказок.

					КС КРБ 123.124.000.000 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		Олійник М.Р.			<i>Аналіз технічного завдання</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушіє</i>
<i>Перевір.</i>		Осухівська Г.М.					7	11
<i>Реценз.</i>		Дмитроца Л.П.				<i>ТНТУ, каф. ФІС, гр. СІ-41</i>		
<i>Н. контр.</i>		Тиш Є.В.						
<i>Затверд.</i>		Осухівська Г.М.						

У КСРДДА необхідно забезпечити можливість дистанційного віддаленого управління, що дозволить батькам або опікунам контролювати виконання завдань та вносити зміни до розкладу в режимі реального часу. Це може бути реалізовано за допомогою мобільного додатку.

Систему необхідно розробити з урахуванням максимальної зручності для користувачів. Інтерфейс має бути інтуїтивно зрозумілим, а сам пристрій – компактним та легким у використанні. Важливо, щоб усі елементи були доступні та зрозумілі як дітям, так і дорослим, що спрощує процес навчання та використання системи.

КСРДДА повинна відповідати всім вимогам безпеки, включаючи електробезпеку, знижений рівень електромагнітного випромінювання та відсутність дрібних деталей, які можуть бути небезпечними для дітей.

Діти з аутизмом мають особливі потреби в організації свого дня [2]. Типовий перелік завдань дня для дитини у послідовності від ранку до вечору, і відповідні вимоги до системи, щодо реалізації коректного відображення даної події наведено в табл. 1.1., яка сформована відповідно до рекомендацій поведінкових дитячих терапевтів [2].

Таблиця 1.1 – Розклад дня дитини та вимоги системи щодо його реалізації

Дія	Вимога до системи
Пробудження	Система повинна забезпечити нагадування про час пробудження, завдяки звуковому сигналу та допомоги розпочати день без зайвих зусиль.
Застелення ліжка	Після пробудження дитині необхідно навчитися застеляти своє ліжко. Система відображає відповідну картинку.

Дія	Вимога до системи
Ранкові гігієнічні процедури	Система нагадує про необхідність чищення зубів за допомогою зображень.
Сніданок	Система нагадує дитині про час прийому їжі, відображаючи зображення сніданку. Вона також включає голосові підказки.
Школа або заняття	Визначений час для відвідування школи або інших освітніх заходів. Система повинна нагадувати про це зображенням та звуковим сигналом.
Повернення додому та обід	Обідній час за допомогою зображень та звукових сигналів, допомагаючи дитині дотримуватися режиму харчування.
Відпочинок	Система відображає зображення відповідних видів діяльності, таких як ігри або інші заняття, допомагаючи дитині організувати свій вільний час.
Вечеря	Система нагадує дитині про час прийому їжі, відображаючи зображення вечері.

Дія	Вимога до системи
Підготовка до сну	Система відображає відповідні зображення та надає звукові сигнали, які нагадують дитині про необхідність підготовки до сну.
Читання казки на ніч	Система нагадує про цей ритуал за допомогою зображень та звукових сигналів, створюючи комфортну атмосферу для завершення дня.
Сон	Система відображає зображення, що сигналізує про початок сну, і може використовувати м'які звукові сигнали або мелодії, щоб допомогти дитині заснути.

Отже, розробка комп'ютеризованої системи розкладу дня для дитини з аутизмом, передбачає врахування всіх вищезгаданих вимог та характеристик для забезпечення максимальної ефективності та зручності у використанні.

1.2 Аналіз можливих рішень поставленого завдання

КСРДДА можна реалізувати, використовуючи різні програмні та апаратні засоби. Але, враховуючи вимоги до системи щодо компактності і зручності до використання, для візуалізації задач дня дитини можна використати такі монітори як Raspberry Pi Display, Adafruit 7" Touchscreen Display, Waveshare 10.1" HDMI LCD.

Raspberry Pi 3 Display має компактний розмір, який ідеально підходить для проектів, де важливі портативність та зручність використання [4]. Цей дисплей

					КС КРБ 123.124.000.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

підтримує сенсорне управління, що спрощує взаємодію з системою. Вартість Raspberry Pi 3 Display є значно нижчою порівняно з деякими іншими аналогами, що робить його доступним для широкого кола користувачів. Крім того, налаштування та інтеграція Raspberry Pi 3 Display є простими завдяки повній сумісності з іншими компонентами Raspberry Pi [4].

Adafruit 7" Touchscreen Display [5] має аналогічні характеристики щодо розміру екрану та підтримки сенсорного управління як Raspberry Pi Display, однак, цей дисплей є значно дорожчим у порівнянні з моніторами Raspberry Pi, що може бути критичним фактором для проектів з обмеженим бюджетом. Крім того, налаштування та інтеграція Adafruit дисплея можуть бути складнішими, що вимагає більше часу та зусиль для початку роботи.

Waveshare 10.1" HDMI LCD пропонує більший екран з вищою роздільною здатністю, що може бути корисним для певних завдань, де потрібна велика площа екрана [5]. Проте, збільшені розміри дисплея знижують його портативність та компактність, що є серйозним недоліком для комп'ютеризованої системи, яка має мати можливість, бути розміщеною, практично у будь-якому місці перебування дитини, наприклад, місця її проживання, навчання, дозвілля чи побуту. Хоча додаткові порти підключення надають більше можливостей для розширення, але це також збільшує вартість та складність налаштування.

Отже, із вибраного списку, найбільш підходящим варіантом для комп'ютеризованої системи є Raspberry Pi 3 Display, за своїми параметрами, а саме, враховуючи баланс між вартістю, функціональністю та простотою використання.

Реалізація КСРДДА потребує логічного ядра. Розглянемо, існуючі варіанти на ринку.

Arduino Uno є популярним вибором для простих мікроконтролерних проектів так як він простий у використанні та має велику кількість доступних бібліотек і модулів. Це зручна платформа для початківців і для невеликих проектів, які потребують лише базових обчислювальних потужностей та кількох входів/виходів. Arduino Uno дозволяє легко програмувати та тестувати різні додатки, але має

					КС КРБ 123.124.000.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

обмежені обчислювальні можливості та пам'ять, що може бути недостатньо для складніших завдань [6].

BeagleBone Black надає більшу потужність обчислень та більшу кількість входів/виходів, що робить його більш придатним для складніших проектів. Цей пристрій має потужний процесор, більшу оперативну пам'ять та підтримує різні операційні системи, включаючи Linux [7].

Raspberry Pi 3 Uno пропонує значні обчислювальні можливості, підтримку різних операційних систем та широкий спектр периферійних пристроїв. Ця плата є легкою у налаштуванні і підключенні, що робить її ідеальною для різноманітних проектів, включаючи складніші завдання, ніж ті, які може виконати Arduino Uno [4].

Було обрано саме Raspberry Pi 3, через поєднання потужності і компактного розміру. Також, тому, що вона значно легша у налаштуванні і підключенні, у порівнянні із своїми аналогами, що і робить її найбільш підходящою для реалізації КСРДДА.

Комбінація Raspberry Pi Display та Raspberry Pi 3 дозволяє гнучко програмувати функціонал системи розкладу дня, включаючи відображення різних задач на протязі доби для дітей з аутизмом.

На рис. 1.1 зображено Raspberry Pi Display.

					КС КРБ 123.124.000.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

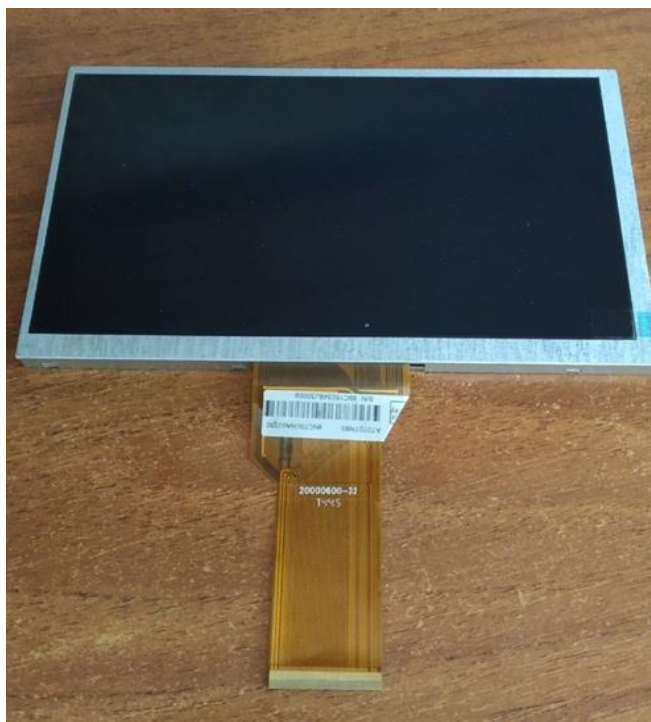


Рисунок 1.1 – Монітор Raspberry Pi Display

Цей монітор доволі компактний, що дозволяє всій комп'ютеризованій системі використовуватись практично у будь-якому місці, залежно від потреб дитини, або бажання батьків. Він може бути розташований у дитячій кімнаті, на кухні чи у вітальні, що забезпечує максимальну зручність та доступність для дитини. Крім того, його невеликі розміри дозволяють легко змінювати місце розташування системи у разі необхідності, наприклад, для більш зручного використання під час різних видів діяльності дитини.

На рис. 1.2 зображено плату Raspberry Pi 3

					КС КРБ 123.124.000.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13



Рисунок 1.2 – Raspberry Pi 3 Uno

Даний елемент є ядром системи, що оснащений процесором ARM і різноманітними інтерфейсами підключення. Ця модель має 1 ГБ оперативної пам'яті та підтримує роботу з картами пам'яті microSD для зберігання даних [4].

1.3 Огляд джерел і аналогів комп'ютеризованих систем

Поведінковими неврологами було розроблено ряд комунікативних інструментів для дітей із аутизмом [1-3], таких як: система обміну комунікативними зображеннями, тайм-таблиці із зображеннями, карта впевненості, тощо.

Система обміну комунікативними зображеннями (PECS- Picture Exchange Communication System) – система, що включає використання картинок для комунікації (див рис.1.3). Діти вчаться обмінюватися картинками, щоб висловити свої потреби і бажання. Ця система також може бути адаптована для створення розкладів дня [4].











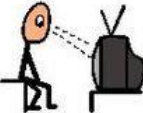





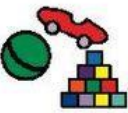



computer 	milk 	biscuit 	chocolate 	bread 
cereal 	school 	music 	apple 	exercise 
watch TV 	walk 	sit 	eat 	goodbye 
rest 	toys 	night 	cloudy 	sunny 

Рисунок 1.3 – Зображення PECS

Тайм-таблиці із зображеннями – система розпорядку дня, що має багато спільного із системою PECS, але дещо покращена, а саме разом і з картинками, відображає час доби у доволі зручній формі, що легко запам'ятовується (див.рис.1.4) [3]. Наприклад, на ранок існує картинка ранніх процедур дитини, разом із картинкою часу, із відповідним текстом, що дозволяє дитині, не лише краще організувати свій день, а поступово навчатись використанню відліку часу, у гайливій і ненав'язливій формі.

зображення, що допомагає дитині орієнтуватись у своєму розкладі без допомоги дорослих.

Комп'ютеризована система представляє прилад, що відображає, малюнки, які відповідають розкладу дитини. Розклад може встановлюватись батьками. А може, бути вибраний за замовчуванням.

В даній роботі запропоновано використання інформаційних технологій для покращення побуту дітей з особливими потребами, а також необхідного контролю їх дій з боку батьків. Така реалізація системи, використовуючи комп'ютеризовані засоби, дозволить створити інструмент, що, враховуючи рекомендації психотерапевтів і поведінкових неврологів, надасть необхідну і якісну підтримку дитині з аутизмом, сприяючи їх соціалізації та зменшуючи тривожності.

					КС КРБ 123.124.000.000 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА

2.1 Розробка узагальненої структури комп'ютеризованої системи

КСРДДА запропоновано реалізувати на основі Raspberry Pi 3 Uno з монітором Raspberry Pi Display. На рис. 2.1 зображена узагальнена архітектура комп'ютеризованої системи.

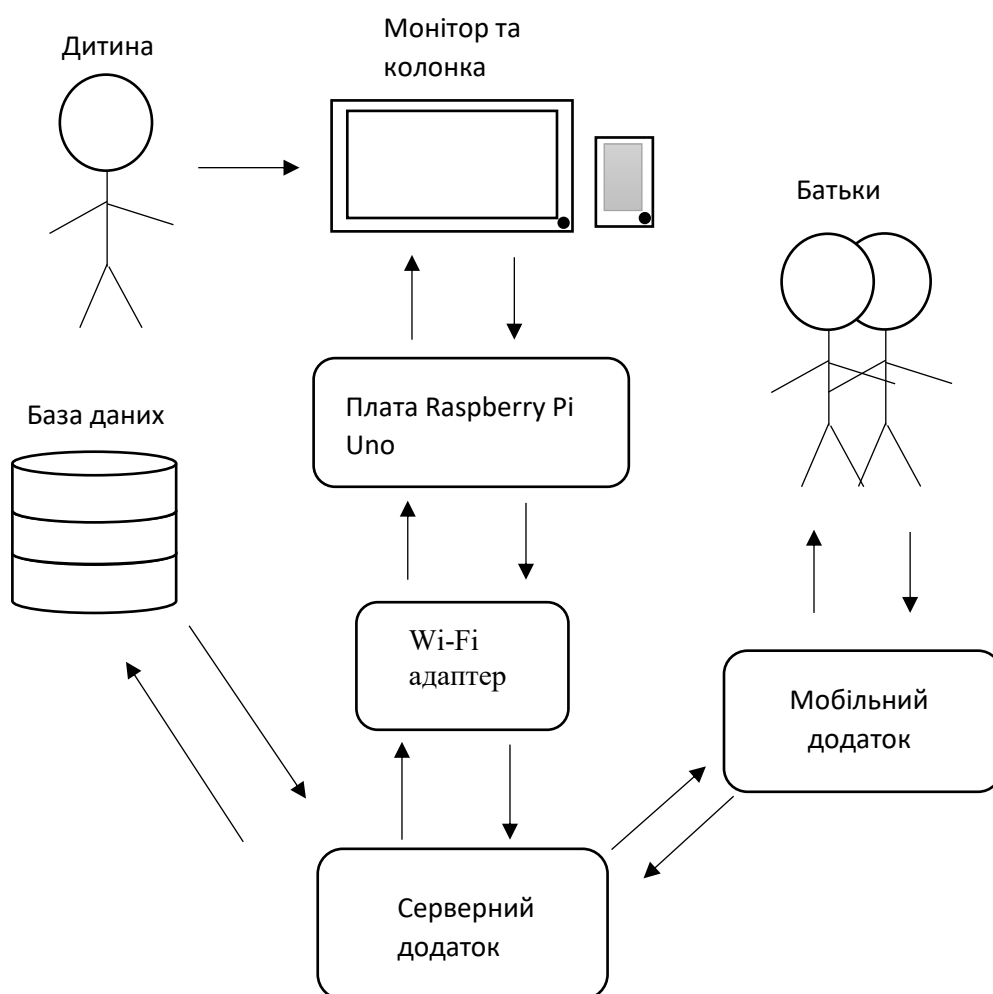


Рисунок 2.1 – Узагальнена архітектурна схема комп'ютеризованої системи розкладу дня для дитини з аутизмом

					КС КРБ 123.124.000.000 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Олійник М.Р.			Літ.	Арк.	Акрушів
Перевір.		Осухівська Г.М.				18	16
Реценз.		Дмитроца Л.П.			ТНТУ, каф. ФІС, гр. СІ-41		
Н. контр.		Тиш Є.В.					
Затверд.		Осухівська Г.М.					
					Проектна частина		

Розроблювана система включає у себе кілька основних компонентів, які забезпечують її функціонування. Складові системи:

- 1) Серверний додаток, який приймає і передає інформацію про час доби пристрою.
- 2) Мобільний додаток – це програма, за допомогою котрої, батьки можуть відправляти дані про розклад дня дитини на монітор
- 3) Wi-Fi адаптер – прилад, що дозволяє платі Raspberry Pi Uno приймати сигнали Wi-Fi.
- 4) Raspberry Pi – одноплатний комп'ютер, що є ядром системи. Виконує обробку даних так керує іншими компонентами КСРДДА.
- 5) Монітор та колонка – показує інформацію про актуальну задачу для дитини на даний момент доби і, за потреби, подає необхідний звук чи мелодію.
- 6) База даних – місце зберігання зображень, а також стандартних і створених батьками розкладів дня дитини.

Серверний додаток призначений для прийому та обробки даних від користувача і для відправки заданих батьками даних на монітор Raspberry Pi Display. Додаток реалізовано на мові програмування Java, через доволі велику швидкодію даної мови програмування. Додаток реалізовано на мові програмування Java, через доволі велику швидкодію даної мови програмування.

Мобільний додаток було розроблено на вбудованому середовищі розробки Android Studio. Даний додаток відправляє дані, а саме розклад дня, серверу, для подальшої візуалізації на моніторі Raspberry Pi Display, де його зможе спостерігати дитина.

Wi-Fi – адаптер тут забезпечує зв'язок плати Raspberry Pi 3 Uno із серверним додатком.

Адаптер підтримує функцію автоматичного підключення до раніше збережених мереж, що забезпечує безперервне з'єднання навіть у випадку перезапуску системи або тимчасових перебоїв у роботі мережі.

					КС КРБ 123.124.000.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

2.2 Обґрунтування вибору апаратного забезпечення проектного комп'ютерного засобу

Raspberry Pi 3 Uno – це одноплатний комп'ютер, який пропонує значно покращену продуктивність і функціональність порівняно з попередніми моделями серії. Цей пристрій оснащений чотириядерним процесором ARM Cortex-A53 з тактовою частотою 1.2 ГГц, що забезпечує високу обчислювальну потужність, яка є достатньою для виконання вимогливих завдань, таких як обробка зображень, робота з базами даних і виконання складних алгоритмів. Оперативна пам'ять об'ємом 1 ГБ також сприяє покращенню продуктивності системи, дозволяючи ефективно працювати з багатьма додатками одночасно.

Raspberry Pi 3 оснащений широким спектром портів і роз'ємів для підключення різноманітних периферійних пристроїв. 40-контактний GPIO роз'єм дозволяє підключати різноманітні сенсори, виконавчі пристрої та інші електронні компоненти, що робить цю плату ідеальною для проектів у галузі інтернету речей (IoT) [7].

Наявність 4 USB портів дозволяє підключати клавіатуру, мишу, флеш-накопичувачі та інші периферійні пристрої одночасно. HDMI вихід забезпечує можливість підключення до моніторів і телевізорів, дозволяючи використовувати Raspberry Pi як повноцінний медіацентр або настільний комп'ютер.

Raspberry Pi 3 Uno оснащений вбудованим модулем бездротового зв'язку Wi-Fi стандарту 802.11n та Bluetooth 4.1, що дозволяє легко підключати пристрій до бездротових мереж і використовувати Bluetooth-пристрої без необхідності додаткових адаптерів.

Raspberry Pi 3 Uno підтримує кілька операційних систем, включаючи Raspbian – спеціально адаптовану версію Debian для Raspberry Pi. Raspbian забезпечує повний набір програмного забезпечення для розробки, включаючи Python, Scratch, Java та інші мови програмування, що робить його ідеальним для навчальних цілей і розробки різноманітних проектів. Крім Raspbian, на Raspberry

					КС КРБ 123.124.000.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

Pi можуть бути встановлені інші дистрибутиви Linux, а також операційні системи на базі Android та навіть спеціальні медіа-центри, такі як OSMC та LibreELEC.

Однією з ключових переваг Raspberry Pi 3 Uno є його низьке енергоспоживання і характеристики (див. табл. 2.1), що є підходящими для компактної і зручної системи. Пристрій споживає значно менше енергії порівняно з традиційними настільними комп'ютерами, що робить його економічно вигідним варіантом для постійного використання. Це особливо важливо для проектів, де пристрій повинен працювати безперервно, таких як домашні сервери, системи моніторингу або інтелектуальні домашні пристрої.

Таблиця 2.1 – Характеристики плати Raspberry Pi 3 Uno

Параметр	Значення
Процесор	ARM Cortex-A53, чотириядерний, 1.2 ГГц
Оперативна пам'ять	1 ГБ LPDDR2
Порти USB	До 150 Мбіт/с
Відеовиходи	HDMI, композитний відеовихід
Накопичувач	MicroSD
Енергоспоживання	5 В, продовж. табл. 2.1
Інтерфейси	GPIO, UART, I2C, SPI, PWM
Розміри	85.6 мм × 56.5 мм × 17 мм

Розміри Raspberry Pi 3 Uno становлять лише 85.6 мм × 56.5 мм, що робить його надзвичайно компактним і портативним. Така компактність дозволяє легко інтегрувати плату в різні проекти, де простір є обмеженим, наприклад, у вбудованих системах, робототехніці або переносних пристроях. Компактний розмір також сприяє легкості транспортування, що важливо для мобільних проектів

або демонстраційних цілей. Завдяки своїм технічним характеристикам та вбудованій підтримці бездротових з'єднань, Raspberry Pi 3 Uno є найкращим вибором для

проекту комп'ютеризованої системи розкладу дня для дітей з аутизмом.

Візуалізація завдань дитини протягом дня, відбувається на моніторі Raspberry Pi Display, призначений для забезпечення комфортної роботи для комп'ютеризованої системи, оскільки володіє малими розмірами, що підвищує компактність системи. Також, однією із ключових переваг є функція подавлення синього світла, яка зменшує вплив шкідливих синіх випромінювань, дозволяючи безпечно користуватись монітором протягом довгого дня, що особливо важливо для дітей.

Монітор Raspberry Pi Display підтримує роздільну здатність до 800x480 пікселів для моделі з 7-дюймовим екраном. Це забезпечує достатню якість зображення для більшості завдань, таких як програмування, перегляд мультимедіа, та робота з текстовими документами. Сенсорний екран підтримує багатоточковий дотик, що дозволяє використовувати різні жести для управління додатками.

Монітор володіє характеристиками (див. табл. 2.2), що забезпечує легкість налаштування та підключення.

Переваги використання монітора Raspberry Pi Display: Економічність, компактність, портативність, функція подавлення синього світла, низьке енергоспоживання і простота підключення.

Raspberry Pi Display є доступними за ціною, що робить їх привабливим вибором для бюджетних проектів та навчальних цілей. Невеликі розміри моніторів дозволяють використовувати їх у портативних проектах та вбудованих системах, де важлива компактність. Зменшує вплив шкідливого синього випромінювання, що дозволяє тривале використання монітора без шкоди для очей. Наявність HDMI порту та підтримка plug-and-play забезпечують легкість підключення та налаштування монітора. Монітори споживають мало енергії, що є важливим для автономних систем та проектів з обмеженим енергопостачанням. Підтримка високої роздільної здатності дозволяє отримати чітке та деталізоване зображення,

					КС КРБ 123.124.000.000 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

що покращує загальне користувацьке враження. Raspberry Pi Display сумісні з різними операційними системами, такими як Raspbian, Ubuntu, та інші, що забезпечує гнучкість у виборі програмного забезпечення.

Таблиця 2.2 – Характеристики монітор Raspberry Pi Display

Параметр	Значення
Діагональ екрану	7 дюймів
Роздільна здатність	1920 x 1080 (Full HD)
Тип матриці	IPS
Час відгуку	5 мс
Кути огляду	178°/178°
Технологія подавлення синього світла	Так

Підключення плати Raspberry Pi 3 Uno до монітору здійснюється через роз'єм (PIN31) GP106, який використовується для передачі сигналів із процесора плати на монітор, забезпечуючи відображення інформації на екрані. Монітор має відповідні порти: D0, D1, D2, D3, D4, D5, D6, D7. Дані порти використовуються для передачі додаткових сигналів, необхідних для коректної роботи монітора. На рис. 2.2 зображено підключення плати Raspberry Pi 3 Uno до монітора.

2.3 Обґрунтування вибору програмного забезпечення КСРДДА

Для розробки програмного забезпечення на платформі Raspberry Pi 3 Uno, було обрано текстовий редактор Nano [4]. Приклад написання тестового коду у данному редакторі зображено на рис 2.3. Цей вибір обґрунтований кількома ключовими факторами.


```

GNU nano 2.5.3                               New Buffer                               Modified
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
^G Get Help      ^O Write Out    ^W Where Is    ^K Cut Text    ^J Justify    ^C Cur Pos
^X Exit          ^R Read File   ^\ Replace     ^U Uncut Text ^T To Spell   ^_ Go To Line

```

Рисунок 2.3 – Написання тестового коду у текстовому редакторі Nano

Nano є легким і швидким редактором, що особливо важливо для систем з обмеженими ресурсами, таких як Raspberry Pi.

Простота у використанні Nano робить його зручним для редагування коду, навіть у віддаленому режимі через SSH. Він не вимагає великих ресурсів для запуску, що дозволяє використовувати його на пристроях з невеликою обчислювальною потужністю.

Інтуїтивно зрозумілий інтерфейс, що не потребує тривалого навчання. Це дозволяє швидко освоїти його основні функції та почати роботу над проектом.

Підтримка базових функцій редагування, такі як пошук та заміна тексту, що полегшує процес налагодження коду.

Має функцію автоматичного відступу, що сприяє зручнішому форматуванню коду та зменшує кількість помилок, пов'язаних з неправильною структурою коду.

Можливість легко працювати з кількома файлами одночасно, що забезпечує ефективне керування великими проектами.

Підтримка синтаксичного підсвічування для багатьох мов програмування, і

особливо до Java, що полегшує читання та розуміння коду.

Nano є редактором з відкритим вихідним кодом, що гарантує його безкоштовне використання та можливість модифікації відповідно до потреб проекту.

Активна спільнота користувачів, що забезпечує доступ до великої кількості ресурсів та документації, що спрощує вирішення можливих проблем.

В порівнянні з іншими текстовими редакторами, такими як vim або emacs, Nano відрізняється менш складним інтерфейсом і меншою кількістю функцій, що робить його більш підходящим для швидкого редагування коду без необхідності вивчати складні команди.

Таким чином використання текстового редактору nano для використання Java-коду на Raspberry Pi забезпечує ефективність і простоту в роботі з обмеженими ресурсами системи, що робить його оптимальним вибором для даного проекту.

Для розробки програмного додатку, з якого відбувається контроль за розкладом дня дитини, використовується вбудоване середовище розробки – Android Studio.

Android Studio є офіційним середовищем розробки для створення додатків на платформі Android. Воно базується на середовищі розробки IntelliJ IDEA і надає інструменти для створення, тестування та налагодження мобільних додатків. Приклад написання коду у вбудованому середовищі Android Studio зображено на рис. 2.4.

					КС КРБ 123.124.000.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

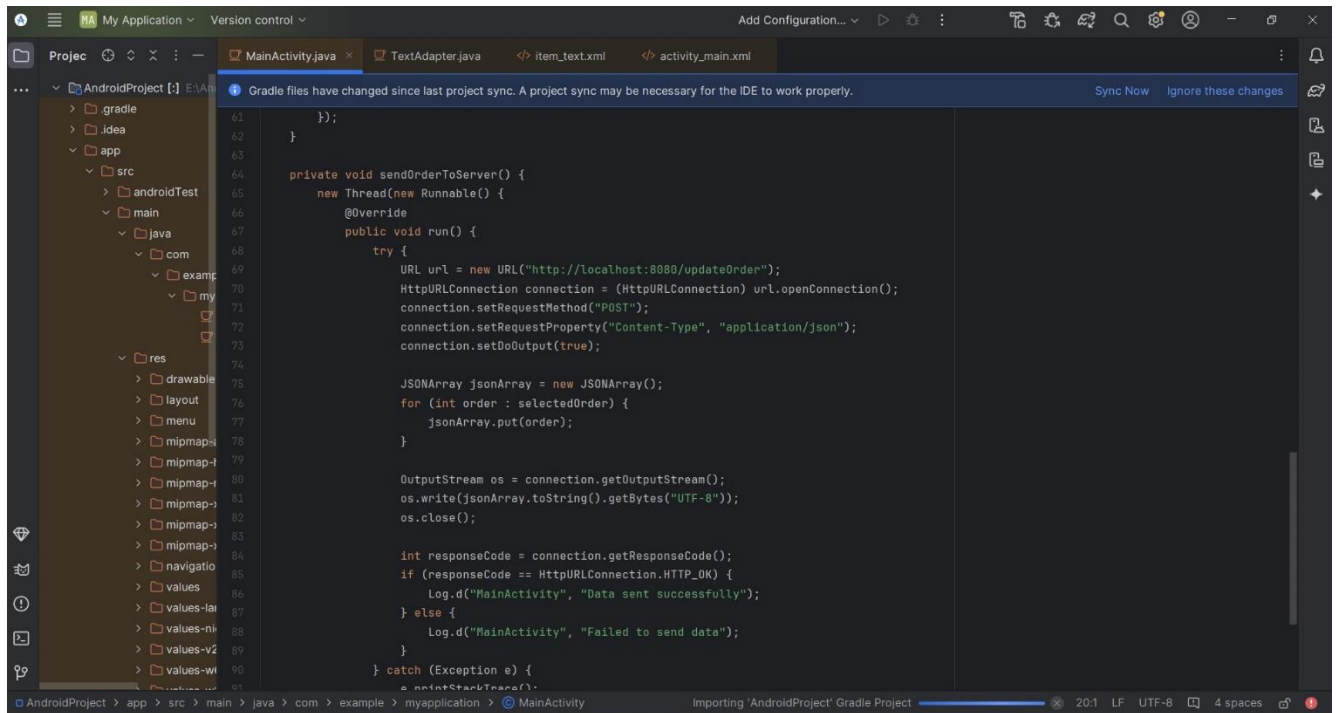


Рисунок 2.4 – Приклад написання коду КСРДДА у вбудованому середовищі Android Studio

Середовище володіє цілим рядом переваг.

Інтеграція з Android SDK: Повна інтеграція з Android SDK забезпечує доступ до всіх необхідних інструментів і бібліотек для розробки додатків.

Підтримка мови Java, Kotlin, а також інтеграція з іншими мовами програмування.

Вбудований емулятор дозволяє швидко тестувати додатки на різних версіях Android і на різних пристроях.

Набір інструментів для налагодження коду, аналізу продуктивності та оптимізації додатків.

Платформа підтримує JUnit для модульного тестування та Espresso для UI-тестування, що дозволяє забезпечити високу якість коду.

Інтеграція з системою збірки Maven дозволяє легко керувати залежностями та процесами збірки додатків.

Інструменти для автоматичного рефакторингу дозволяють підтримувати чистоту та зрозумілість коду, що важливо для довготривалих проектів.

					КС КРБ 123.124.000.000 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

Підтримка Git, SVN та інших систем контролю версій спрощує співпрацю в команді розробників.

Можливість розширення функціональності за допомогою плагінів дозволяє адаптувати середовище під індивідуальні потреби розробника.

Android Studio надає інтегрований доступ до офіційної документації Android, що дозволяє розробникам швидко знаходити необхідну інформацію та приклади коду безпосередньо під час розробки. Інструменти налагодження в Android Studio включають можливість встановлення точок зупину (breakpoints), перегляд змінних під час виконання коду та аналіз стеку викликів, що значно полегшує процес відлагодження програм.

Інтегрований інструмент для створення користувацьких інтерфейсів (UI) дозволяє розробникам швидко і зручно створювати та налаштовувати макети додатків за допомогою drag-and-drop інтерфейсу.

Android Studio пропонує великий набір готових шаблонів для створення нових проектів, що допомагає розробникам швидко почати роботу над новими додатками, використовуючи попередньо налаштовані структури.

Інструмент Layout Inspector дозволяє в реальному часі переглядати та аналізувати структуру макетів, виявляти проблеми та оптимізувати інтерфейси для різних типів пристроїв.

Завдяки можливості створення модульних проектів та бібліотек, Android Studio дозволяє легко перевикористовувати код та компоненти між різними додатками, що сприяє більш ефективному управлінню кодовою базою та зменшенню дублювання зусиль.

Дані переваги роблять Android Studio підходящим інструментом для розробки програмного додатку контролю за комп'ютеризованою системою.

					КС КРБ 123.124.000.000 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

2.4 Проектування комп'ютерного засобу

Проектування комп'ютеризованого засобу включає розробку та обґрунтування його структури, вибір компонентів, а також розробку блок-схеми КСРДДА і схеми підключення плати Raspberry Pi 3 Uno до монітора. Реалізація КСРДДА складається з кількох ключових етапів, починаючи з ініціалізації системи, бібліотек та датчиків. Після цього відбувається оголошення необхідних змінних для роботи системи, що включає налаштування середовища для подальших операцій. Цей етап забезпечує правильну підготовку та готовність системи до роботи.

Далі система встановлює з'єднання з сервером, де зберігається додаток, і починає збір інформації про розклад періодів дня. На цьому етапі перевіряється наявність отриманих даних. Якщо дані про періоди дня успішно отримані, система переходить до їх встановлення, інакше продовжує чекати на дані. Це забезпечує динамічне оновлення інформації та дозволяє системі адаптуватися до змін у розкладі в режимі реального часу. Після отримання даних відбувається конфігурація системи згідно з отриманими даними. Система відображає відповідну картинку на екрані, що відповідає поточному часу та розкладу, забезпечуючи візуальну підтримку для дитини.

Для забезпечення надійності та безперебійної роботи системи використовуються сучасні протоколи передачі даних, які гарантують швидкий та безпечний обмін інформацією між сервером і комп'ютеризованим засобом. Дані про розклад включають точний час початку та завершення кожного періоду, а також специфічні завдання та активності, що мають бути виконані дитиною. Використання ефективних алгоритмів обробки даних дозволяє мінімізувати затримки та забезпечити високу точність відображення розкладу на моніторі.

Після отримання даних відбувається конфігурація системи згідно з отриманими даними. Система відображає відповідну картинку на екрані, що відповідає поточному часу та розкладу, забезпечуючи візуальну підтримку для дитини. Цей процес включає в себе відображення графічних зображень, текстових

					КС КРБ 123.124.000.000 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

повідомлень та звукових сигналів, що допомагає дитині легше орієнтуватися у своєму розкладі.

Система переходить у режим очікування дії дитини, зокрема натискання клавіші Enter або Shift. У випадку натискання клавіші Enter, система відправляє повідомлення про виконання завдання батькам, забезпечуючи таким чином зворотний зв'язок. Це дозволяє батькам або опікунам отримувати своєчасну інформацію про виконання завдань дитиною, що важливо для підтримки ефективного моніторингу та корекції поведінки дитини. Реалізація комп'ютеризованої системи є важливим і в той же час складним етапом розробки.

Таким чином, опис блок-схеми включає початкову ініціалізацію системи, збір та перевірку даних, конфігурацію відображення, та інтерактивний контроль за виконанням завдань дитиною з зворотним зв'язком для батьків. Уся система інтегрована таким чином, щоб забезпечити максимально ефективно та зручне використання для підтримки щоденного розкладу дитини з аутизмом.

Алгоритм роботи комп'ютеризованої системи керування розкладом дня для дитини з аутизмом:

- 1) Початок – Ініціалізація системи, бібліотек і датчиків.
- 2) Оголошення змінних – Підготовка необхідних змінних для роботи системи.
- 3) Підключення до сервера – Встановлення з'єднання з сервером, де зберігається додаток.
- 4) Отримання даних від додатку – Збір інформації про розклад періодів дня з сервера.
- 5) Дані про періоди дня отримані? – Перевірка наявності отриманих даних.
 - а) Якщо так, система встановлює періоди дня.
 - б) Якщо ні, система продовжує чекати даних.
- 6) Встановлення періодів дня – Конфігурація системи згідно з отриманими даними.

					КС КРБ 123.124.000.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

7) Відображення розкладу – Відображення відповідної картинки на екрані, що відповідає поточному часу та розкладу.

8) Очікування дії дитини – Очікування натискання клавіші Enter або Shift.

9) Виконано завдання? – Перевірка дії дитини:

а) Якщо так, то відправляється повідомлення батькам про виконання дитинию завдання

б) Якщо ні, очікування дії дитини.

10) Оновлення даних – Періодичне оновлення даних про розклад з сервера.

11) Кінець – Завершення роботи системи.

Блок-схема зображена на рис. 2.5.

					КС КРБ 123.124.000.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

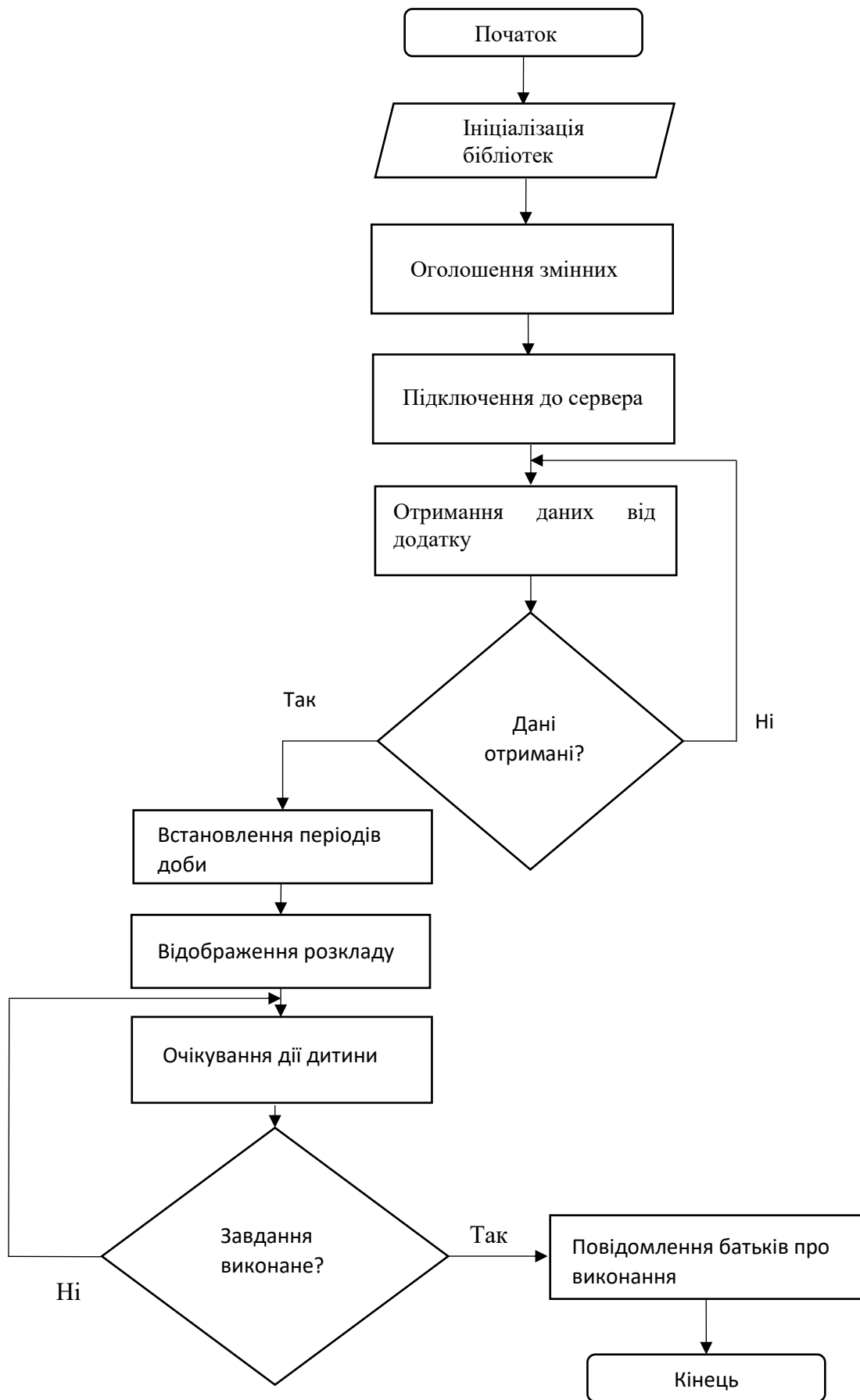


Рисунок 2.5 – Блок-схема комп'ютеризованої системи розкладу дня

Змн.	Арк.	№ докум.	Підпис	Дата

Підключення плати Raspberry Pi 3 Uno до монітору здійснюється через роз'єм (PIN31) GP106, який використовується для передачі сигналів із процесора плати на монітор, забезпечуючи відображення інформації на екрані. Монітор має відповідні порти: D0, D1, D2, D3, D4, D5, D6, D7. Дані порти використовуються для передачі додаткових сигналів, необхідних для коректної роботи монітора. На рис. 2.6 зображено підключення плати Raspberry Pi 3 Uno до монітора.

					КС КРБ 123.124.000.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

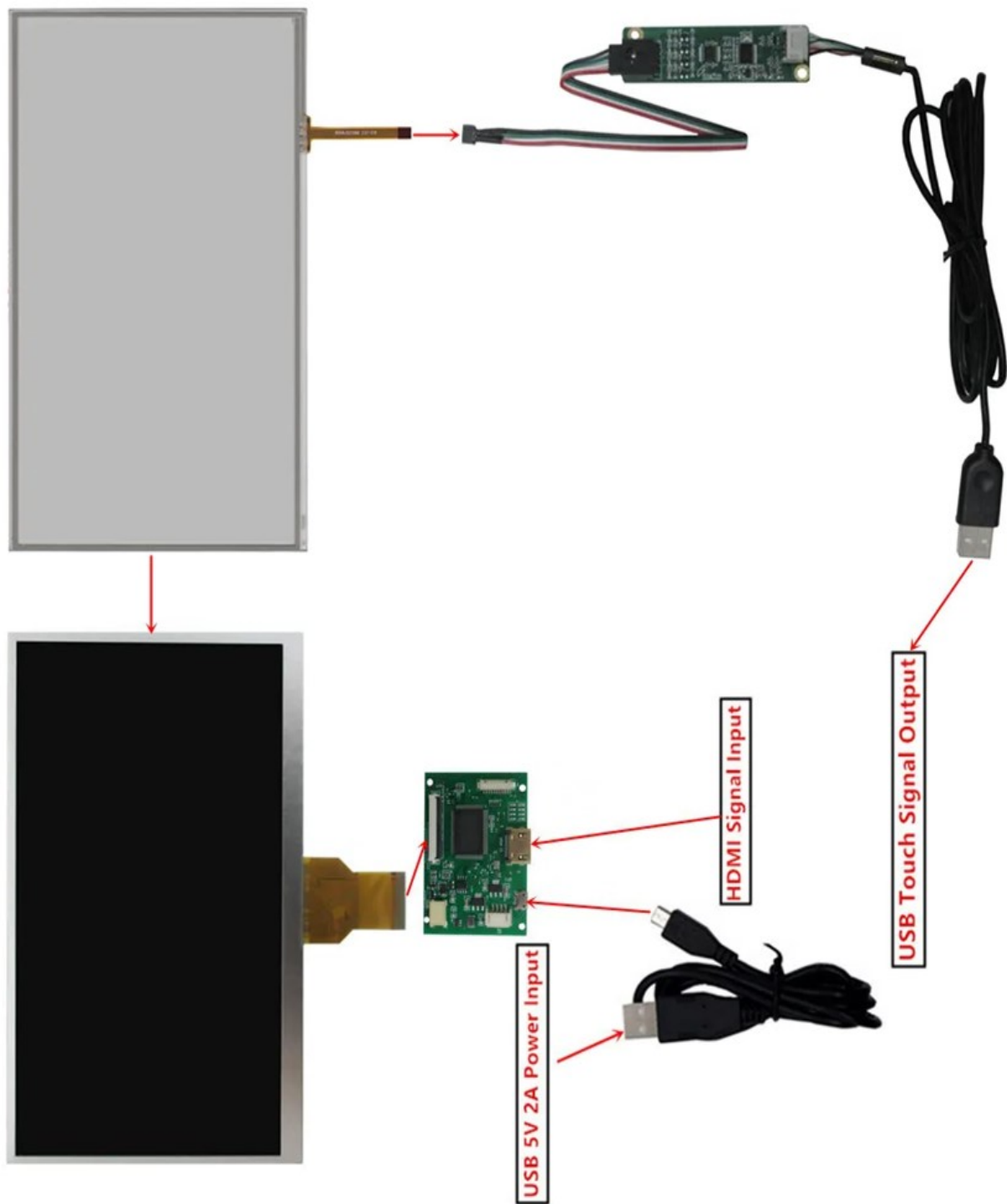


Рисунок 2.6 – Схема під'єднання плати Rasp-
berry Pi 3 Uno до монітору

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.124.000.000 ПЗ

Арк.

33

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Реалізація програмного забезпечення

Головний клас програми є MainActivity.java, він забезпечує користувацький інтерфейс та функціональність вибору текстових елементів та надсилання їх на сервер. Клас підключає пакети всіх необхідних класів і пакетів (див. рис. 3.1).

```
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
```

Рисунок 3.1 – Лістинг коду підключення пакетів класів і бібліотек

Головний метод даного класу є onCreate (див.рис.3.2), який викликається при створенні активності.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
```

Рисунок 3.2 – Лістинг сигнатури методу і початкових дій

Він також виконує ініціалізацію користувацького інтерфейсу (див.рис. 3.3) та встановлює макет для активності. Також даний клас встановлює XML-макет

Змн.	Арк.	№ докум.	Підпис	Дата	КС КРБ 123.124.000.000 ПЗ			
Розроб.		Олійник М.Р.			Реалізація та тестування системи	Літ.	Арк.	Акрушів
Перевір.		Осухівська Г.М.					34	12
Реценз.		Дмитроца Л.П.				ТНТУ, каф. ФІС, гр. СІ-41		
Н. контр.		Тиш Є.В.						
Затверд.		Осухівська Г.М.						

активності, визначений у файлі конфігурації.

```
texts = new ArrayList<>();
    texts.add("Підйом");
    texts.add("Застелити ліжко");
    texts.add("Почистити зуби");
    texts.add("Сніданок");
    texts.add("Одягнутися");
    texts.add("Йти до школи");
    texts.add("Обід");
    texts.add("Гратися");
    texts.add("Робити домашнє завдання");
    texts.add("Вечеря");
    texts.add("Готуватись до сну");
    texts.add("Читати казку");
    texts.add("Спати");
    selectedOrder = new ArrayList<>();
```

Рисунок 3.3 – Лістинг XML-макету активності

У наступному фрагменті коду (див.рис. 3.4) створюється новий потік для виконання мережових операцій. Встановлюється URL-адреса сервера та налаштовується HTTP-з'єднання.

```
private void sendOrderToServer() { try {
    URL url = new URL("http://localhost:8080/updateOrder");
    HttpURLConnection connection = (HttpURLConnection)
url.openConnection();
    connection.setRequestMethod("POST");
    connection.setRequestProperty("ContentType""application/json");
```

Рисунок 3.4 – Лістинг коду відправки даних на сервер

					КС КРБ 123.124.000.000 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

Після написання програми додатку для керування комп'ютеризованої системи потрібно написати програму, яка буде запускатись і виконуватись безпосередньо на платі Raspberry Pi Uno 3 B. Написаний код буде у текстовому редакторі Nano.

Клас ImageController (див.рис. 3.5) відповідає за обробку HTTPS-запиту, та управління порядком зображень для відображення на екрані. Він містить REST-контролер, який приймає список чисел (номера зображень) та оновлює порядок відображення зображень відповідно до цього списку.

```
@RestController
public class ImageController {
    private List<Integer> imageOrder;
    @PostMapping("/updateOrder")
    public void updateOrder(@RequestBody List<Integer>
newOrder) {
        this.imageOrder = newOrder;
    }
    public List<Integer> getImageOrder() {
        return imageOrder;
    }
}
```

Рисунок 3.5 – Лістинг класу ImageController

Клас ImageDisplayFrame (див.рис. 3.6) відповідає за створення графічного інтерфейсу користувача, який відображає зображення відповідного до отриманого списку.

					КС КРБ 123.124.000.000 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public class ImageDisplayFrame extends JFrame {

    private JLabel leftImageLabel;

    private JLabel centerImageLabel;

    private JLabel rightImageLabel;

    private List<String> imageNames;

    public ImageDisplayFrame(List<String> imageNames) {
        this.imageNames = imageNames;
        initializeUI();
    }
}

```

Рисунок 3.6 – Лістинг класу ImageDisplayFrame і його полів

Також необхідний метод що буде настраювати параметри вікна, такі як заголовок, розмір та макет (див.рис.3.7).

```

private void initializeUI() {

    setTitle("Daily Routine");

    setExtendedState(JFrame.MAXIMIZED_BOTH);

    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    setLayout(new BorderLayout());
}

```

Рисунок 3.7 – Лістинг методу настройки інтерфейсу вікна

Наступним кроком буде налаштовано об'єкти для відображення зображень (див.рис. 3.8), що будуть додані до макету вікна.

					КС КРБ 123.124.000.000 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

```

leftImageLabel = new JLabel();

centerImageLabel = new JLabel();

rightImageLabel = new JLabel();

add(leftImageLabel, BorderLayout.WEST);

add(centerImageLabel, BorderLayout.CENTER);

add(rightImageLabel, BorderLayout.EAST);

```

Рисунок 3.8 – Лістинг коду створення об'єктів для відображень зображень

Наступний фрагмент коду на рис. 3.9 додає обробник подій для відображення вікна. Коли вікно відобразиться, викликається метод `updateImages`, який оновлює зображення.

```

addComponentListener(new ComponentAdapter() {
    @Override
    public void componentShown(ComponentEvent e) {
        updateImages();
    }
});

Timer timer = new Timer(60000, new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        updateImages();
    }
});

timer.start();

```

Рисунок 3.9 – Лістинг коду обробника подій для відображення вікна

Насамкінець має бути добавлений код, для обробки натискання клавіш дитиною (див. рис. 3.10).

```

addKeyListener(new KeyAdapter() {
    @Override
    public void keyPressed(KeyEvent e) {

        if (e.getKeyCode() == KeyEvent.VK_ENTER) {

centerImageLabel.setBorder(BorderFactory.createLineBorder(Color.GREEN, 5));

        if (e.getKeyCode() == KeyEvent.VK_ESCAPE) {
            System.exit(0);
        }
    }
});

```

Рисунок 3.10 – Лістинг коду обробника натискання клавіш

Після цього напишемо код, котрий відповідатиме за встановлення порядку зображень. Метод, на рис.3.11, приймає колекцію List<Integer>, кожне ціле число даного списку це ідентифікатор зображення

```

public void setImageOrder(List<Integer> order) {
    if (order.size() != imageNames.size()) {
        throw new IllegalArgumentException("Order list must have the same length as imageNames list");
    }
    List<String> newImageOrder = new ArrayList<>();
    for (Integer i : order) {
        if (i < 1 || i > imageNames.size()) {
            throw new IllegalArgumentException("Order values must be between 1 and " + imageNames.size());
        }
        newImageOrder.add(imageNames.get(i - 1));
    }
    imageNames = newImageOrder;
}

```

Рисунок 3.11 – Лістинг методу встановлення порядку зображень

Після закінчення розробки програмної частини комп'ютеризованої системи, можна розробити і її апаратну частину.

					КС КРБ 123.124.000.000 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2 Апаратна частина

Підключимо плату Raspberry Pi 3 В до джерела живлення. Можемо бачити результат підключення на рис. 3.11.



Рисунок 3.11 – Плата Raspberry Pi Uno 3 В підключена до живлення.

Наступним кроком буде підключення клавіатури, на рис. 3.12, для можливості написання коду безпосередньо на платі.

					КС КРБ 123.124.000.000 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.12 – Плата Raspberry Pi 3 Uno B підключена до клавіатури

І насамкінець підключимо безпосередньо сам монітор, на рис.3.13.

					КС КРБ 123.124.000.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.13 – Підключення монітора Raspberry Pi

На даному етапі програмна і апаратна частини комп'ютеризованої системи розкладу дня для дитини з аутизмом закінчені.

3.3 Тестування

Тестування комп'ютеризованої системи буде відбуватись через її запуск, і відправлення з програмного додатку відповідних запитів на плату Raspberry Pi 3 Uno B, після чого спостерігатимемо результат на екрані.

Припустимо, що на момент тестування година – 07:00 ранку.

					КС КРБ 123.124.000.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

На мобільному додатку встановимо розклад, що на 07:00 має відбуватись подія – сніданок, а на годину 07:10 – навчання в школі, і відправимо запит на сервер. Результат можемо спостерігати на рис. 3.14.

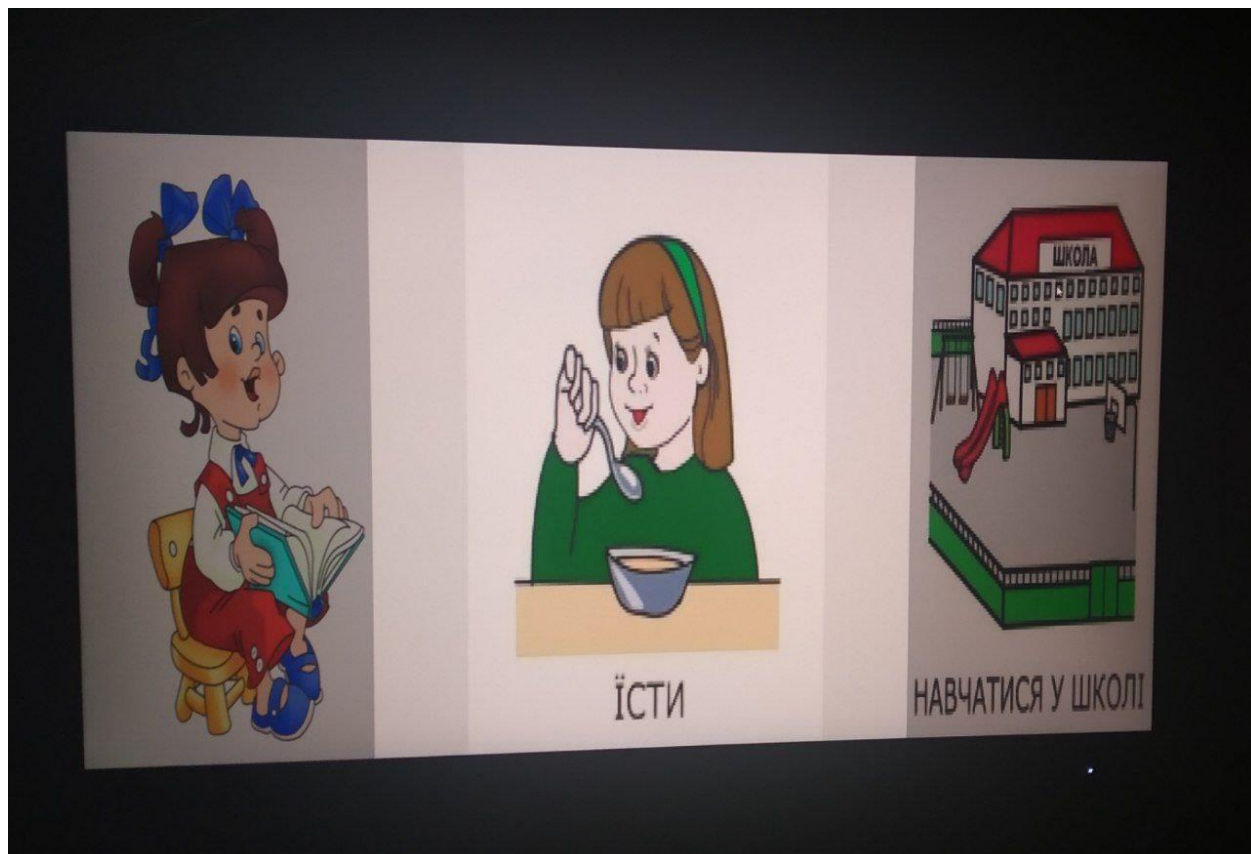


Рисунок 3.14 – Відображення встановленого ранкового розкладу на моніторі

Тепер перевіримо роботу ще одної функції, для позитивного заохочення дитини. Після натискання клавіші Enter зображення встановленої на цю подію зображення засвітиться зеленим кольором, підказуючи дитині про виконане завдання, на рис. 3.15.

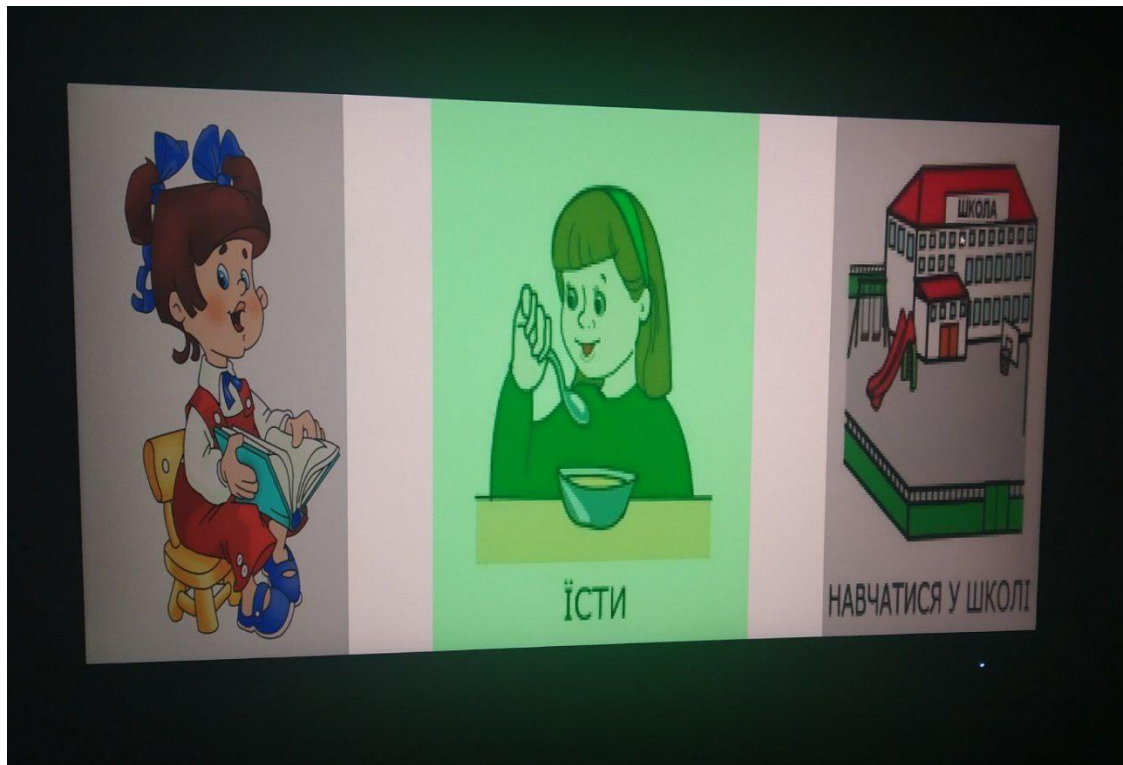


Рисунок 3.15 – Приклад зображення події, виконаною дитиною.

Припустимо, що на момент тестування година – 12:30. Із мобільного додатку встановимо подію на цю годину – дозвілля. Наступною подією встановимо – виконання домашнього завдання, а попередньою встановимо подію обіду. Результат можемо спостерігати на рис. 3.16.

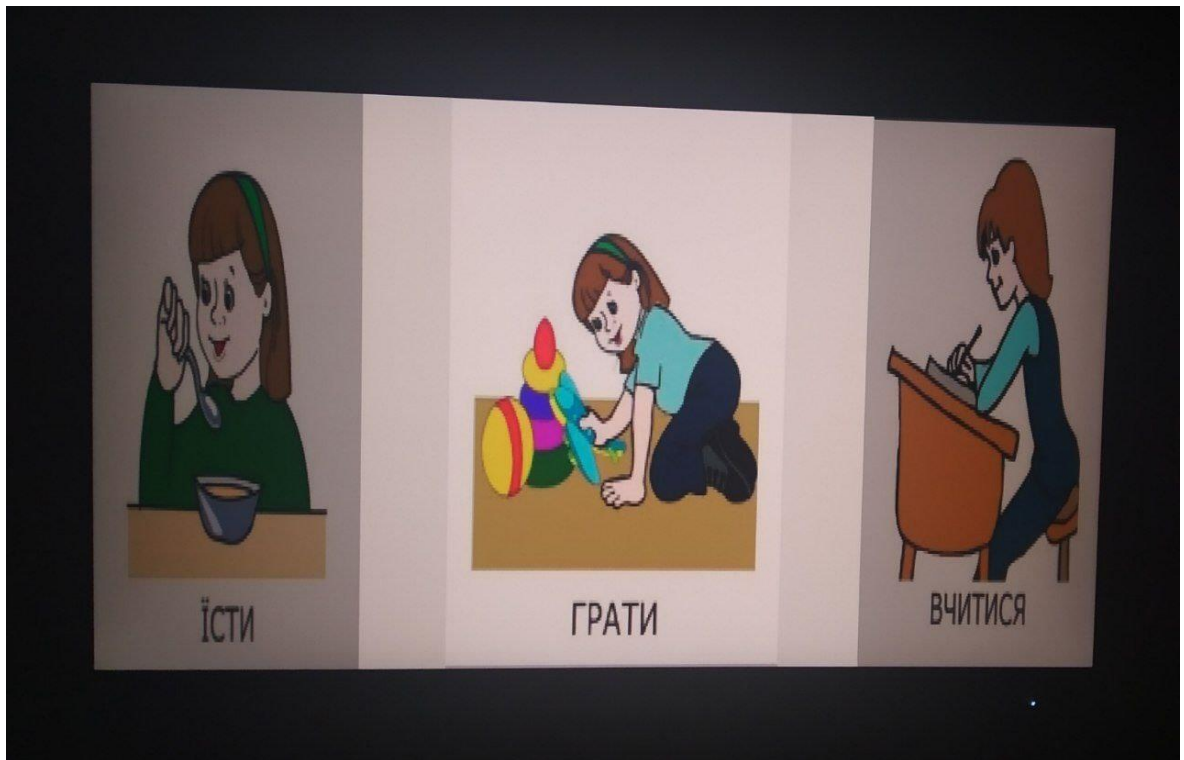


Рисунок 3.16 – Відображення встановленого обіднього розкладу на моніторі

Таким чином, в рамках даного розділу було завершена програмна і апаратна частина комп'ютеризованої системи.

					КС КРБ 123.124.000.000 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

Безпека життєдіяльності та охорона праці є важливими аспектами будь-якого проекту, пов'язаного з розробкою та впровадженням комп'ютеризованих систем.

4.1 Долікарська допомога при отруєннях

Долікарська допомога при отруєннях є ключовим елементом у забезпеченні безпеки та здоров'я постраждалих. Отруєння можуть бути викликані різними речовинами, включаючи хімікати, лікарські засоби, продукти харчування, рослини та гази. У контексті комп'ютеризованої системи розкладу дня для дитини з аутизмом, необхідно розробити алгоритми для надання першої допомоги при отруєннях.

При виявленні отруєння працівник зобов'язаний негайно визначити тип і джерело отрути. У випадку, якщо отруєння сталося через споживання їжі або рідини, важливо зберегти залишки цієї їжі або рідини для подальшого аналізу медичними фахівцями. При підозрі на вдихання токсичних газів або парів працівник зобов'язаний негайно забезпечити доступ до свіжого повітря, вивівши постраждалого з зараженої зони.

Працівник зобов'язаний оцінити стан постраждалого, перевіряючи його свідомість, дихання та пульс. Якщо постраждалий у свідомості і може ковтати, слід дати йому випити невелику кількість води. У разі виявлення хімічного отруєння не рекомендується викликати блювоту без консультації з медичними фахівцями, оскільки деякі речовини можуть викликати додаткові ушкодження при повторному проходженні через стравохід.

					КС КРБ 123.124.000.000 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушіє</i>
<i>Розроб.</i>		Олійник М.Р.					46	4
<i>Перевір.</i>		Осухівська Г.М.						
<i>Консульт.</i>		Пилипець М.І.						
<i>Н. контр.</i>		Тиш Є.В.						
<i>Затверд.</i>		Осухівська Г.М.						
						<i>ТНТУ, каф. ФІС, гр. СІ-41</i>		

У випадках отруєння через шкіру або очі працівник зобов'язаний негайно промити уражені ділянки великою кількістю води. Для очей це слід робити протягом не менше 15 хвилин, уникаючи сильного натиску.

Якщо одяг забруднений отруйними речовинами, працівник зобов'язаний обережно зняти його, щоб уникнути подальшого контакту зі шкірою.

Комп'ютеризована система розкладу дня для дитини з аутизмом включає візуальні та текстові інструкції для дій у випадку отруєння. Це допомагає дитині та її оточенню швидко і правильно реагувати на ситуацію. Розробка таких систем потребує співпраці з фахівцями з медицини та дитячої психології для адаптації алгоритмів до специфічних потреб дитини з аутизмом.

До прибуття медиків слід зберігати спокій і контролювати стан постраждалого, регулярно перевіряючи його свідомість, дихання та пульс. Якщо стан постраждалого погіршується, необхідно вжити заходів для підтримки життєво важливих функцій, включаючи серцево-легеневу реанімацію, якщо це необхідно.

Комп'ютеризована система повинна включати контакти екстрених служб та надавати інструкції щодо їх виклику. У випадку отруєння важливо уникати самостійного призначення будь-яких ліків або домашніх засобів без консультації з лікарем, оскільки це може погіршити стан постраждалого.

Регулярне навчання для дитини та її батьків або опікунів є необхідним. Це включає практичні заняття з надання першої допомоги при отруєннях, використання симуляцій та інтерактивних модулів у системі. Навчання має бути орієнтоване на формування навичок, які можна легко застосувати у разі реальної небезпеки.

Розробка таких систем повинна базуватися на передових дослідженнях у галузі медицини і психології. Інтеграція з існуючими медичними протоколами та рекомендаціями, зокрема з огляду на специфічні потреби дітей з аутизмом, забезпечить комплексну і ефективну підтримку як у повсякденному житті, так і в екстрених ситуаціях.

					КС КРБ 123.124.000.000 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

Інтеграція алгоритмів долікарської допомоги при отруєннях у комп'ютеризовану систему розкладу дня сприяє підвищенню безпеки і готовності дитини до надзвичайних ситуацій.

4.2 Заходи щодо захисту установки від короткого замикання

Захист установки від короткого замикання призводить до серйозних пошкоджень обладнання, пожеж та загроз життю людей. Мною розроблено наступний комплекс заходів, для попередження інцидентів.

Працівник зобов'язаний застосовувати автоматичні вимикачі, що миттєво розривають електричний ланцюг при виявленні надмірного струму. Автоматичний вимикач нормально реагує на аномальні умови роботи і відключає живлення до того, як пошкодження можуть поширитися на інші частини системи. Автоматичні вимикачі налаштовані відповідно до характеристик системи.

Працівник повинен використовувати запобіжники, які забезпечують захист від короткого замикання шляхом плавлення під впливом надмірного струму. Запобіжники прості в експлуатації та обслуговуванні, їх заміна працівником не вимагає значних витрат часу і коштів. Працівник повинен встановлювати запобіжники з правильними номінальними характеристиками, щоб забезпечити оптимальний рівень захисту.

Обслуговуючий персонал пристрою повинен застосовувати реле захисту для виявлення аномалій в комп'ютеризованій системі розкладу дня для дитини з аутизмом. Реле реагує на зміни струму, напруги або частоти, і ініціюють відключення комп'ютеризованої системи при виявленні небезпечних умов. Інтеграція реле захисту в комп'ютеризовану систему розкладу дня, забезпечує високу точність і швидкість реакції на коротке замикання.

Працівник зобов'язаний регулярно перевіряти та обслуговувати електричне обладнання комп'ютеризованої системи. Включити у програму обслуговування періодичні інспекції так тестування автоматичних вимикачів,

					КС КРБ 123.124.000.000 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

запобіжників та реле захисту. Працівник повинен виявляти та усувати потенційних проблем на ранніх стадіях, що дозволить запобігти серйозним інцидентам і знизити ризик короткого замикання

Перед початком роботи із комп'ютеризованою системою, працівник зобов'язується забезпечити належне заземлення електричних установок. Ефективне заземлення зменшує ризик ураження електричним струмом та забезпечує безпеку роботи електричної плати комп'ютеризованої системи для розкладу дня.

Обов'язок обслуговуючого персоналу – використовувати кабелі з ізоляцією високою якості, що знижує ймовірність короткого замикання через пошкодження ізоляційного матеріалу. Працівник повинен обирати кабелі з відповідними номінальними характеристиками, що відповідають умовам експлуатації. При монтажі забезпечити належну укладку та захист кабелів від механічних ушкоджень.

Розробник комп'ютеризованої системи зобов'язаний впровадити програму навчання персоналу щодо захисту від короткого замикання. Проводити регулярні тренінги та симуляції, що підвищують рівень обізнаності до дій, у надзвичайних ситуаціях. Освічений персонал здатний ефективно реагувати на загрози і мінімізувати ризики.

Комплексний підхід до захисту установок від короткого замикання, що включає використання автоматичних вимикачів, запобіжників, реле захисту, регулярне обслуговування обладнання, належне заземлення, високоякісні кабелі, систему моніторингу та навчання персоналу, забезпечує високу ступінь безпеки електричних схем комп'ютеризованої системи розкладу дня для дитини з аутизмом.

					КС КРБ 123.124.000.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У дипломній роботі було розроблено комп'ютеризовану систему керування розкладу дня для дитини з аутизмом, на основі плати Raspberry Pi 3 Uno. Основною метою системи є реалізація структури дня, забезпечуючи візуальну і звукову підтримку відповідно до рекомендацій поведінкових неврологів.

Розробка КСРДДА включала ідентифікацію та аналіз вимог, вибір відповідних апаратних та програмних компонентів, а також їх інтеграцію та тестування. Використання плати Raspberry Pi 3 Uno у прив'язці до Raspberry Pi Display забезпечило компактність системи, що дозволяє розміщувати її у будь-якому зручному місці будинку. База даних PostgreSQL дозволила зберігати та обробляти дані про розклади дня.

КСРДДА полегшує виконання щоденних рутинних завдань і покращуючи якість життя дитини, і розширюючи батьківський контроль.

					КС КРБ 123.124.000.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Палат О.Т, Хаєр К. В., Методики оцінки розвитку дитини з аутизмом. Стокгольм, SU. 2018. 29 с. (Дата звернення 24.05.2024).
2. Калан Т. В., Основи діагностики дітей з розладами аутистичного спектру. Київ, КНУ. 2015. 126 с. (Дата звернення 24.05.2024).
3. Стартовий сет графічних таблиць для дітей з аутизмом URL: <https://autismlittlelearners.com/free-visual-supports/> (Дата звернення 24.05.2024).
4. Опис плати Raspberry Pi 3 Uno URL: <https://raspberrypi.com.ua/p/raspberrypi-3ode-1-b-armv8-with-1g-ram/> (Дата звернення 24.05.2024).
5. Опис монітор Raspberry Pi Display URL: <https://www.raspberrypi.com/products/raspberry-pi-touch-display/> (Дата звернення 24.05.2024).
6. Рекомендації поведінкових неврологів URL: <https://www.mayinstitute.org/autism-schools/index.html> (Дата звернення 29.05.2024).
7. Бібліотека управління дисплеями по SPI з DMA. Версія 1.4 (<https://github.com/vadrov/stm32-display-spi-dma>) (Дата звернення 29.05.2024).
8. Осухівська Г. М., Тиш Є. В., Луцик Н. С., Паламар А. М. Методичні вказівки до виконання кваліфікаційних робіт здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль, ТНТУ. 2022. 28 с. (Дата звернення 29.05.2024).
9. Микитишин А. Г., Митник М. М., Стухляк П. Д. Телекомунікаційні системи та мережі. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2017. 384 с. (Дата звернення 29.05.2024).
10. Осухівська Г.М., Шаблій Н.Р. Методичні вказівки для виконання лабораторних робіт з дисципліни «Технологія проектування комп'ютерних систем». Тернопіль: ТНТУ імені Івана Пулюя, 2017. 150 с. (Дата звернення 29.05.2024).
11. Коротке замикання: що це та які причини. URL: <https://vsee.com/ua/novosti/korotkoe-zamykanie-hto-eto-i-kakie-prichiny/>. (Дата звернення 29.05.2024).

					КС КРБ 123.124.000.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

12. Проект роботи файлового менеджера з використанням бібліотеки FATFS (<https://github.com/vadrov/stm32-filemanager-sd-fatfs-display-buttons>). (Дата звернення 29.05.2024).
13. How autism can affect you and your family URL: <https://www.nhs.uk/conditions/autism/autism-and-everyday-life/help-for-families/> (Дата звернення 24.05.2024).
14. How to help your autistic child with day-to-day life URL: <https://www.nhs.uk/conditions/autism/autism-and-everyday-life/help-for-day-to-day-life/> (Дата звернення 29.05.2024).
15. Parenting a Child on the Autism Spectrum URL: <https://www.webmd.com/brain/autism/parenting-child-with-autism> (Дата звернення 24.05.2024).
16. Acceptance and commitment therapy URL: <https://raisingchildren.net.au/autism/ther-apies-guide/acceptance-commitment-therapy>
17. Raspberry Pi Schematics URL: <https://www.flux.ai/sitemap/schematics/raspberrypi-?page=1> (Дата звернення 29.05.2024).
18. Schematics diagram of wiring connection to the Raspberry Pi. URL: https://www.re-searchgate.net/figure/Schematics-diagram-of-wiring-connection-to-the-RaspberryPi_fi-g1_363327637 (Дата звернення 29.05.2024).
19. Organizations and resources for families with an autistic child URL: <https://www.w-ellchild.org.uk/get-support/information-hub/organisations-andresources-for-families-with-an-autistic-child/> (Дата звернення 29.05.2024).
20. Parenting an autistic child URL: <https://www.gingerbread.org.uk/findinformation/p-arenting-perspectives/parenting-autistic-child/> (Дата звернення 29.05.2024).
21. Події розкладу дня дитини із аутизмом URL: <https://www.istockphoto.com/uk/%D1%96%D0%BB%D1%8E%D1%81%D1%82%D1%80%D0%B0%D1%86%D1%96%D1%97/autistic-child-and-parent> (Дата звернення 29.05.2024).
22. Routine for child with Autism URL: <https://spectrumofhope.com/blog/create-routine-for-kids-with-autism/> (Дата звернення 29.05.2024).

					КС КРБ 123.124.000.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

Додаток А
Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя
Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

«Затверджую»

завідувач кафедри КС

_____Осухівська Г.М.

" ____ " _____ 2024 р.

КОМП'ЮТЕРИЗОВАНА СИСТЕМА РОЗКЛАДУ ДНЯ ДЛЯ ДИТИНИ З АУТИЗМОМ

ТЕХНІЧНЕ ЗАВДАННЯ

на 4 листках

Вид робіт: Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

Керівник кваліфікаційної роботи

_____доцент Осухівська Г.М.

« ____ » _____ 2024 р.

«ВИКОНАВЕЦЬ»

Студент групи СІ-41

_____Олійник М.Р.

« ____ » _____ 2024 р.

Тернопіль 2024

1. Повна назва та її умовне позначення.

Повна назва теми кваліфікаційної роботи «Комп'ютеризована система розкладу дня для дитини з аутизмом».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.124.00.00

2. Виконавець

Студент групи СІ-41 факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Олійник Максим Романович.

3. Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№4/7-408 від 24.04.2024 р.)

4. Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 24.04.2024 р. Плановий термін завершення виконання кваліфікаційної роботи –

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ІСО, ГОСТ, ЕСКД, ЕСПД та ДСТУ.

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи. Попередній захист кваліфікаційної роботи відбувається при готовності

роботи на 90% , наявності пояснювальної записки та графічного матеріалу.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

5. Призначення і цілі створення системи

Система що розробляється призначена для розкладу і контролю розпорядку дня дітей із розладами аутистичного спектру.

6. Мета створення системи

Метою кваліфікаційної роботи є розробка система що встановлює події і задачі на відповідну годину, і повідомляє батьків про виконання. Батьки, мають мати можливість встановлювати розклад.

7. До складу виробу повинні входити:

а)Плата Raspberry Pi 3;

б)Монітор Raspberry Pi Display.

8. Конструктивні вимоги

Конструювання корпусу приладу комп'ютеризованої системи розкладу дня для дитини з аутизмом – не передбачено.

Для побудови системи використана сучасна компонентна база.

9. Техніко-економічні показники

Планова собівартість проєкту повинна становити не більше 5000 тисяч гривень.

10. Стадії та етапи проєктування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної плати

№ етапу	Назва етапів роботи	Термін виконання етапів роботи
1	Затвердження технічного завдання	01.02-03.02.24
2	Аналіз технічного завдання	05.02-11.02.24
3	Розробка структурної та функціональної схеми	25.04-04.05.24
4	Вибір елементної бази	05.06-18.05.24
5	Вибір програмного забезпечення	03.06-09.06.24
6	Реалізація проектної складової	07.06-9.06.24
7	Опрацювання розділу «Безпека життєдіяльності, основи охорони праці»	10.06.24
8	Оформлення пояснювальної записки кваліфікаційної роботи	12.06.24
9	Оформлення графічної частини	15.06.24
10	Підготовка до захисту	20.06.24
11	Захист кваліфікаційної роботи	28.06.24

11. Під час виконання кваліфікаційної роботи у технічне завдання можуть вноситись зміни та доповнення.

Додаток Б. Лістинг програми

```
package org.example;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ComponentAdapter;
import java.awt.event.ComponentEvent;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import java.time.LocalDateTime;
import java.util.Arrays;
import java.util.List;
import javax.imageio.ImageIO;
import java.util.ArrayList;

public class Main extends JFrame {
    private JLabel leftImageLabel;
    private JLabel centerImageLabel;
    private JLabel rightImageLabel;
    private boolean highlightCenterImage = false;

    private List<String> imageNames = Arrays.asList(
        "src/main/java/org/example/images/7_rise.jpg",
        "src/main/java/org/example/images/710_bed.jpg",
        "src/main/java/org/example/images/730_teeth.png",
        "src/main/java/org/example/images/740_breakfast.jpg",
        "src/main/java/org/example/images/800_clothes.jpg",
        "src/main/java/org/example/images/830_school.jpg",
        "src/main/java/org/example/images/1210_lunch.jpg",
        "src/main/java/org/example/images/1230_fun.jpg",
        "src/main/java/org/example/images/1530_homework.jpg",
        "src/main/java/org/example/images/1700_dinner.jpg",
        "src/main/java/org/example/images/1900_prepare_for_bed.jpg",
        "src/main/java/org/example/images/2000_story.jpg",
        "src/main/java/org/example/images/7_rise.jpg",
        "src/main/java/org/example/images/740_breakfast.jpg",
        "src/main/java/org/example/images/830_school.jpg",
        "src/main/java/org/example/images/2100_sleep.jpg"
    );
};
```

```

private List<LocalTime[]> timeIntervals = Arrays.asList(
    new LocalTime[]{LocalTime.of(7, 0), LocalTime.of(7, 10)},
    new LocalTime[]{LocalTime.of(7, 10), LocalTime.of(7, 30)},
    new LocalTime[]{LocalTime.of(7, 30), LocalTime.of(7, 40)},
    new LocalTime[]{LocalTime.of(7, 40), LocalTime.of(8, 0)},
    new LocalTime[]{LocalTime.of(8, 0), LocalTime.of(8, 30)},
    new LocalTime[]{LocalTime.of(8, 30), LocalTime.of(12, 10)},
    new LocalTime[]{LocalTime.of(12, 10), LocalTime.of(12, 30)},
    new LocalTime[]{LocalTime.of(12, 30), LocalTime.of(15, 30)},
    new LocalTime[]{LocalTime.of(15, 30), LocalTime.of(17, 0)},
    new LocalTime[]{LocalTime.of(17, 0), LocalTime.of(19, 0)},
    new LocalTime[]{LocalTime.of(19, 0), LocalTime.of(20, 0)},
    new LocalTime[]{LocalTime.of(20, 0), LocalTime.of(21, 0)},
    new LocalTime[]{LocalTime.of(21, 0), LocalTime.of(7, 0)}
);

public Main() {
    setTitle("Daily Routine");
    setExtendedState(JFrame.MAXIMIZED_BOTH);
    setUndecorated(true);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    setLayout(new BorderLayout());

    leftImageLabel = new JLabel();
    leftImageLabel.setHorizontalAlignment(JLabel.CENTER);
    leftImageLabel.setVerticalAlignment(JLabel.CENTER);
    centerImageLabel = new JLabel();
    centerImageLabel.setHorizontalAlignment(JLabel.CENTER);
    centerImageLabel.setVerticalAlignment(JLabel.CENTER);
    rightImageLabel = new JLabel();
    rightImageLabel.setHorizontalAlignment(JLabel.CENTER);
    rightImageLabel.setVerticalAlignment(JLabel.CENTER);

    add(leftImageLabel, BorderLayout.WEST);
    add(centerImageLabel, BorderLayout.CENTER);
    add(rightImageLabel, BorderLayout.EAST);

    // Add ComponentListener to call updateImage once the window is shown
    addComponentListener(new ComponentAdapter() {
        @Override
        public void componentShown(ComponentEvent e) {
            updateImages();
        }
    });
};

```

```

Timer timer = new Timer(60000, new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        updateImages();
        highlightCenterImage = false; // Reset highlight when image changes
    }
});
timer.start();

// Add KeyListener to handle the Esc and Enter keys
addKeyListener(new KeyAdapter() {
    @Override
    public void keyPressed(KeyEvent e) {
        if (e.getKeyCode() == KeyEvent.VK_ESCAPE) {
            System.exit(0);
        } else if (e.getKeyCode() == KeyEvent.VK_ENTER) {
            highlightCenterImage = true;
            updateImages();
        }
    }
});

// Request focus to ensure key events are captured
setFocusable(true);
}

private void updateImages() {
    int index = getCurrentTimeIndex();
    int prevIndex = (index - 1 + imageNames.size()) % imageNames.size();
    int nextIndex = (index + 1) % imageNames.size();

    updateImageLabel(leftImageLabel, imageNames.get(prevIndex), true);
    updateImageLabel(centerImageLabel, imageNames.get(index), false, highlightCenterImage);
    updateImageLabel(rightImageLabel, imageNames.get(nextIndex), true);
}

private void updateImageLabel(JLabel label, String imageName, boolean dimmed) {
    updateImageLabel(label, imageName, dimmed, false);
}

private void updateImageLabel(JLabel label, String imageName, boolean dimmed, boolean
highlight) {
    try {
        BufferedImage img = ImageIO.read(new File(imageName));
        int width = dimmed ? getWidth() / 4 : getWidth() / 3;
        int height = getHeight();

```

```

Image scaledImg = img.getScaledInstance(width, height, Image.SCALE_SMOOTH);
if (dimmed) {
    scaledImg = createDimmedImage(scaledImg);
}
if (highlight) {
    scaledImg = createHighlightedImage(scaledImg);
}
label.setIcon(new ImageIcon(scaledImg));
} catch (IOException e) {
    e.printStackTrace();
}
}

private Image createDimmedImage(Image src) {
    BufferedImage dimmedImage = new BufferedImage(src.getWidth(null), src.getHeight(null),
BufferedImage.TYPE_INT_ARGB);
    Graphics2D g2d = dimmedImage.createGraphics();
    g2d.drawImage(src, 0, 0, null);
    g2d.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC_ATOP, 0.3f)); //
Increased dimming
    g2d.setColor(Color.BLACK);
    g2d.fillRect(0, 0, src.getWidth(null), src.getHeight(null));
    g2d.dispose();
    return dimmedImage;
}

private Image createHighlightedImage(Image src) {
    BufferedImage highlightedImage = new BufferedImage(src.getWidth(null), src.getHeight(null),
BufferedImage.TYPE_INT_ARGB);
    Graphics2D g2d = highlightedImage.createGraphics();
    g2d.drawImage(src, 0, 0, null);
    g2d.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC_ATOP, 0.3f));
    g2d.setColor(Color.GREEN);
    g2d.fillRect(0, 0, src.getWidth(null), src.getHeight(null));
    g2d.dispose();
    return highlightedImage;
}

private int getCurrentTimeIndex() {
    LocalTime now = LocalTime.now();
    for (int i = 0; i < timeIntervals.size(); i++) {
        LocalTime[] interval = timeIntervals.get(i);
        if (now.isAfter(interval[0]) && now.isBefore(interval[1])) {
            return i;
        }
    }
}

```

```

        return timeIntervals.size() - 1; // Default to last image if no match found
    }

    // New method to set image order
    public void setImageOrder(List<Integer> order) {
        if (order.size() != imageNames.size()) {
            throw new IllegalArgumentException("Order list must have the same length as imageNames
list");
        }
        List<String> newImageOrder = new ArrayList<>();
        for (Integer i : order) {
            if (i < 1 || i > imageNames.size()) {
                throw new IllegalArgumentException("Order values must be between 1 and " +
imageNames.size());
            }
            newImageOrder.add(imageNames.get(i - 1));
        }
        imageNames = newImageOrder;
        updateImages();
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                Main main = new Main();
                main.setVisible(true);

                // Example of setting image order
                main.setImageOrder(Arrays.asList(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13));
            }
        });
    }
}

package org.example.androidapp;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;

```



```

public class TextAdapter extends RecyclerView.Adapter<TextAdapter.TextViewHolder> {
    private ArrayList<String> texts;
    private ArrayList<Integer> selectedOrder;

    public TextAdapter(ArrayList<String> texts, ArrayList<Integer> selectedOrder) {
        this.texts = texts;
        this.selectedOrder = selectedOrder;
    }

    @NonNull
    @Override
    public TextViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_text, parent,
false);
        return new TextViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull TextViewHolder holder, int position) {
        String text = texts.get(position);
        holder.textView.setText(text);
        holder.textView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                selectedOrder.add(position + 1); // Adding 1 to match the numbering
                holder.textView.setEnabled(false); // Disable the item after selection
            }
        });
    }

    @Override
    public int getItemCount() {
        return texts.size();
    }

    public static class TextViewHolder extends RecyclerView.ViewHolder {
        public TextView textView;

        public TextViewHolder(@NonNull View itemView) {
            super(itemView);
            textView = itemView.findViewById(R.id.textView);
        }
    }
}

```