

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра Інженерія програмного забезпечення

Пояснювальна записка

до бакалаврської роботи

на ступінь вищої освіти бакалавр

на тему: **«Розробка програмного забезпечення для реабілітації при афазії
засобами C#, WPF. Спецчастина: Розробка програмного модуля для
відпрацювання вправ письмової мови»**

Виконав: студент 4 курсу, групи ПД-44 спеціальності

121 Інженерії програмного забезпечення

(шифр і назва спеціальності)

Міщук А.Ю.

(прізвище та ініціали)

Керівник Золотухіна О.А.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Київ – 2022

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення
Ступінь вищої освіти - “Бакалавр”
Спеціальність - 121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри
інженерії програмного забезпечення

Негоденко О.В.

“ ” _____ 2022 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Мішука Андрія Юрійовича

(прізвище, ім'я, по батькові)

Тема роботи: «Розробка програмного забезпечення для реабілітації при афазії засобами C#, WPF. Спецчастина: Розробка програмного модуля для відпрацювання вправ письмової мови»

Керівник роботи Золотухіна Оксана Анатоліївна, к.т.н., доц.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затвержені наказом вищого навчального закладу від “18” лютого 2022 року

Строк подання студентом роботи 03.06.2022

Вихідні дані до роботи:

Microsoft Visual Studio;

WPF;

Figma;

Офіційна документація мови програмування C#

Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

Аналіз існуючих рішень для відновлювання письмової мови;

Аналіз використаних технологій;

Опис проектування системи:

Опис використаних технологій:

Перелік графічного матеріалу

1. Титульний слайд
2. Мета, об'єкт дослідження, предмет дослідження
3. Аналіз існуючих аналогів
4. Технічне завдання
5. Засоби програмної реалізації
6. Діаграма діяльності
7. Діаграма класів
8. Розроблений додаток
9. Висновки

Дата видачі завдання 11.04.2022

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи до	Примітка
1	Вибір теми бакалаврської роботи	21.04	виконано
2	Дослідження актуальності теми	22.04	виконано
3	Розгляд аналогів	23.04	виконано
4	Проектування логіки проекту	26.04-10.05	виконано
5	Проектування та реалізація інтерфейсу	14.05-21.05	виконано
6	Реалізація функцій проекту	21.05-31.05	виконано
7	Оформлення пояснювальної записки та демонстраційного матеріалу	03.06.2022	виконано
8	Здача роботи в деканат	06.06.2022	виконано

Студент _____ Міщук А.Ю.

(підпис) (прізвище та ініціали)

Керівник роботи _____ Золотухіна О.А.

(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи с. 76, рис. 41, табл. 2, джерел 18, додатків 1.

Об'єкт дослідження – виконання вправ по відновленню письмової мови.

Предмет дослідження – програмне забезпечення для виконання вправ по відновленню письмової мови.

Мета роботи – ІТ-підтримка процесу реабілітації пацієнтів з афазією шляхом відпрацювання вправ письмовою мовою за допомогою програмного забезпечення.

Під час досліджень та розробки додатку, був проведений аналіз предметної галузі. Після цього були встановленні основні переваги і недоліки додатків-аналогів.

У роботі було проведено аналіз існуючих додатків, які можна використовувати в процесі реабілітації при афазії для відновлення порушень письмової мови.

Виконано проектування додатку: розроблено діаграму варіантів використання, діаграму класів, створено макети форм інтерфейсу.

Мовою C# було розроблено додаток, який дозволяє відпрацювати шість видів вправ письмової мови, а також реалізована статистика для перегляду прогресу реабілітації пацієнта.

АФАЗІЯ, ВПРАВИ ПИСЬМОВОЇ МОВИ, C#, .NET, WPF

ЗМІСТ

УМОВНІ ПОЗНАЧЕННЯ	8
ВСТУП	9
1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	10
1.1. Особливості порушення писемного мовлення при афазії	10
1.2. Методи і вправи для відновлення письмової мови	12
1.3. Огляд інструментів для відновлення порушень при афазії	16
2. ОПИС ПРОЕКТУВАННЯ СИСТЕМИ	28
2.1. Модель для розробки ПЗ	28
2.2. iOS	34
2.3. Android	36
2.4. Microsoft Windows	38
2.5. Вибір мови програмування для розробки ПЗ	39
2.6. Платформа .NET	40
2.7. WPF – основні компоненти та інтерфейс середовища	42
2.8. Вибір програми для створення дизайну для додатку	43
3. ПРОЕКТУВАННЯ ТА РОЗРОБКА ДОДАТКУ	47
3.1. Діаграма варіантів використання	47
3.2. Опис математичної моделі вправи «Провести лінію»	49
3.3. Діаграма класів	52
3.4. Опис функціонування додатку	56
ВИСНОВКИ	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	68
ДОДАТОК А ДЕМООНСТРАЦІЙНІ МАТЕРІАЛИ	70

УМОВНІ ПОЗНАЧЕННЯ

ДП - домашня практика

ПЗ – програмне забезпечення

РАС – це стан, який виникає внаслідок порушення розвитку головного мозку.

ВСТУП

Проблема збереження здоров'я людей є однією із ключових задач держави. Порушення письмової мови, які є частиною афазії, спостерігаються у дітей та дорослих з розладами аутистичного спектру (РАС), як наслідок захворювання на інсульт, після черепно-мозкових травм та інших захворювань. Відновлення працездатності хворого або послаблення наслідків хвороби повинно відбуватися за мінімальний час та з максимальним ефектом. На практиці швидке відновлення потребує значних коштів як на рівні закладів лікування так і власних коштів, часу та зусиль хворого та сім'ї. Успіх значною мірою залежить від самостійної роботи хворого та допомоги оточуючих – членів сім'ї, знайомих, друзів.

Зважаючи на складність процесів реабілітації, як допоміжний інструмент розумним є використання програмних засобів, які дозволяють хворим чи людям з їх оточення організувати самостійну роботу з відновлення письмових навичок та вмінь, що пов'язані з письмом (пам'ять, когнітивні здібності тощо).

Більшість додатків, які призначені для реабілітації порушень при афазії, в тому числі письмової мови, платні. Окрім того, інтерфейс додатків не призначено для людей, які навіть не можуть прочитати опис завдання, тож використання додатку постійно потребує помічника, який пояснював би вголос суть завдань. В більшості випадків у користувачів відсутня можливість коригувати завдання, що може викликати у хворих незадоволення надлишковою простотою чи, навпаки, складністю завдань, а це значною мірою знижує ефект від використання додатку та зменшує імовірність успішного лікування.

Таким чином, є нагальна потреба в розробці програмного забезпечення для українського ринку, яке б мало необхідний функціонал з різними рівнями складності реабілітації для кожного пацієнта та україномовний інтерфейс, максимально адаптований під інклюзивні потреби користувача. Використання зазначеного програмного забезпечення дозволить зробити процес реабілітації більш комфортним для хворих та їх сімей та в багатьох випадках скоротити час

і вартість реабілітації, а також постійну потребу у спеціалізованих фахівцях-логопеда.

1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1. Особливості порушення писемного мовлення при афазії

Мова є засобом спілкування, та мовлення — процесом спілкування. При деяких захворюваннях, наприклад, порушенні мозкового кровообігу (інсульту) часто спостерігаються порушення мовлення. В залежності від локалізації інсульту уражаються ті чи інші функції організму, згідно зі статистичними даними, близько 25% пацієнтів що перенесли інсульт, страждають тим чи іншим ступенем порушення мовлення. Відновлення порушення мовлення при інсульті відбувається набагато повільніше, ніж при рухових розладах.

Афазія – це системне порушення мови, що виникає при органічних ураженнях мозку, охоплює різні рівні організації мови, впливає на її зв'язки з іншими психічними процесами та призводить до дезінтеграції всієї психічної сфери людини, порушуючи насамперед комунікативну функцію мови. Всі форми афазії істотно відрізняються одна від одної за основним механізмом, за клінічною картиною та психологічною структурою порушення мови. Спільним всім форм афазії є системне порушення промови.

Основна, найчастіша причина розладу мови у дорослих - судинні порушення в головному мозку - геморагічні та ішемічні інсульти. Афазія при інсульті проявляється приблизно у 15-35% постраждалих.

Кожна форма афазії виникає при ураженні певної ділянки кори лівої півкулі мозку, так званої мовної зони, і залежить від локалізації вогнища ураження і фактора. В одних випадках афазія протікає як грубе порушення усного – експресивного та імпресивного мовлення, в інших – як порушення імпресивного мовлення через дефекти розуміння логіко-граматичних конструкцій мови. У багатьох випадках афазія супроводжується аграфією та олексією, тобто порушеннями письма та читання. При афазії порушуються практично всі три види промови – усне (експресивне та імпресивне), письмове

(читання та письмо) та внутрішнє мовлення. Всі ці види мови та їх різні форми порушуються по-різному при різних формах афазії та ступені тяжкості говоріння, розуміння мови, вміння писати і читати так само різна: від грубої міри порушення до легкої. При афазії грубо порушується предметна віднесеність слова: хворі що неспроможні за словом знайти предмет, зобразити його, зрозуміти значення слова. Стійке порушення здатності говорити призводить до інвалідизації: людина втрачає можливість виконувати багато соціальних функцій і гостро потребує реабілітації.

Лікування афазії в обов'язковому порядку передбачає вплив на першопричинні фактори, для чого залежно від діагнозу може знадобитися допомога невролога або нейрохірурга, консервативне або хірургічне лікування. У деяких випадках афазія проходить самостійно, коли відновлюється нормальний кровотік у мозкових тканинах, але такий сприятливий результат зустрічається у невеликого числа постраждалих. Найчастіше корекція афазії вимагає безліч зусиль з боку медиків і самого пацієнта, тривалого періоду реабілітації.

Письмова мова, зокрема один з видів порушення — це складний психічний процес, що має свій психологічний зміст, непросту структуру і характеризується особливими шляхами його формування, а також складною взаємодією з мозком. На сьогоднішній день відомо, що процес письма, порушуючись внаслідок ураження мозку, має різні форми, які залежать від того, яка ланка у структурі письма порушена, що, у свою чергу, пов'язане з локалізацією вогнища ураження мозку. Порушення письма у дорослих найчастіше виникає внаслідок ушкодження вищих психічних функцій. Причини насамперед пов'язані з травмами голови, операцій на головному мозку, важкими інфекціями, інсультами, а також кисневим голодуванням клітин головного мозку. Порушення може розвинутися на тлі когнітивних розладів, пам'яті, зниженої працездатності, тимчасової та просторової дезорієнтації, важких соціальних умов. Почерк стає нерівним, людина, навіть добре володіючи

правилами мови, неспроможна їх застосувати, плутається у написанні букв, неспроможна розташувати слова у потрібному порядку, пише дуже повільно.

Для правильного написання слова необхідно усвідомлювати послідовність звуків у слові, а не лише вміти аналізувати звукообробний склад цього слова. Саме дотримання послідовності звуків при записі представляє одну з найбільших труднощів, що виявляються при ураженні задньолобних відділів кори лівої півкулі мозку. В основі цих дефектів лежить порушення механізму перемикання з однієї літери на іншу.

Порушення процесу перемикання є центральним дефектом. У цілому нині картина порушення проявляється у ряді помилок в написанні, персеверацій (лише на рівні складу, слова та речення) до його грубого розпаду. Внаслідок розпаду усвідомлення хворим внутрішньої схеми слова і послідовності букв відзначаються перестановки букв у слові, пропуски букв, що позначають голосні за її збігу, повторення однієї й тієї ж мови (слова), недописування слів (порушення сукцесивного процесу).

Порушення письма ускладнюються дефектами розуміння мови, зниженням розуміння значення слів та його сенсу. Порушення семантики мовлення відбивається на стані писемного мовлення і є одним з видів труднощів при відновленні писемного мовлення. У письмі характерне порушення внутрішньої схеми слова та речення, усвідомлення послідовності букв у слові (слів у реченні). У хворих порушується усвідомлення складних (динамічних) відносин слів усередині речення, завдяки яким не слово, а речення стає одиницею значення та сенсу. Цим хворим важко створити «в умі» схему цілої граматичної конструкції, розмістити та зв'язати окремі слова у речення. У грубих випадках у хворих відсутні всі види письмової мови, крім ідеограмного. Навіть при списуванні вони допускають безліч помилок, промовляючи у своїй кожен звук, склад.

1.2. Методи і вправи для відновлення письмової мови

Відновлення мови - найважливіше завдання реабілітації. Реабілітація мови повинна починатися, як тільки дозволить загальний стан хворого і стан його свідомості. Через велику втому пацієнтів заняття на перших етапах проводяться по 10-15 хвилин кілька разів на день.

Вправа на мовні та комунікативні навички. Використовується вправа, розроблена для покращення розуміння прочитаного, навичок письма, а також поповнення або відновлення словникового запасу різного рівня складності. Практикуються заходи, які підтримують повсякденне життя.

Паралельно з мовним спілкуванням і заняттями з відновлення усної мови пацієнтів необхідно привчати читати і писати, адже відновлення цих функцій значною мірою сприятиме відновленню мови. Якщо немає надії на відновлення письма правою рукою, пацієнта слід навчити писати лівою. Краще почати вчитися писати, ніж друкувати на друкарській машинці/комп'ютері, адже краще почати вчитися писати, ніж друкувати. Письмо словами тісно пов'язане з функцією мови. Надалі при задовільному відновленні навичок письма словами можна приступати до навчання типу на друкарській машинці / комп'ютері, особливо в тих випадках, коли пацієнт вже володів ними до хвороби. Краще почати читати з друкованих слів, намагатися писати письмові завдання для пацієнта спочатку блоковими літерами і тільки в подальшому переходити на великі літери, написані розбірливо і чітко. З перших днів, ще до повного відновлення функції читання, пацієнту слід дозволити переглядати газети і журнали, які він читав до хвороби, адже у пацієнтів навіть з порушеннями функцій аналітичного читання часто зберігається можливість вгадування читання.

Лікування порушень писемного мовлення завжди комплексне і спрямоване насамперед на нормалізацію процесів аналізу та синтезу письмових елементів – від окремих літер до цілих речень. Кожна вправа корекційної програми має бути націлена на зміцнення зв'язків між мозковими структурами та формуванням нових, правильних навичок:

- постановка проблемних звуків;

- розширення словникового запасу;
- освоєння граматичного устрою;
- поліпшення зв'язності мовлення, аналітичних здібностей, сприйняття інформації на слух.

Для досягнення стійкого позитивного результату, крім регулярних занять із логопедом, хворим необхідно виконувати самостійно спеціальні вправи.

Центральним завданням відновлення писемної мови при афазії є відновлення аналітико-синтетичного письма: усвідомлення послідовності звуків і літер у слові та її роль значення слова. Хворому пропонуються пари-трійки предметних картинок, найменування яких відрізняється лише порядком літер у слові. Під кожним малюнком лежить складене з букв слово:

- хворий повинен написати слова з пам'яті, прочитати пари заданих слів, співвіднести їх із картинкою, показати різницю у написанні, списати ці слова парами;

- хворому пропонується переставити задані літери місцями те щоб вийшло інше за значенням слово, пояснити різницю у написанні слів, списати їх, написати з пам'яті.

На другому етапі переходимо до відновлення аналітичного методу, спрямованого на усвідомлений аналіз складу слова. Робота над написанням слова складається з кількох операцій:

- хворий повинен промовити слово за складами;
- кількісний аналіз слова (підрахувати та записати кількість звуків у слові за допомогою зовнішніх опор – паличок, фішок, або позначаються у зошиті різними ручками у вигляді пунктир або точок);
- складання схеми слова на основі проведеного аналізу (літерна та послогова схема);
- Заповнення схеми відповідними літерами самостійно.

На наступному етапі аналітичного методу застосовується ряд вправ, вкладених у закріплення аналітичного навички написання слів. В даний вид вправ входить:

- розбір слова за складами та буквами, спочатку з опорою на картинку та написане слово (у тому числі і слів, близьких за фонетичною структурою та серійною організацією), потім на слух; - дописування незакінчених слів, слів на заданий склад;

- Вписування пропущених літер у словах;
- Складання слів з даних врозбивку складів.

Третій етап навчання письма слів спрямовано стимулювання хворого до аналізу послідовності букв у слові, усвідомлення її зв'язку зі значенням слова. Хворим даються різні завдання, створені задля складання слів з даних врозкид букв, складання слів з урахуванням кореневої частини слова, анаграми. Другою стадією навчання є відновлення написання фрази. Проводиться робота над формуванням уміння складати речення певних граматичних конструкцій.

Розглянемо спосіб складання картинних фраз. Перед хворим викладається ряд картинок із зображенням суб'єктів, об'єктів (предметів) та дій (дії зображуються стрілкою). Пропонується скласти з картинок одну пропозицію, розкласти слова під відповідними картинками, вибрати потрібне закінчення слів, вимовити отриману фразу і записати її в зошит. Далі пропонується вигадати аналогічну фразу. Наступний етап роботи полягає у складанні фраз з використанням відповідних питань до слів фрази, що сприяє відновленню «почуття мови», структури фрази (хто? що? що робить? що? (чим? кого? кому?). Хворому пропонується розкласти картинки відповідно до схемою, а потім заповнити схему словами з потрібними відмінковими закінченнями. Пацієнту пропонується самостійно поставити запитання до кожної картинки та відповідного запису складеної таким чином пропозиції. Для усунення персеверацій під час письма використовують метод повільного промовляння, метод «паузування», тобто. збільшення пауз між написанням кожного слова у фразі. Для тренування перемикання уваги під час паузи можна запропонувати виконати якусь короткочасну операцію. Поступово паузи замінюємо відповідними словами картинками чи картинками, що позначають дієслова. Цей метод дозволяє подолати персеверації при написанні як фраз, так й текстів.

Для тренування моторики використовують вправу Лабіринт, коли хворому пропонується подолати бездоріжну лінію. При цьому хворий повинен рухати тільки рукою, не дозволяйте йому змінювати положення написання. Також ефективно знаходження предметів і букв в сюжетних картинках, малювання і штрихування графічних диктантів.

Для розвитку уваги і при оптико-просторних руйнуваннях рекомендується виконувати завдання на конструювання букв з елементів, перетворень; на розшифровки схем і символів, якими позначили літери, наприклад, 2-п, 3-т. Можна також шукати задані літери в предметах, вставляти пропущені літери в слова, пропозиції і тексти. Запам'ятати глядовий образ боків допомагає вправи, при яких хворий повинен викреслювати, підкреслювати або ознайомитись з роботою.

1.3. Огляд інструментів для відновлення порушень при афазії

Fast ForWord.

Fast ForWord (рис.1.1) - комп'ютерна корекційна методика, розроблена співробітниками компанії Scientific Learning - провідними вченими в галузі корекції порушень мовлення та аутизму.



Рисунок 1.1 – Fast ForWord (головне меню)

Назва програми перекладається як «Бігом за словом» або «Швидкословом». Ця програма тренує систему слухового сприйняття, тобто створює фундамент для формування усного та писемного мовлення.

Як це відбувається? У багатьох країнах діти, що починають говорити різними мовами, у своєму белькоті використовують типові поєднання голосних і приголосних звуків на кшталт «да-да-да» та «ба-ба-ба». Перші слова, які діти вимовляють, теж складаються з таких поєднань: «тато», «мама», «баба». Для дитини з мовними проблемами розрізнення цих звуків надовго залишається великою проблемою. Мозок таких дітей не може обробляти типові поєднання приголосних та голосних, які швидко вимовляються. Діти не встигають почути початок складу або те, як звуки змінюються всередині мови.

Ця нездатність подолати елементарне завдання порушує всі сторони мовного розвитку дитини. Він витрачає багато сил та часу на спроби розрізнити звуки та зрозуміти, яке саме слово прозвучало. Це стає причиною порушення розуміння мови, вимовної сторони мови, читання та письма.

У Sky Gym (Рис 1.2) користувачі слухають частотно-модульовані звуки, які змінюють висоту звуку від низького до високого (піднімання вгору або «Weer») або від високого до низького (стрибок вниз або «Woor»). Багато звуків мови (фонем) включають розгортки. Наприклад, щоб відрізнити /b/ від /d/, потрібно
вміти розрізнити та розпізнавати компоненти розгортки цих звуків мовлення.



Рисунок 1.2 – Fast ForWord (завдання Sky GYM)

Sky Gym починається з одиночних звуків, потім переходить до парних. Учні повинні натиснути кнопку «Перейти», щоб почути розгортку або пару розгорток. Якщо вони почують один раз, вони повинні натиснути кнопку стрілка вгору або вниз, щоб відповідати цьому звуку. Якщо вони почують пару замахів, вони повинні натиснути кнопку вгору та стрілки вниз у правильному порядку, щоб відповідати шаблону звуків.

Коли пацієнт натискає кнопку «Go», Ноор Nut(Рис 1.3) представляє цільовий склад, наприклад /ba/. Спочатку жолудь зліва рухається і грає склад, після нього жолудь праворуч рухається і грає склад. Один з жолудів грає цільовий склад, а інший — альтернативний склад для поточної пари фонем. Пацієнт повинен натиснути на жолудь, який зіграв цільовий склад.



Рисунок 1.3 - Fast ForWord (завдання Hoop Nut)

Вправа Ele-Bot (Рис 1.4) починається з представлення Еллі, електричного робота.

1. Натискання кнопки «Go» активує робота.
2. На екрані з'являється від двох до чотирьох картинок.
3. Заслуховується твердження чи запитання.
4. Пацієнт натискає на зображення, яке найкраще відображає твердження або відповідає на запитання.
5. Подається коригувальний відгук.
6. набір зображень зникає, створюючи основу для наступного запитання.



Рисунок 1.4 – Fast ForWord (завдання ELE-BOT)

Проблема цієї програми - інтерфейс, що орієнтований в більшості на дітей, може викликати незадоволення та роздратування у дорослих хворих. Крім того, програма орієнтована на англомовних користувачів та особливості звучання звуків англійської мови. Поставляється вона також лише англійською мовою - перекладу на жодну іншу мову світу немає.

iName it.

iName - це програма для одного гравця, спеціально розроблена для використання логопедом або опікуном, наприклад, членом сім'ї. На головному екрані користувач має можливість вибрати «почати практику», «про», «результати» або «підтримка». Вибравши «почати практику», користувач може вибрати або створити аватар клієнта.

Після створення профілю (Рис 1.5) клієнта екран автоматично заповнює вибір із п'яти візуальних сцен. У додатках є п'ять налаштувань. Кожен параметр включає десять словникових слів, а кожне словникове слово включає п'ять рівнів підказки.

Дотик до екрана дозволяє вибрати одну з п'яти сцен(Рис.1.6).

У цій вправі (Рис 1.7) панель відображає елементи, які можна знайти на сцені. Коли ви вже на місці події, просто натисніть на один із десяти пунктів, запропонованих на iName, щоб почати практику. Вибраний елемент є єдиним, який залишається кольоровим, тоді як решта сцени затінена.



Рисунок 1.5 – iName it (меню створення профілю)



Рисунок 1.6 – iName it (п'ять візуальних сцен)



Рисунок 1.7 – iName it (приклад завдання)

В цілому програма проста та легка у використанні, має збір даних, мені також подобається реалістична графіка яка орієнтована на дорослих та дітей. Є поле з 10 предметів домашнього вжитку в 5 наборах, загалом 50 предметів. Я хотів би побачити більше категорій і більше фотографій. Деякі предмети важко побачити людям із вадами зору. Було б корисно мати масштабування.

CommunicAide.

CommunicAide, розроблений двома досвідченими логопедами, спрямований на тисячі людей щодня, які живуть із довготривалими наслідками афазії та інших мовних розладів.

CommunicAide пропонує найбільший вибір справжніх фотографій і справжніх людських голосів практично для будь-якої ситуації, а також можливість імпортувати власні фотографії для справді настроюваного досвіду. Завдяки надзвичайно зручному інтерфейсу (Рис 1.8, 1.9) ця програма дає можливість людям, які живуть з мовними розладами, повідомляти про свої бажання та потреби іншим. Просто вибравши фотографію та/або фразу, користувачі можуть сформулювати повні речення та запитання, надаючи їм нове відчуття незалежності та свободи вираження. Обидва логопеди

використовували CommunicAide в клініці мовлення та слуху LIU/Post з чудовими результатами. Норе4Speech у поєднанні з Objectgraph сподіваються, що ця програма допоможе покращити якість життя своїх користувачів.



Рисунок 1.8 – CommunicAide (інтерфейс програми)



Рисунок 1.9 – CommunicAide (вибір завдання)

Особливості включають:

- гладкий і зручний інтерфейс для будь-якого віку;
- настроюване поле фотографій;
- можливість імпортувати особисті фотографії та фрази;
- можливість збереження улюблених фраз;
- прості у використанні сенсорні функції;
- справжні людські голоси для всіх фотографій і тексту (варіант на вибір - чоловічий або жіночий);
- справжні професійні фотографії;
- зображення тіла з вимірювачем болю.

Програма проста в навігації. Створювати нові елементи швидко та легко. Непотрібні (або небажані) предмети можна легко перемістити в бібліотеку, щоб створити менш відволікаюче поле для пацієнта. Пацієнт може побудувати речення, що складається з кількох елементів, або використовувати фотографії для спілкування одним словом залежно від рівня навичок пацієнта. Додаток зроблений тільки англійською мовою, інтерфейс застарілий, оновлень немає. Більшість завдань дуже легкі.

Lexico Cognition.

Додаток створений логопедами, вчителями та дизайнерами, цей додаток можна використовувати як у контексті розвитку, так і набутих мовних розладів. Простий, логічний інтерфейс користувача та привабливі зображення мотивують навіть молодших дітей працювати самостійно. Для логопедії після інсульту або травми голови Lexico Cognition пропонує новий підхід до розуміння понять і словникового запасу. Пацієнти (у супроводі чи поодиноці) можуть працювати з цікавим інструментом у реабілітації своїх мовних навичок. Додаток зосереджено на розумінні, нарощуванні словникового запасу, когнітивних навичках, пам'яті та слухових навичках.

На рис 1.10 можна побачити інтерфейс програми та вибір завдань на різний рівень складності.



Рисунок 1.10 – Lexico Cognition(вибір завдання)

В завданні «Який предмет краще підходить?» (Рис 1.11) пацієнту потрібно проаналізувати картинки та вибрати правильну відповідь. Після чого він може перевірити себе.

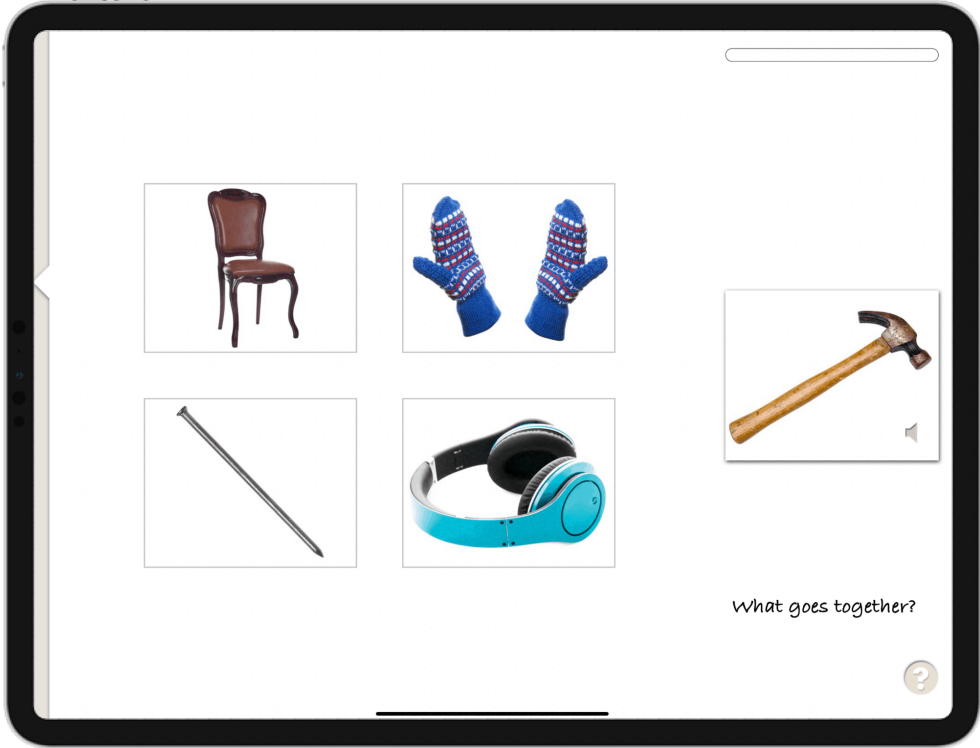


Рисунок 1.11 – Lexico Cognition (приклад завдання)

При виконанні вправи «Що він робить?» (Рис 1.12) користувач повинен вибрати правильну відповідь на питання яке йому поставить програма. За допомоги цієї вправи у людини розширюється словарний запас.

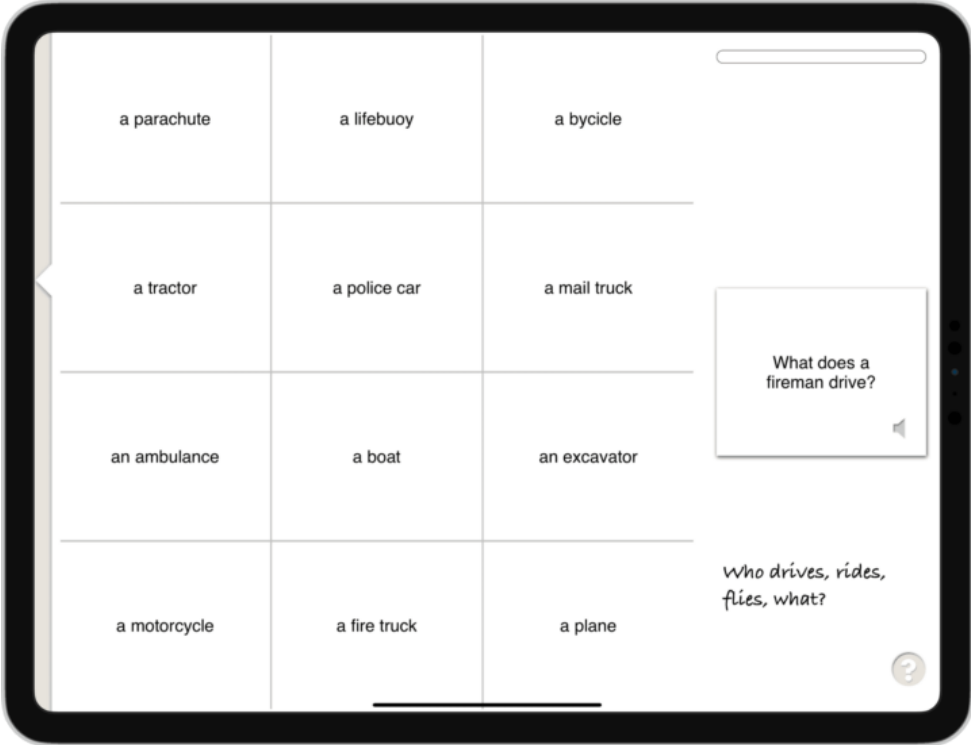


Рисунок 1.12 – Lexico Cognition (приклад завдання)

Не зважаючи на інклюзивний дизайн у програмі багато рівнів мають дуже низькоякісні малюнки, які важко інтерпретувати. І багато малюнків непридатні для використання з дітьми. Крім того, вибрані вами фотографії не мають культурного різноманіття.

В таблиці 1.1 наведено результати порівняльного аналізу додатків для реабілітації письмової мови.

Таблиця 1.1 – Порівняльний аналіз додатків для реабілітації письмової мови

Додатки	Переваги	Недоліки
iName It	Заповнення пропущених літер та складів у словах. Письмо найзвичніших слів - ідеограм, списування простих текстів. Є функція озвучки завдань.	Не вистачає озвучення тесту. Немає вправи на відпрацювання рисування ліній, що є елементами літер. Задання ніяк не прогресують.
Fast ForWord	Заповнення в текстах при письмі пропущених слів, що відпрацьовуються в мовленні. Розкладання підписів під картинками (предметними та сюжетними).	Не вистачає озвучення тесту. Маленький шрифт. Програма не підлаштовується під рівень пацієнта, комусь завдання будуть надто важкі, а комусь навпаки легкі.
Communicaid	Списування слів, фраз та невеликих текстів із	Немає вправи на відпрацювання рисування

	встановленням на ліній, що є елементами самоконтроль та самостійне літер. Важкий для виправлення помилок. Є функція озвучки завдань.	сприйняття інтерфейс.
Lexico Cognition	Писання під диктовку слів з звуковою структурою, що поступово ускладнюється, простих фраз, а також окремих складів і літер. Є функція озвучки завдань.	Немає вправи на відпрацювання рисування ліній, що є елементами літер. Завдання орієнтовані тільки на дитячу аудиторію.

1.4 Постановка завдань дипломної роботи

Об'єкт дослідження – виконання вправ по відновленню письмової мови.

Предмет дослідження – програмне забезпечення для виконання вправ по відновленню письмової мови.

Мета роботи – ІТ-підтримка процесу реабілітації пацієнтів з афазією шляхом відпрацювання вправ письмовою мови за допомогою програмного забезпечення.

Для досягнення поставленої мети необхідно виконати наступні задачі:

1. Провести аналіз предметної галузі для визначення вправ, які допоможуть пацієнтам відновити письмову мову.
2. Провести аналіз існуючих аналогів, які застосовуються для реабілітації письма та визначити переваги і недоліки цих додатків.
3. Спроекувати додаток, а саме розробити діаграму варіантів вигjhbcnfuyz, діаграму класів та створити макети форм інтерфейсу.
4. Розробити програмний додаток за допомогою C#, який дозволить відпрацьовувати шість різних вправ для відновлювання письма, створити інклюзивний дизайн, який враховує потреби користувача та реалізувати

можливість для лікаря або родича переглядати статистику для розуміння прогресу реабілітації пацієнта при афазії.

Програмне забезпечення повинно відповідати наступним вимогам:

1. Розробка вправ різної складності та типу для тренування письма українською мовою для пацієнтів при афазії:

- вправи, спрямовані на стабілізацію та розвиток мілкої моторики пальців та рук;
- вправи, спрямовані на тренування письмової пам'яті та логіки побудови друкованих слів.

2. Підтримка дизайну, орієнтованого на хворих на афазію – вікові особливості, психологічний стан, можливі фізіологічні проблеми з пальцями та руками.

3. Можливість для лікаря та родичів переглядати прогрес пацієнта за допомогою доступу до статистики виконання вправ.

2. ОПИС ПРОЕКТУВАННЯ СИСТЕМИ

2.1. Модель для розробки ПЗ

Для розробки додатку перш за все потрібно визначитись з якою саме моделлю буде створено проект.

Модель розробки програмного забезпечення — це структура, яка систематизує різні типи проектування дій та їх взаємодію. Вибір моделі залежить від масштабу і складності проекту.

Найвідомішими моделями розробки є водоспадна, ітеративна, спіральна та гнучкий. Ми розберемо кожен з цих моделей, щоб визначити найбільш ефективну для розробки ПЗ.

Модель водоспаду є прикладом послідовної моделі. У цій моделі діяльність з розробки програмного забезпечення поділяється на різні фази, і кожна фаза складається з ряду завдань і має різні цілі.

Це була перша модель, яка широко використовувалася в індустрії програмного забезпечення. Він поділяється на фази, і вихід однієї фази стає входом наступної фази. Обов'язковим є завершення етапу до початку наступного етапу. Коротше кажучи, у моделі Waterfall немає перекриття.

У водоспаді (Рис 2.1) розвиток однієї фази починається лише тоді, коли попередня фаза завершена. Через цю природу кожна фаза моделі водоспаду досить чітко визначена. Оскільки фази падають з вищого рівня на нижчий, як водоспад, це називається моделлю водоспаду.

Переваги використання моделі Водоспаду такі:

- Простий і легкий для розуміння та використання.
- Для невеликих проектів модель водоспаду працює добре і дає відповідні результати.
- Оскільки фази жорсткі та точні, одна фаза виконується по черзі, її легко підтримувати.
- Критерії в'їзду та виїзду чітко визначені, тому легко та систематично виконувати якість.

- Результати добре задокументовані.

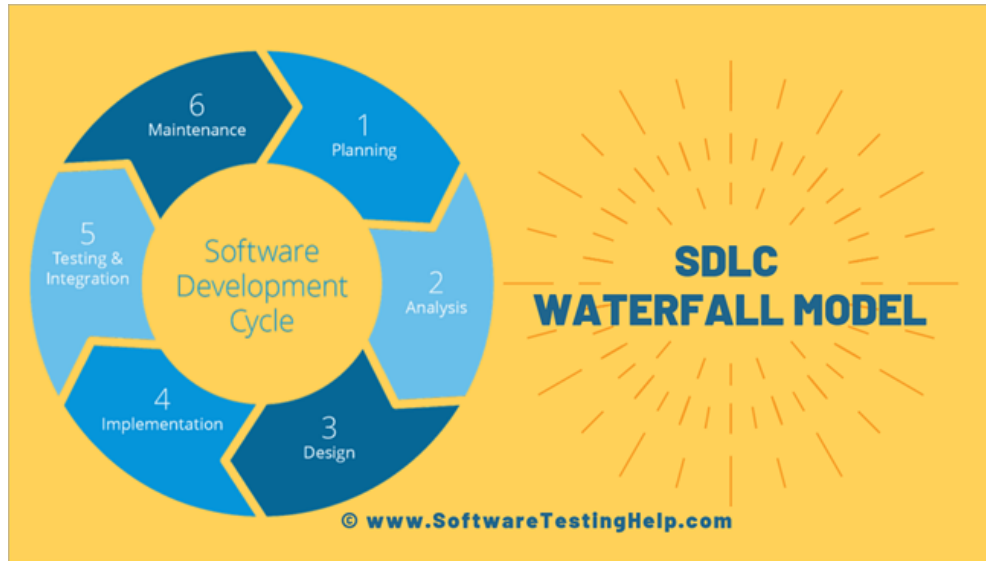


Рисунок 2.1 – Водоспадна модель

Недоліки використання моделі Водоспаду такі:

- Не вдається прийняти зміни у вимогах
- Повернутися до фази стає дуже важко. Наприклад, якщо додаток перейшов на стадію тестування і змінилися вимоги, стає важко повернутися і змінити його.
- Доставка кінцевого продукту затримується, оскільки немає прототипу, який був би продемонстрований відразу.
- Для великих та складних проектів ця модель не є хорошою, оскільки фактор ризику вищий.
- Не підходить для проектів, де вимоги часто змінюються.
- Не працює для тривалих та тривалих проектів.
- Оскільки тестування проводиться пізніше, це не дозволяє ідентифікувати виклики та ризики на попередньому етапі, тому стратегію зменшення ризику важко підготувати.

Ітеративна модель є однією з найпростіших у реалізації моделей життєвого циклу розробки програмного забезпечення. Існують певні сценарії, коли початкові або основні вимоги до програмного забезпечення чітко визначені, але фактичний діапазон чи повний набір функцій проекту невідомі.

Переваги використання Ітераційної моделі такі:

- Ця модель виробляє робоче програмне забезпечення дуже швидко.
- Ця модель дуже гнучка. Оскільки нові функції можуть бути додані до нього в будь-який момент розробки.
- Ця модель значно дешева, оскільки змінювати вимоги є менш витратними в порівнянні з іншими моделями процесу.
- Кінцевий користувач або зацікавлені сторони можуть швидко надати свої відгуки, які потім можна впровадити в систему.
- Помилки в системі можна виявити завчасно.

Недоліки використання Ітераційної моделі такі:

- Можуть виникнути проблеми, пов'язані з архітектурою системи, оскільки всі вимоги не зібрані заздалегідь.
- Це не найкращий вибір для невеликих проектів.
- Більш ресурсомістка, ніж модель водоспаду.
- Аналіз ризиків потребує висококваліфікованих спеціалістів для перевірки ризиків у нашій системі.
- Весь процес важко керувати.

Спіральна модель є однією з найважливіших моделей життєвого циклу розробки програмного забезпечення, яка забезпечує підтримку управління ризиками. У своєму схематичному зображенні він виглядає як спіраль з безліччю петель. Точна кількість петель спіралі невідома і може відрізнятись від проекту до проекту. Кожен цикл спіралі називається фазою процесу розробки програмного забезпечення. Точна кількість етапів, необхідних для розробки продукту, може бути змінена менеджером проекту залежно від ризиків проекту. Оскільки керівник проекту динамічно визначає кількість фаз, керівник проекту відіграє важливу роль у розробці продукту з використанням спіральної моделі.

Радіус спіралі в будь-якій точці представляє витрати (вартість) проекту на даний момент, а кутовий розмір представляє прогрес, досягнутий на даний момент на поточній фазі.

Переваги використання Спіральної моделі (Рис 2.3) такі:

- Управління ризиками: проекти з багатьма невідомими ризиками, які виникають під час розробки, у цьому випадку спіральна модель є найкращою моделлю розвитку через аналіз ризиків та управління ризиками на кожній фазі.

- Добре для великих проектів: Спіральну модель рекомендується використовувати у великих і складних проектах.

- Гнучкість у вимогах: запити на зміну вимог на пізнішому етапі можуть бути точно включені за допомогою цієї моделі.

- Задоволеність клієнта: Клієнт може побачити розробку продукту на ранній стадії розробки програмного забезпечення і, таким чином, він звик із системою, використовуючи її до завершення повного продукту.

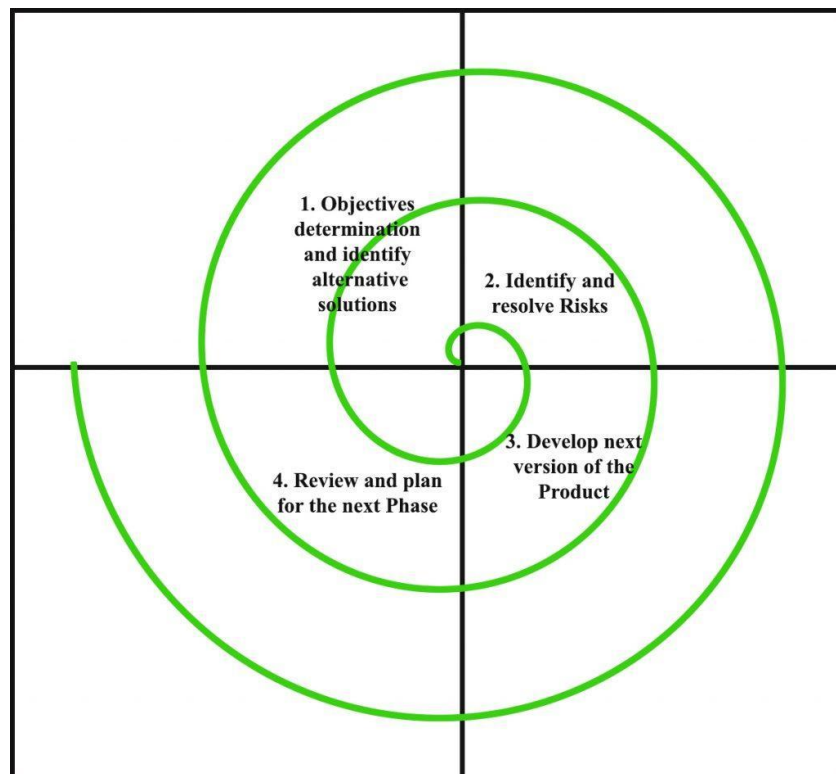


Рисунок 2.3 – Спіральна модель

Недоліки використання Спіральної моделі такі:

- Складна: Спіральна модель набагато складніша, ніж інші моделі SDLC.
- Дорого: Спіральна модель не підходить для невеликих проектів, оскільки вона дорога.

- Занадто велика залежність від аналізу ризиків: успішне завершення проекту дуже залежить від аналізу ризиків. Без дуже досвідчених експертів розробка проекту з використанням цієї моделі буде невдалою.

- Труднощі в управлінні часом: оскільки кількість етапів невідома на початку проекту, оцінка часу дуже складна.

Гнучка модель (Рис 2.4) відноситься до підходу до розробки програмного забезпечення, заснованого на ітераційній розробці. Гнучкі методи розбивають завдання на менші ітерації або частини безпосередньо не передбачають довгострокового планування. Обсяг проекту та вимоги визначаються на початку процесу розробки. Плани щодо кількості ітерацій, тривалості та обсягу кожної ітерації чітко визначені заздалегідь.

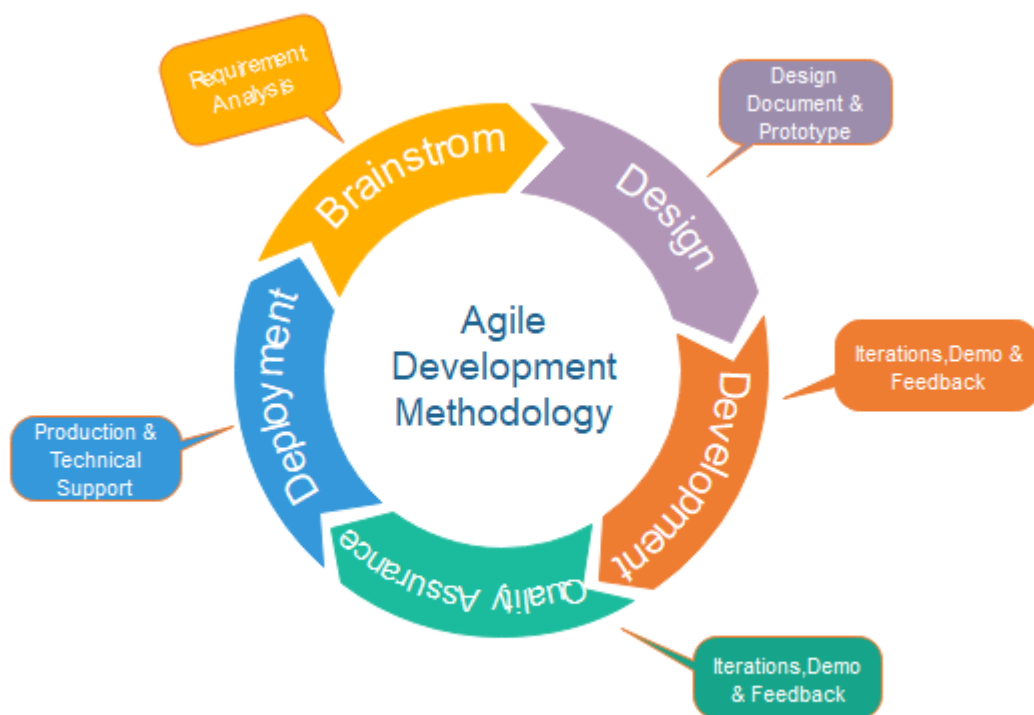


Fig. Agile Model

Рисунок 2.4 – Гнучка модель

Переваги використання Гнучкої моделі такі:

- Це дуже реалістичний підхід до розробки програмного забезпечення.
- Сприяє командній роботі та крос-тренуванням.

- Функціональність можна швидко розвинути та продемонструвати.
 - Вимоги до ресурсів мінімальні.
 - Підходить для фіксованих або змінних вимог.
 - Забезпечує ранні часткові робочі рішення.
 - Хороша модель для середовища, яке постійно змінюється.
 - Мінімальні правила, документація легко використовується.
 - Дозволяє паралельну розробку та доставку в рамках загального запланованого контексту.
 - Планування або зовсім не потрібно.
 - Легко керувати.
 - Надає розробникам гнучкість.
- Недоліки використання Спіральної моделі такі:
- Не підходить для складних проектів.
 - Більший ризик стійкості, ремонтпридатності та розширюваності.
 - Суворе управління доставкою диктує обсяг, функціональні можливості, які необхідно доставити, і коригування для дотримання термінів.
 - Дуже залежить від взаємодії з клієнтом, тому, якщо клієнт не зрозумілий, команду можна скерувати в неправильному напрямку.
 - Існує дуже висока індивідуальна залежність, оскільки створюється мінімум документації.
 - Передача технології новим членам команди може бути досить складною через відсутність документації.

Приймаючи до уваги усі переваги та недоліки описаних вище моделей, було прийнято рішення про використання гнучкої моделі розробки, оскільки створюване ПЗ невелике за масштабами і занадто велика документація лише сповільнить розробку, не надавши значних переваг.

2.2. iOS

Операційна система — це програмне забезпечення для вашого пристрою, яке поєднує апаратні та програмні ресурси. Це дозволяє різним типам

обладнання працювати разом, забезпечуючи платформу для різних програмних компонентів для взаємодії з цим обладнанням та іншими програмними компонентами.

IOS (рис. 2.5) мобільна операційна система для пристроїв Apple. iOS працює на iPhone, iPad, iPod Touch і Apple TV.

iOS відома тим, що є основним програмним забезпеченням, яке дозволяє користувачам iPhone взаємодіяти зі своїми телефонами за допомогою таких жестів, як прокрутка екрана, натискання та стиснення. Ці дії пальців зазвичай виконуються на сенсорних екранах, які забезпечують швидку відповідь і приймають введення від кількох пальців.



Рисунок 2.5 Логотип операційної системи iOS

Користувачі звертають увагу на доопрацювання додатків, доступних в App Store. Зазвичай вони значно менше заїкаються, і до того ж навряд чи несуть небезпеку несвідомого встановлення вірусів. Це пов'язано з тим, що додаток ретельно аналізується фахівцями перед тим, як потрапити в App Store. З цієї причини проблеми виникають набагато рідше, ніж у випадку з Android, який орієнтований на широку доступність і меншою мірою контролює ступінь захисту програм, доданих в Google Play.

З самого початку iOS мала бути простою у використанні системою, яка безперебійно працює на будь-якому пристрої. Розробники програмного забезпечення також подбали про безпеку та конфіденційність користувачів. На даний момент iOS є другою за популярністю операційною системою для мобільних пристроїв у світі — відразу після Android. Понад десяток років Apple працює над розширенням інтеграції програмного забезпечення для настільних комп'ютерів і мобільних пристроїв, доступного в її пропозиції, щоб забезпечити користувачам максимально можливий комфорт.

Пристрої зберігають історію переглядів, завдяки чому легко читати повідомлення як на смартфоні, так і на комп'ютері, а потім редагувати файли без затримки в часі. З цієї причини люди, які користуються iPhone дуже часто вирішують купити планшет або ноутбук від Apple. Це покращує функціональність і значно розширює можливості обох продуктів завдяки їх плавній взаємодії.

iOS має величезну перевагу в тому, що вона дуже проста та сумісна з кожним наступним оновленням. Насправді, користувачі iPhone вважають це найкращою особливістю платформи, оскільки її функції та програми продовжують працювати так само, як і повинні на нових пристроях. Це забезпечує стабільність продуктивності без значних змін у дизайні інтерфейсу.

Користувачі телефонів Apple можуть переміщатися по карті Google, використовуючи все, що вона пропонує, а також інші платформи. Дуже часто можна чути, що інші користувачі платформи скаржаться на те, що не можуть використовувати всі функції, які пропонує Google map, але не для користувачів iOS!

2.3. Android

Android (рис.2.6) - це операційна система від Google. Його творці намагаються зробити його універсальним і простим у налаштуванні. Система використовує дещо інший підхід, ніж у випадку iOS – останнє програмне забезпечення було розроблено так, щоб бути інтуїтивно зрозумілим, але з

обмеженою кількістю опцій. Хоча зміна інтерфейсу та введення додаткових накладок Android не є проблемою, рішення Apple важко змінити з точки зору зовнішнього вигляду - за винятком шпалер або порядку віджетів.

Android — це операційна система, розроблена спеціально для портативних електронних пристроїв, нетбуків, смартфонів і планшетів на базі системи Linux. Його основні переваги – відкритість, простота, зручність та мінімальне споживання пам'яті.



Рисунок 2.6 – Логотип операційної системи Android

На даний момент Android є найпопулярнішою операційною системою у світі. Statcounter від GlobalStats ставить Android на перший план ринку мобільних операційних систем з 69,74% станом на січень 2022 року. Це набагато більше, ніж iOS від Apple, яка є другою за популярністю операційною системою для мобільних пристроїв у світі з 29,49%. Серед незаперечних переваг користувачі Android часто вказують на:

- Швидкість системи;
- Відмінна якість інтернет-серфінгу, високошвидкісний доступ в Інтернет;
- Ідеальна інтеграція з багатьма сервісами Google;
- Великий магазин додатків на будь-який смак і потреби користувача;
- Немає ліцензії на операційну систему, через що вона стає вільною у використанні;

– Більшість додатків є безкоштовними.

В офіційному магазині Google Play доступно понад 2,6 мільйона програм, але ви також можете завантажувати програми з Інтернету. Ця різноманітність робить ці телефони дуже ефективними та налаштовуваними.

Базовий код Android заснований на програмному забезпеченні з відкритим вихідним кодом під назвою Linux. Це означає, що Android за визначенням також має бути відкритим вихідним кодом.

2.4. Microsoft Windows

Windows (рис. 2.7) — загальна назва однієї з найвідоміших операційних систем для комп'ютерів IBM PC. Ця система, розроблена та систематично вдосконалена американською комп'ютерною компанією Microsoft Corporation на початку дев'яностих років, використовуючи прозорий графічний інтерфейс, змінила спосіб спілкування середніх користувачів із комп'ютером, зробивши її цілком корисним інструментом для роботи та розваг також у сфері домашніх програм. Його наступні версії з використанням все більш прогресивних технологій забезпечують інтуїтивно зрозумілу та просту роботу з передовими програмами та керування пристроями, підключеними до комп'ютера. Разом із системою виробник також надає користувачеві базові текстові та графічні додатки, а за додаткову плату — розширені офісні пакети, такі як MS, що тісно співпрацюють із системою.



Рисунок 2.7 – Логотип операційної системи Windows

Актуальною версією операційної системи Windows є версія Windows 10. І головні її особливості:

- покращено функціональність робочого столу;
- прискорено та оптимізовано роботу всієї системи;
- підвищена захищеність;
- інтегрований голосовий помічник зі штучним інтелектом Cortana (який отримав ім'я на честь персонажа з гри Halo);
- ексклюзивно для Windows 10 поставляється Direct X 12, що особливо цікаво буде хардкорним геймерам;
- універсальність додатків під консолі, комп'ютери та мобільні пристрої.

Графічний інтерфейс Windows змінюється від версії до версії, що змушує виробників програмного забезпечення змінювати свої програми, але є багато програм, які пропонують власні рішення у зовнішньому вигляді та в спілкуванні з користувачем.

2.5. Вибір мови програмування для розробки ПЗ

C # (рис.2.8) - сучасна, об'єктно-орієнтована та безпечна для типів мова програмування. C # дозволяє розробникам створювати багато типів безпечних і надійних програм, які працюють у .NET. C # має свої коріння в сімействі мови C і буде відразу знайомим програмістом C, C ++, Java та JavaScript.

Синтаксис C # значною мірою спрощує складність мови C і пропонує унікальні функції, такі як значення NULL, обчислювані вирази, делеговані вирази, лямбда-вирази та прямий доступ до пам'яті, які недоступні в Java. C # підтримує стандартні методи та типи, які пропонують кращу безпеку та продуктивність, а також ітератори, які дозволяють реалізаторам класів конструкторів описувати користувацькі ітераційні поведінки, які клієнтський код може легко використовувати. Інтегровані в запити вирази запиту (LINQ) створюють запит із розширеним типом першокласних мовних конструкцій.



Рисунок 2.8 – Логотип мови програмування C#

C# — це компонентно-орієнтована мова програмування. C# надає мовні конструкції для безпосередньої підтримки цих концепцій, роблячи C# природною мовою для створення та використання програмних компонентів. З моменту свого зародження C# додав функції для підтримки нових робочих навантажень і нових методів проектування програмного забезпечення. За своєю суттю C# є об'єктно-орієнтованою мовою.

Завдяки тому, що мова C# - це мова, яка постійно розробляється Microsoft, вона залишається в авангарді найпопулярніших мов, що використовуються для програмування. Його застосування досить широке - від створення ігор PokemonGO або Angry Birds, через великі складні системи, які використовуються, в тому числі, банками або логістичними компаніями, до програмування ботів і дронів.

Незважаючи на те, що у світі програмістів важко передбачити, в якому напрямку рухатися на кілька років вперед, оскільки розвиток усіх ІТ-сфер і технологій дуже динамічний, можна однозначно сказати, що мова C# не зникне. Через те, що в ньому написано багато програм, попит на програмістів, які займаються їх адмініструванням, точно не зменшиться. Як мова, яка постійно розвивається і вдосконалюється, вона дає все більше можливостей – завдяки

об'єктно-орієнтованому програмуванню це чудовий інструмент для роботи над штучним інтелектом.

Програми на C# працюють на .NET, віртуальній системі виконання, яка називається загальномовним середовищем виконання (CLR) і наборі бібліотек класів. CLR — це впровадження Microsoft спільної мовної інфраструктури (CLI), міжнародного стандарту. CLI є основою для створення середовищ виконання та розробки, в яких мови та бібліотеки безперервно працюють разом.

Вихідний код, написаний на C#, компілюється на проміжну мову (IL), яка відповідає специфікації CLI. Код IL та ресурси, такі як растрові зображення та рядки, зберігаються в збірці, як правило, з розширенням .dll. Збірка містить маніфест, який надає інформацію про типи, версію та культуру збірки.

2.6. Платформа .NET

.NET (рис.2.9) — це безкоштовна кросплатформна платформа розробки з відкритим кодом для створення багатьох різних типів додатків.



Рисунок 2.9 – Логотип .NET

З .NET можна використовувати кілька мов, редакторів і бібліотек для створення веб-сайтів, мобільних пристроїв, настільних комп'ютерів, ігор та Інтернету речей.

При розробці .NET Framework були враховані наступні цілі:

- Забезпечення узгодженого об'єктно-орієнтованого середовища розробки для локального зберігання та виконання об'єктного коду, для виконання локального коду для поширення через Інтернет або для віддаленого виконання.
- Надання середовища виконання коду, в якому:
 - мінімізує ймовірність конфліктів у процесі впровадження програмного забезпечення та управління версіями;
 - гарантується безпечне виконання коду, включаючи код, створений невідомим або повністю довіреним стороннім постачальником;
 - виключає проблеми з продуктивністю при виконанні скриптів або інтерпретації коду;
- Передбачені єдині правила розробки для різних типів програм, таких як програми Windows та веб-додатки;
- Забезпечується галузева сумісність, що забезпечує інтеграцію коду .NET Framework з будь-яким іншим кодом.

2.7. WPF – основні компоненти та інтерфейс середовища

Windows Presentation Foundation (2.10) – це структура інтерфейсу користувача, яка створює настільні клієнти. Платформа розробки WPF підтримує широкий спектр додаткових функцій розробки, включаючи моделі програмного забезпечення, ресурси, елементи керування, графіку, макет, прив'язку даних, документи та безпеку.

WPF є частиною платформи .NET, тому якщо ви раніше створювали програми .NET за допомогою ASP.NET або Windows Forms, ваше середовище розробки має бути знайомим.

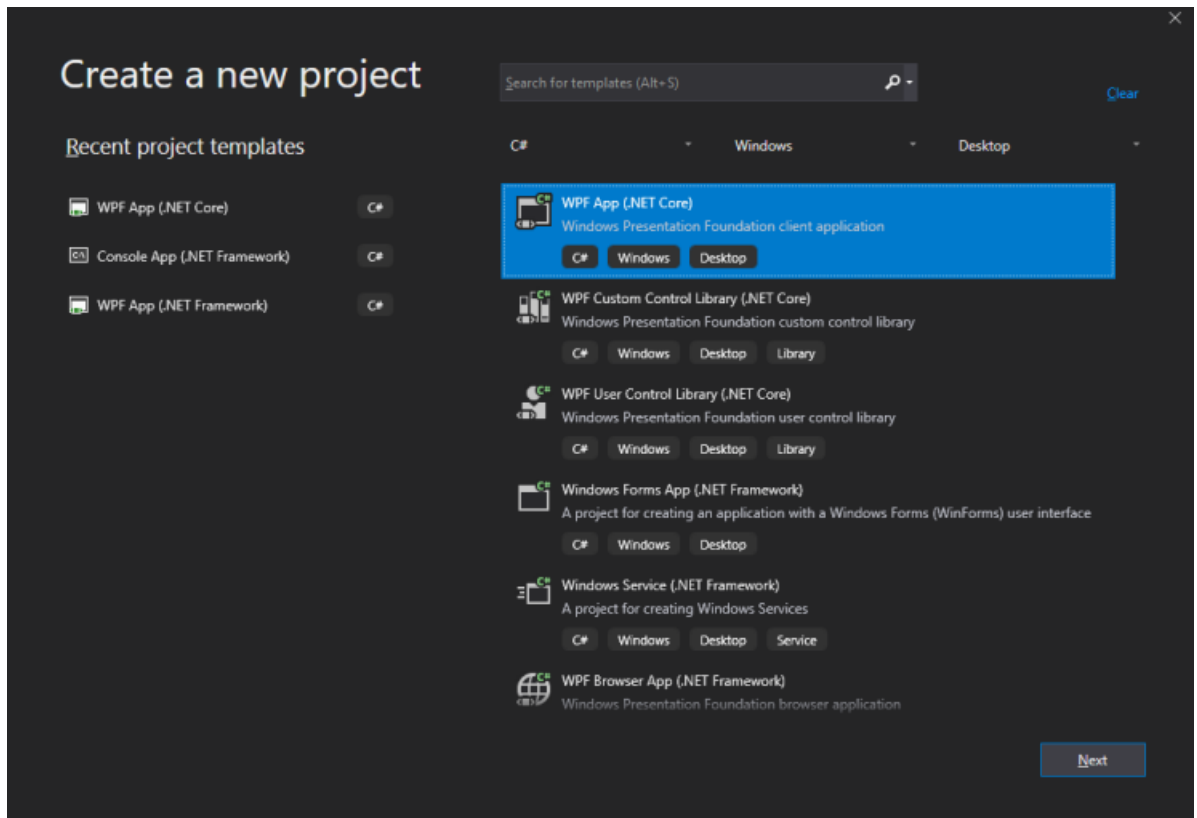


Рисунок 2.10 – Створення проекту використовуючи шаблон WPF

WPF— це механізм, що відповідає за створення, відображення та керування користувальницькими інтерфейсами, документами, зображеннями, елементами керування та медіа в Windows 7.8 та новіших версіях. WPF — це набір бібліотек, який має всі функції, необхідні для створення, запуску та запуску Windows.

До WPF інші фреймворки користувальницького інтерфейсу, пропонувані Microsoft, такі як форми MFC і Windows, були лише обгортками навколо DLL User32 і GDI32, але WPF використовує лише мінімальну кількість User32. Тому,

- WPF — це більше, ніж просто обгортка.
- Це частина платформи .NET.
- Він містить суміш керованого та некерованого коду.

У попередніх фреймворках графічного інтерфейсу не було реального поділу між тим, як виглядає програма і як вона веде себе. І графічний інтерфейс, і поведінка були створені однією мовою, напр. C# або VB.Net, що

вимагало б від розробника більше зусиль для реалізації інтерфейсу користувача та поведінки, пов'язаної з ним.

У WPF елементи інтерфейсу користувача розроблені в XAML, а поведінка може бути реалізована на процедурних мовах, таких як C# і VB.Net. Тому дуже легко відокремити поведінку від коду конструктора.

2.8. Вибір програми для створення дизайну для додатку

Для створення додатку, потрібно розробити інтерфейс з яким буде легше програмувати. В майбутньому його можна буде допрацювати з усіма правилами UI/UX дизайну.

UX – це User Experience, або досвід користувача, який він отримує від процесу користування додатком. Цей розділ відповідає за простоту дизайну, щоб він дуже легко запам'ятовувався людиною. Найважливіше, щоб після довгого періоду, протягом якого людина не користується додатком, він дуже швидко згадав весь інтерфейс і не мав в майбутньому проблем.

UI – це User Interface, або інтерфейс користувача, він відповідає за налаштування самого дизайну та інтерфейсу, як він буде виглядати, які елементи буде мати в собі, кольори, та наскільки зручно буде користуватися саме цим дизайном. Перед усім, це дослідження, після якого можливі зміни, якщо будуть помітні незручності для користувача.

Для створення дизайну свого проекту було обрано сервіс Figma.

Figma (рис.2.11) - одна з найбільш революційних програм для редагування графіки, яка штурмом захопила світ дизайну. Що робить його таким привабливим, так це той факт, що його безкоштовне використання.

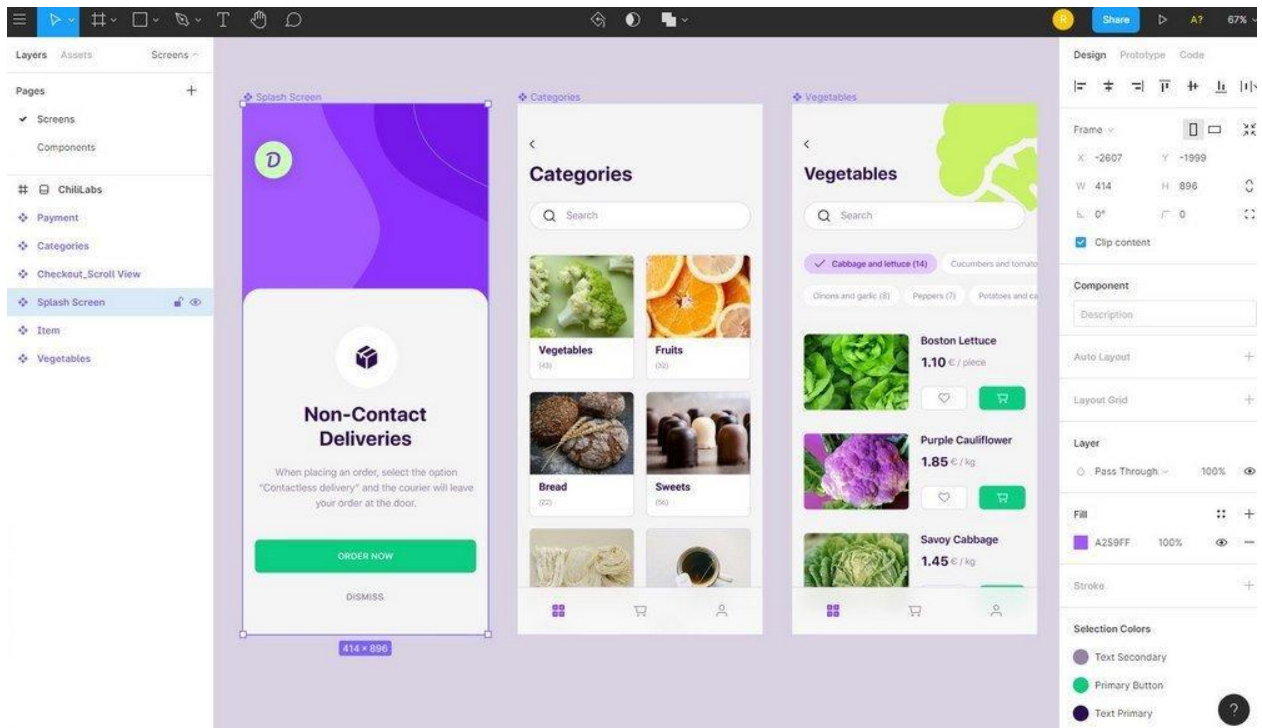


Рисунок 2.11 – Інтерфейс Figma

Figma — це веб-додаток для редагування графіки та дизайну інтерфейсу користувача. Ви можете використовувати його для виконання будь-яких видів графічного дизайну від веб-сайтів, розробки інтерфейсів мобільних додатків, створення прототипів, створення публікацій у соціальних мережах і всього, що між ними.

Figma відрізняється від інших інструментів для редагування графіки. В основному тому, що він працює безпосередньо у вашому браузері. Це означає, що ви отримуєте доступ до своїх проектів і починаєте проектування з будь-якого комп'ютера або платформи без необхідності купувати декілька ліцензій або встановлювати програмне забезпечення.

Десктопна версія Figma доступна на MacOS і Windows. Але використовувати її зовсім не обов'язково. Працювати можна прямо з браузера - в ній доступні всі інструменти і можливості. Підтримуються наступні версії:

- Chrome 49+
- Firefox 51+
- Safari 10.1+
- Edge 14+

Тобто, при переході на сайт [figma.com](https://www.figma.com) з будь-якого ПК або ноутбука можна приступати до роботи. При підключенні до інтернету всі дані макета тут же синхронізуються.

Figma – це платформа, в якій є практично все необхідне для роботи з графікою, векторними об'єктами, шрифтами, ефектами і т. д.

Основний список елементів (Рис 2.12):

- фрейми(Артборди) - є готові варіанти, але можна створювати власні під конкретні завдання.
- модульна сітка- для зручного упорядкування елементів дизайну в фреймах.
- векторні форми- для відтворення різних елементів інтерфейсу.
- криві- для створення кривих і простих векторних форм. Можна додатково завантажити їх з Sketch або Adobe Illustrator.
- зображення- для швидкого додавання графічних елементів в макет.
- Ефекти і маски- в тому числі з ефектами накладення, градієнтами і можливістю зручно працювати з декількома шарами.
- текст- з підтримкою Google Fonts і інструментами підключення додаткових за допомогою Font Installers (для десктопной версії).

Одне з ключових переваг Figma - наявність безкоштовного тарифу. Так, він має деякі обмеження, але в цілому цілком підходить для роботи.

Умови безкоштовного тарифу:

- можливість працювати тільки над трьома проектами одночасно;
- підключення до роботи над проектом не більше ніж одного редактора;
- історія версій зберігається тільки останні тридцять днів.

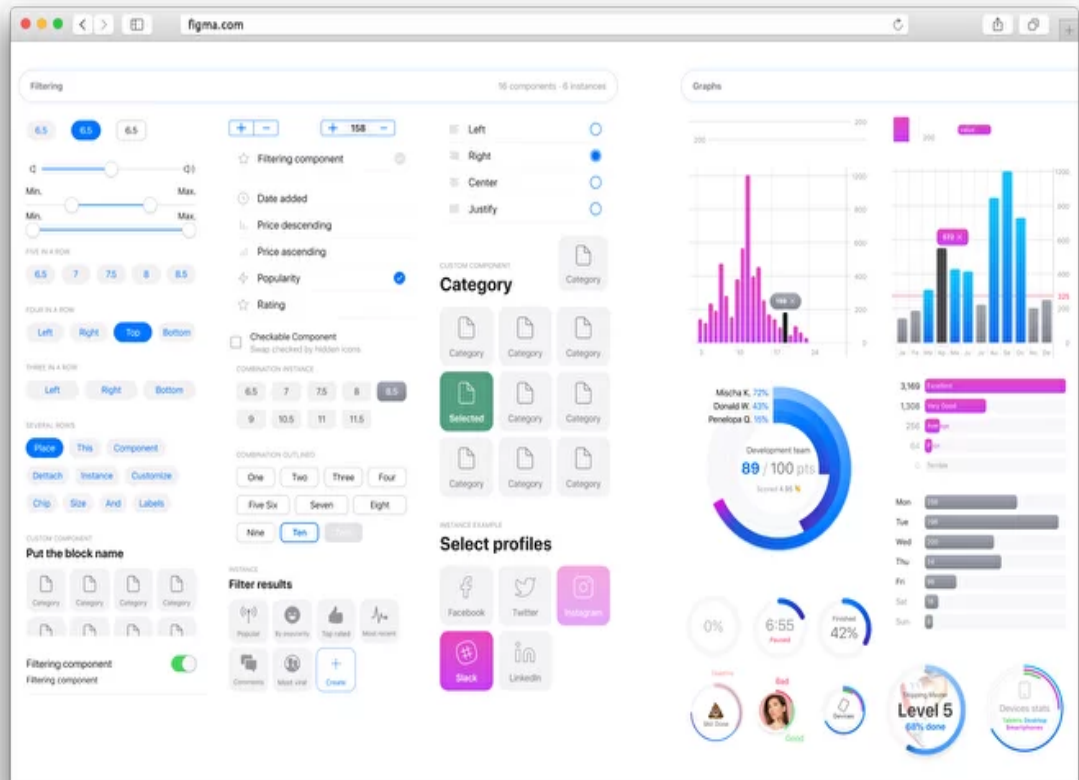


Рисунок 2.12 – Функціонал Figma

3. ПРОЕКТУВАННЯ ТА РОЗРОБКА ДОДАТКУ

3.1. Діаграма варіантів використання

Діаграма використання — це динамічна або поведінкова діаграма в UML. Діаграми використання імітують функціональність системи з акторами та варіантами використання. Параметри використання — це набір дій, послуг і функцій, які повинна виконувати система. У цьому контексті "система" - це те, що розробляється або підтримується, наприклад веб-сайт. «Актори» — це особи або організації, які виконують певні ролі в системі.

Діаграми використання корисні для візуалізації функціональних вимог системи, які впливатимуть на вибір дизайну та пріоритети розробки.

Вони також допомагають визначити будь-які внутрішні або зовнішні фактори, які можуть впливати на систему та повинні бути прийняті до уваги. Вони забезпечують хороший високорівневий аналіз поза системою. Діаграми використання визначають, як система взаємодіє з акторами, не турбуючись про деталі реалізації цієї функції.

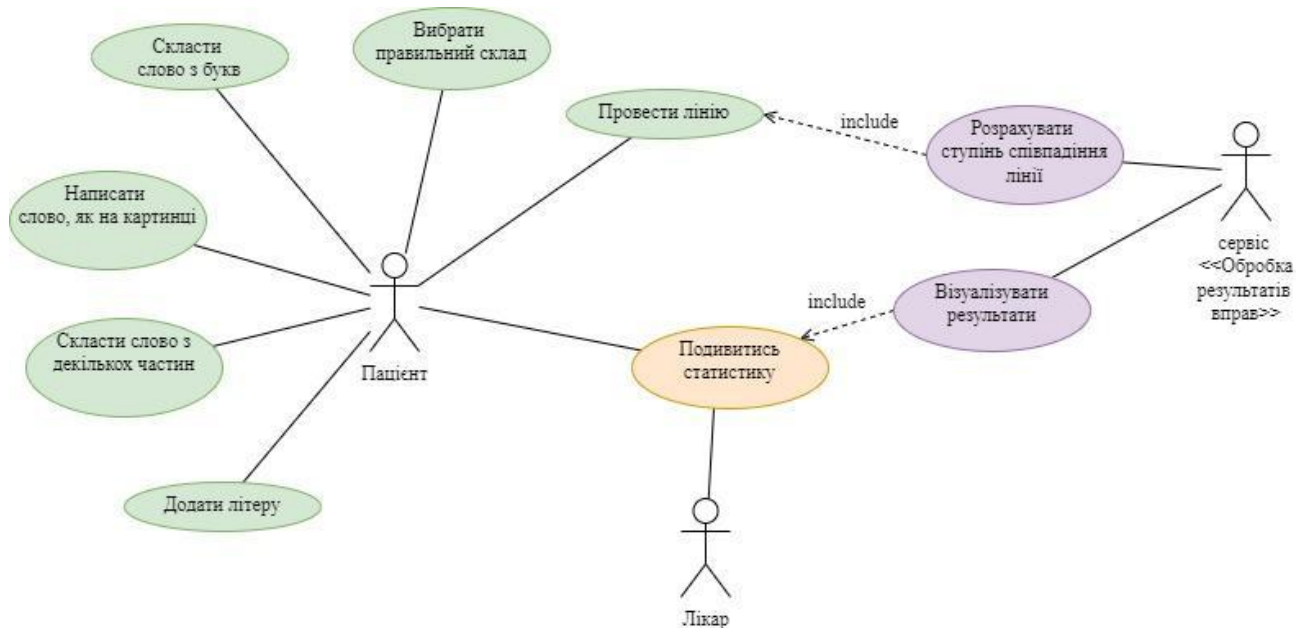


Рисунок 3.1 – Діаграма варіантів використання

В системі передбачено 2 види користувачів – це безпосередньо пацієнт та лікар. Пацієнту доступні різні типи вправ для відновлення письма. Всього реалізовано 6 видів вправ «Додати літеру», «Скласти слово з декількох частин», «Написати слово, як на картинці», «Скласти слово з букв», «Вибрати правильний склад», «Провести лінію».

Додатковою можливістю є відстеження прогресу пацієнта, для легкого розуміння успіхів. Ця функція доступна лікарю та пацієнту.

Таблиця 3.1 – Опис варіантів використання користувача «Пацієнт», які призначені для відпрацювання вправ письмової мови

Варіант використання	Опис вправи, яку він реалізує	Яку навичку письмової мови тренує
Додати літеру	У цій вправі пацієнт повинен додати пропущену літеру у слово, яке зображено на екрані	Сприяє відновленню навичок письма, розвиває мовну компетенцію
Скласти слово з декількох частин	У цьому завданні пацієнт повинен з даних програмою складів, зібрати слово	Допомагає швидко навчитися писати прості й багатоскладові слова, закріпити знання літер і потренувати зорову пам'ять
Написати слово, як на картинці	У цій вправі на екрані буде зображене фото з написом, людина має скласти це слово з генерованих програмою літер.	Збагачує словниковий запас допомагає пацієнту запам'ятати правопис довгих слів

Скласти слово з букв	У вправі необхідно скласти з трьох згенерованих в різному порядку літер слово, на екрані буде зображене фото, яке відповідає цьому слову.	Завдання допомагає пацієнту запам'ятати правопис слів з трьох букв або більше.
Вибрати правильний склад	З наведених на екрані складів, аналізуючи картинку, користувач повинен вибрати правильний склад з усіх представлених, для того щоб вставити його у слово.	Сприяє формуванню навичок експериментування зі словами
Провести лінію	Користувач повинен нарисувати лінію максимально близько до шаблону. В залежності від того, по якій зоні він провів, формується результат виконання вправи	Вправа використовується для відпрацювання рухів м'якої моторики

3.2 Опис математичної моделі вправи «Провести лінію»

При виконанні вправи «Провести лінію» користувач повинен провести максимально від лінії шаблону, який йому демонструється на екрані. На рис. 3.2 наведено приклад шаблону горизонтальної лінії. Зона помічена чорним – це безпосередньо сама лінія, сірий колір визначає околицю лінії. При цьому, чим

темніше сірий колір, тим ближче відповідна зона до лінії шаблону. Оскільки в цій вправі тренується не тільки процес рисування, а й мілка моторика рук, то користувачеві важливо не просто нарисувати лінію, яка буде схожа на шаблон, а й при цьому нарисувати її максимально близько до шаблону.

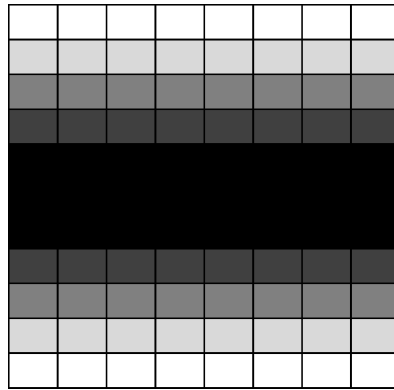


Рисунок 3.2 - Шаблон горизонтальної лінії

Для визначення ступеня близькості будемо використовувати матрицю, представлену на рис.3.3.

0	0	0	0	0	0	0	0	0
0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8
0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
0	0	0	0	0	0	0	0	0

Рисунок 3.3 - Матриця А шаблону лінії

Матриця $A = \{a_{i,j}\}$ визначає коефіцієнти, що визначають ступінь близькості до лінії шаблону, та містить наступні значення:

$a_{i,j} = 1$ для пікселя на лінії шаблону (тобто, якщо користувач попав на лінію шаблону в процесі рисування, то ступінь співпадіння будемо вважати такою, що дорівнює 1);

$a_{i,j} = 0,8$ для пікселя на відстані 1 від лінії шаблону;

$a_{i,j} = 0,5$ для пікселя на відстані 2 від лінії шаблону;

$a_{i,j} = 0,1$ для пікселя на відстані 3 від лінії шаблону;

$a_{i,j} = 0$ для пікселя на відстані 4 та більше від лінії шаблону (якщо

користувач нарисував піксель на такій далекій відстані, то такі пікселі вважаємо помилковими та не враховуємо їх в подальшому. Для цього вони мають коефіцієнт 0).

Коефіцієнти 1, 0.8, 0.5, 0.1, 0 отримано експериментальним шляхом.

Позначимо співпадіння з шаблоном через вагу лінії шаблону VL . Будемо враховувати в цей еталонний показник лише точне співпадіння, таким чином, VL буде розраховуватись як сума одиниць матриці A і виражатиметься формулою (3.1):

$$VL = \sum_{i=1}^M \sum_{j=1}^N a_{i,j} \text{ для } \forall a_{i,j} = 1. \quad (3.1)$$

В результаті виконання вправи користувач рисує певну лінію. На рис.3.2 синім кольором позначено приклад лінії, яку нарисував користувач та її положення відносно шаблону (чорно-сірий колір).

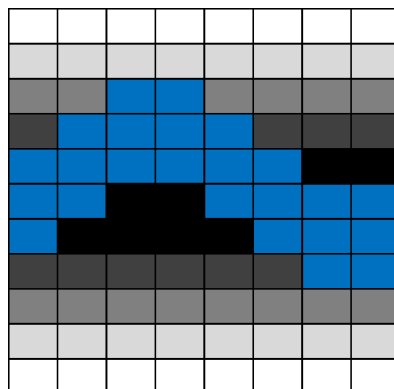


Рисунок 3.4 – Приклад лінії користувача по відношенню до шаблону лінії

Представимо матрицею $B = \{b_{i,j}\}$ лінію, проведену користувачем, де $b_{i,j} = 1$ для кожного пікселя на лінії користувача (синій колір на рис. 3.4).

Приклад матриці B для лінії з рисунку 3.4 наведено на рис.3.5.

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0
0	1	1	1	1	0	0	0	0
1	1	1	1	1	1	0	0	0
1	1	0	0	1	1	1	1	1
1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Рисунок 3.5- Матриця B лінії користувача

Позначимо вагу лінії користувача через VL_{user} та визначимо її як суму тих пікселів матриці шаблону A , для яких є відповідний одиничний піксель в матриці B (3.2):

$$VL_{user} = \sum_{i=1}^M \sum_{j=1}^N a_{i,j} \text{ для } \forall \{i, j\}: b_{i,j} = 1. \quad (3.2)$$

Правильність виконання завдання будемо обчислювати як процентне відношення ваги лінії користувача через VL_{user} до ваги лінії шаблону VL (формула 3.3):

$$R = \frac{VL_{user}}{VL} * 100\% \quad (3.3)$$

Для тренування було розроблено шаблони для різних видів ліній, які є частинами літер – вертикальні, горизонтальні, наклонні та скруглені сегменти.

3.3 Діаграма класів

Діаграма класів показує будівельні блоки будь-якої об'єктно-орієнтованої системи. Діаграми класів зображують статичний вигляд моделі або частини

моделі, описуючи її атрибути та поведінку, а не деталізують методи для виконання операцій. Діаграми класів найбільш корисні для ілюстрації відносин між класами та інтерфейсами. Узагальнення, агрегації та асоціації є цінними для відображення спадкування, складу чи використання та зв'язків відповідно.

Клас – це елемент, який визначає атрибути та поведінку, які може генерувати об'єкт. Поведінка описується можливими повідомленнями, які може зрозуміти клас, а також операціями, які підходять для кожного повідомлення. Класи також можуть містити визначення обмежень, позначені значення та стереотипи.

Діаграма класів додатку представлена на рис 3.6.

Кожна вправа реалізується окремим класом, вправи на підстановку літер та складів реалізовано в одному класі, оскільки вони мають схожу структуру та методи.

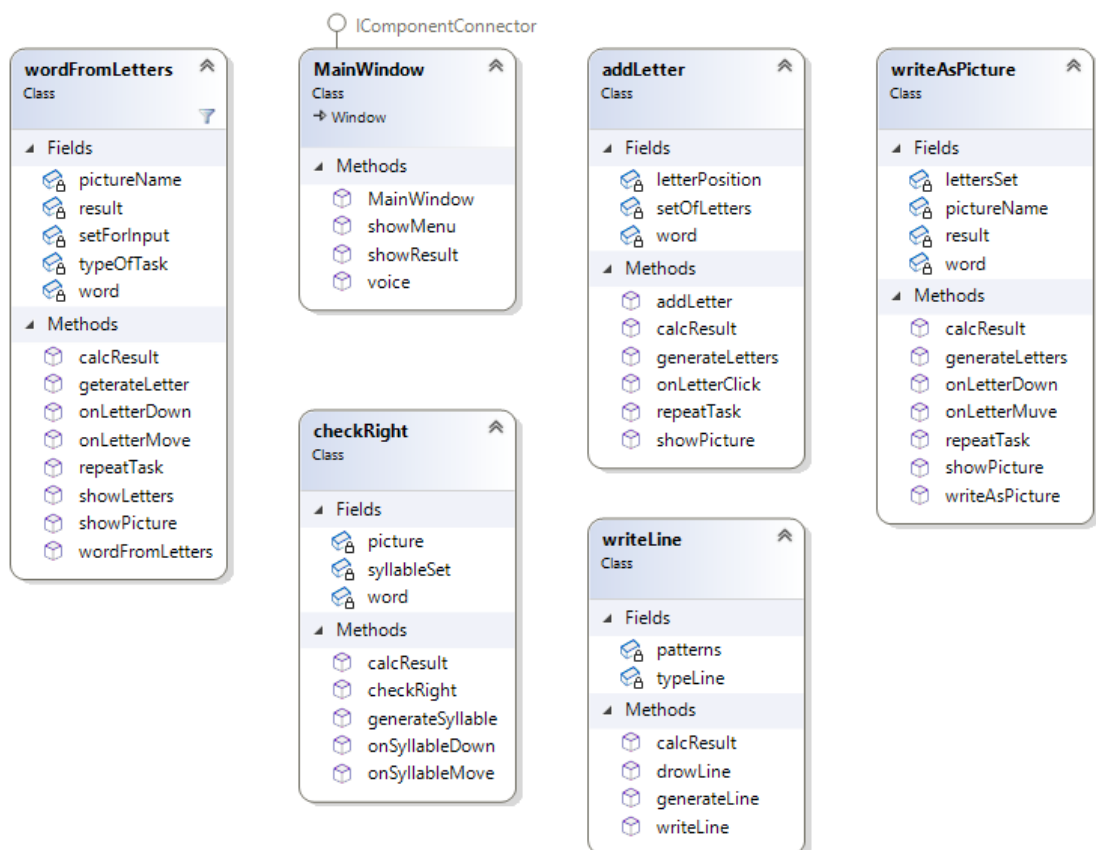


Рисунок 3.6 – Діаграма класів

Клас `wordFromLetters` призначено для реалізації вправ на підстановку літер та складів (вправа «Скласти слово з букв» та «Скласти слово з декількох чатин»). Клас `wordFromLetters` містить в собі п'ять полів та вісім методів:

- Метод `calcResult` підраховує результат виконання вправи.
- Метод `generateLetter` генерує літеру або склад, який користувачу необхідно буде підставити в слово. Для ускладнення вправи вибирається одна літера або склад, який є правильним, і генерується декілька додаткових, які є неправильними варіантами.
- Метод `onLetterDown` призначено для обробки події розміщення літери на необхідній позиції в слові (обробка події «відпускання клавіші миші»). Після того, як спрацьовує цей метод відбувається перевірка, яку саме літеру чи склад перетягнув користувач.
- Метод `onLetterMove` призначено для обробки переміщення літери по робочому полю в процесі операції `drag-and-drop`.
- Метод `repeatTask` дозволяє повторити вправу заново. При цьому обирається на простих рівнях показується те саме слово, з ростом складності обирається випадковим чином слово зі списку доступних слів.
- Метод `showLetters` дозволяє відобразити літери або склади для підстановки в слово. Кожна літера чи склад є інтерактивним елементом, який користувач може переміщувати по екрану зі допомогою технології `drag-and-dprop`.
- Метод `showPicture` дозволяє відобразити картинку, яка відповідає слову в завданні.
- Метод `wordFromLetters` є конструктором класу та дозволяє створити нове вікно з необхідними елементами для діалогу з користувачем відповідно до вправи «Скласти слово з букв».

Клас `MainWindow` містить чотири методи:

- Метод `MainWindow`.
- Метод `showMenu` відображає меню користувача.

- Метод `showResult` відображає результат виконаних вправ пацієнтом, для перегляду прогресу реабілітації.

- Метод `voice` озвучує текст для тих, хто не може прочитати завдання із-за порушень мови.

Клас `addLetter` містить три поля та шість методів:

- Метод `addLetter` є конструктором класу та дозволяє створити нове вікно з необхідними елементами для діалогу з користувачем відповідно до вправи «Додати літеру».

- Метод `calcResult` підраховує результат виконання вправи.

- Метод `generateLetters` генерує літери, які користувачу необхідно буде підставити в слово. Для ускладнення вправи вибирається одна літера, яка є правильною, і генерується декілька додаткових, які є неправильними варіантами.

- Метод `onLetterClick` призначено для обробки натискання літери на екрані.

- Метод `repeatTask` дозволяє повторити вправу заново. При цьому обирається на простих рівнях показується те саме слово, з ростом складності обирається випадковим чином слово зі списку доступних слів.

- Метод `showPicture` дозволяє відобразити картинку, яка відповідає слову в завданні.

Клас `writeAsPicture` містить чотири поля та сім методів:

- Метод `calcResult` підраховує результат виконання вправ.

- Метод `generateLetters` генерує літери або склади, які користувачу необхідно буде підставити в слово. Для ускладнення вправи вибирається одна літера або склад, який є правильним, і генерується декілька додаткових, які є неправильними варіантами.

- Метод `onLetterDown` призначено для обробки події розміщення літери на необхідній позиції в слові (обробка події «відпускання клавіші миші»). Після того, як спрацює цей метод відбувається перевірка, яку саме літеру чи склад перетягнув користувач.

- Метод `onLetterMove` призначено для обробки переміщення літери по робочому полю в процесі операції `drag-and-drop`.

- Метод `repeatTask` дозволяє повторити вправу заново. При цьому обирається на простих рівнях показується те саме слово, з ростом складності обирається випадковим чином слово зі списку доступних слів.

- Метод `showPicture` дозволяє відобразити картинку, яка відповідає слову в завданні.

Клас `checkRight` містить три поля та п'ять методів:

- Метод `calcResult` підраховує результат виконання вправи.

- Метод `checkRight`

- Метод `generateSyllable` генерує склад, який користувачу необхідно буде підставити в слово.

- Метод `onSyllableDown` призначено для обробки події розміщення складу на необхідній позиції в слові (обробка події «відпускання клавіші миші»). Після того, як спрацює цей метод відбувається перевірка, який саме склад перетягнув користувач.

- Метод `onSyllableMove` призначено для обробки переміщення складу по робочому полю в процесі операції `drag-and-drop`.

Клас `writeLine` містить два поля та чотири методи:

- Метод `calcResult` підраховує результат виконання вправи.

- Метод `drowLine` малює лінію

- Метод `generateLine` генерує лінію

- Метод `writeLine` є конструктором класу та відповідає за відображення елементів вікна і генерацію деякого шаблону лінії.

3.4 Опис функціонування додатку

На рисунку 3.7 представлено головне меню додатку, яке відкривається після встановлення додатку. Воно містить логотип додатку та дві кнопки для кожного з користувачів системи – пацієнта та лікаря. Система розрахована на використання одним пацієнтом. В якості користувача «Лікар» може виступати

як безпосередньо лікар пацієнта, так і родичі, які слідкують за динамікою виконання вправ та результатами пацієнта. Оскільки користувач «Пацієнт» має певні проблеми з написанням слів, в тому числі на клавіатурі, то в додатку відсутні будь які засоби авторизації – введення логіну, паролю чи інших елементів, які людина не в змозі написати з фізіологічних причин.

Назва додатка походить від двох слів англійської мови, це слово “Help” яке перекладається як допомога та “UA” скорочення від Україна.

На екрані кнопки з текстом мають однаковий колір, це зроблено для інклюзивності інтерфейсу.

Інтерфейс завдяки своїй контрастності та зручному й зрозумілому макету ідеально підходить для слабозорих користувачів.

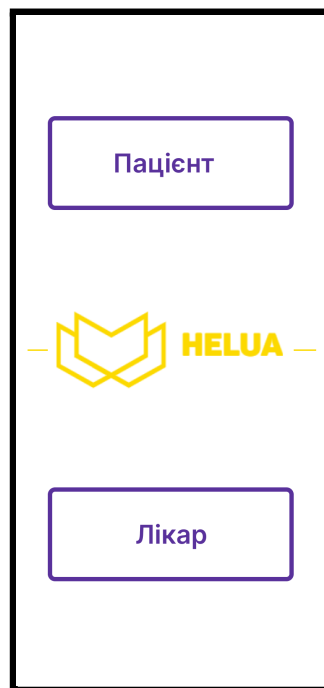


Рисунок 3.7 – Головне меню

Натиснувши на кнопку «Пацієнт» користувач потрапляє на екран з кнопкою «Почати» (Рис. 3.8), де він зможе розпочати виконання вправ для реабілітації при афазії. На екрані зверху присутні ще дві кнопки, кнопка «Назад» повертає користувача до головного меню та кнопка «Озвучення тексту». При натисканні на кнопку озвучення тексту, пацієнт буде чути

приємний і чіткий голос, який буде озвучувати текст на тому екрані де знаходиться сам користувач.

По середині екрану (Рис 3.8) знаходиться текст великим шрифтом «Вправи з письмом» та зображення яке відповідає цьому тексту, всі ці елементи мають тепло-фіолетовій колір, який на фоні дуже добре поєднується з білим.

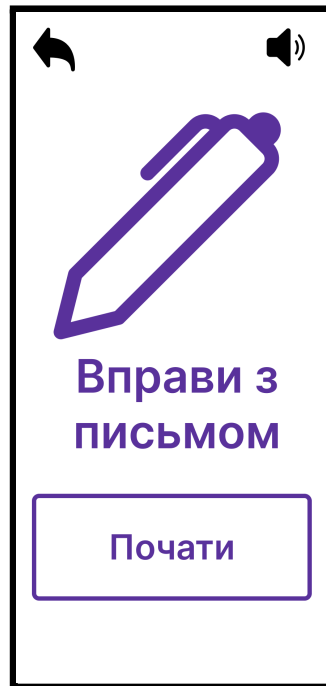


Рисунок 3.8 – Вікно початку виконання вправ

Меню з вибором завдання можна побачити на рисунку (Рис 3.9), кожне завдання озвучується, якщо пацієнт натисне на кнопку «Озвучення тексту», яка знаходиться зверху форми. Також користувач зможе в любий час повернутися до головного меню рис 3.7 натиснувши на кнопку «Додому». На представленому екрані пацієнту доступні різні типи вправ для відновлення письма. На цій екранні формі можна побачити три вправи, а саме «Проведіть лінію», «Складіть слово з декількох частин», «Виберіть правильний склад». Для того щоб пацієнту перейти до виконання інших вправ йому потрібно натиснути на стрілочку, яка зображена на екрані.

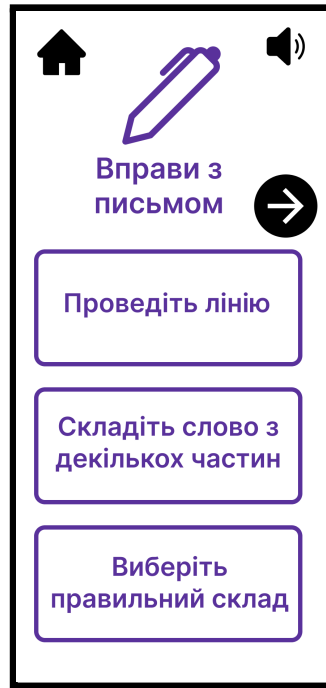


Рисунок 3.9 –Вибір завдань (екран 1)

На рисунку(Рис 3.10) зображені наступні три вправи «Складіть слово з букв», «Пишіть як на картинці» та «Додайте пропущену літеру», які також натиснувши на кнопку «Озвучення тексту» можна буде прослухати. Всі кнопки реалізовані одним розміром для зручності використання. Кнопка «Додому» повертає користувача до головного меню рис 3.7.



Рисунок 3.10 –Вибір завдань (екран 2)

Вправа «Додайте пропущену літеру», яка показана на рисунку (Рис 3.11), розвиває уважність і навички письма пацієнта. На екрані присутнє зображення та назва цього зображення без однієї букви. Виконуючи вправу користувач повинен з п'яти запропонованих літер вибрати одну правильну, для виконання завдання. Завдання пацієнт може прослухати, натиснувши на кнопку «Озвучення тексту». Натиснувши на кнопку «Назад» користувач потрапить до меню вибору завдань рис 3.9, де він зможе приступати до наступної вправи.



Рисунок 3.11 – Вправа «Додайте пропущену літеру»

Вправа «Проведіть лінію» (Рис 3.12) використовується для відпрацювання рухів мілкої моторики – пацієнт повинен нарисувати лінію максимально близько до наведеного шаблону. Тип завдання пацієнт може прослуховувати, натиснувши на кнопку «Озвучення тексту». Для цієї вправи було спроектовано лінію сірого кольору. В залежності від того, по якій зоні провів пацієнт, формується результат виконання вправи. Натиснувши на кнопку «Назад» користувач потрапить до меню вибору завдань рис 3.9, де він зможе приступати до наступної вправи.

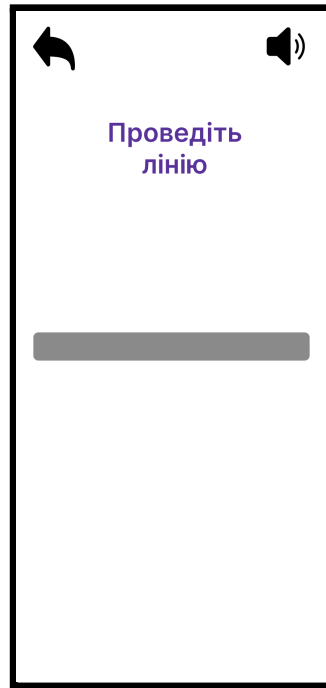


Рисунок 3.12 – Вправа «Проведіть лінію»

Виконуючи завдання «Складіть слово з декількох частин» (Рис 3.13) користувачу доступні для аналізу картинки та слова, розділені на склади в неправильному порядку. Для виконання завдання людині потрібно вставити ці склади у правильному порядку, щоб отримати ціле слово. Це дуже добре розвиватиме зорову пам'ять, дрібну моторику та тренуватиме навички написання багатьох слів різної довжини. Завдання сприятиме збагаченню словникового запасу. Тип завдання пацієнт може прослуховувати, натиснувши на кнопку «Озвучення тексту». Натиснувши на кнопку «Назад» користувач потрапить до меню вибору завдань рис 3.9, де він зможе приступати до наступної вправи.



Рисунок 3.13 – Вправа «Складіть слово з декількох частин»

«Виберіть правильний склад» (Рис 3.14) це завдання в якому у слові бракує одного складу або більше, пацієнту пропонується вписати правильний склад з представлених на екрані. На формі присутні зображення, які пацієнт зможе проаналізувати, для правильної відповіді. Ця вправа допомагає хворому розвинути навички писання друкованих літер, збільшує словниковий запас та розвиває уважність. Тип завдання пацієнт може прослуховувати, натиснувши на кнопку «Озвучення тексту». Натиснувши на кнопку «Назад» користувач потрапить до меню вибору завдань рис 3.9, де він зможе приступати до наступної вправи.



Рисунок 3.14 – Вправа «Виберіть правильний склад»

У вправі «Пишіть як на картинці» (Рис 3.15) представлена картинка з її назвою, користувач повинен з згенерованих в неправильному порядку літер, скласти слово, яке відповідає зразку на екрані. Завдання розвиває увагу, спостережливість, збагачує словниковий запас допомагає пацієнту запам'ятати правопис довгих слів. Тип завдання пацієнт може прослуховувати, натиснувши на кнопку «Озвучення тексту». Натиснувши на кнопку «Назад» користувач потрапить до меню вибору завдань рис 3.9, де він зможе приступати до наступної вправи.

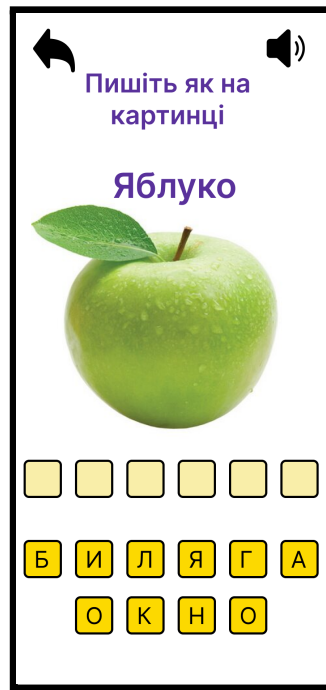


Рисунок 3.15 – Вправа «Пишіть як на картинці»

У наступному завданні «Складіть слово з букв» (Рис 3.16) представлені картинки, дивлячись на них пацієнт повинен з даних в роздріб літер, скласти слово, яке буде відповідати картинці. Виконуючи цю вправу практикується правопис коротких слів, завдання сприяє розвитку уваги й логічного мислення. Тип завдання пацієнт може прослуховувати, натиснувши на кнопку «Озвучення тексту». Натиснувши на кнопку «Назад» користувач потрапить до меню вибору завдань рис 3.9, де він зможе приступати до наступної вправи.

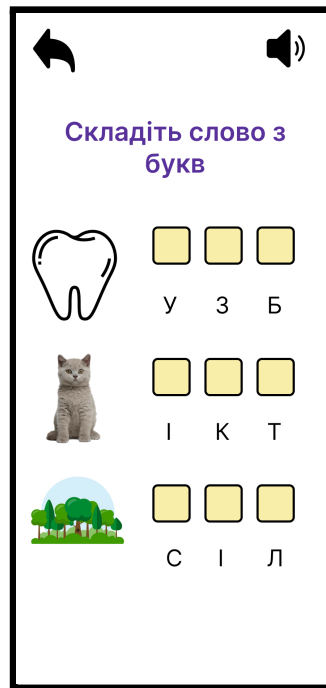


Рисунок 3.16 – Вправа «Складіть слово з букв»

Статистика зображена на рисунку (Рис 3.17) була розроблена спеціально для лікаря або родича, який доглядає за пацієнтом, для того щоб можна було контролювати процес відновлення письмової мови, але доступ до статистики також має пацієнт. Ця статистика буде відображати прогрес кожної виконаної вправи пацієнтом, для легкого розуміння успіхів. Зверху, у випадяючому списку буде представлений вибір завдання, в якому лікар або доглядаюча людина зможе подивитися результат виконаної вправи. Далі загальна точність виконаних вправ, показано як кругова діаграма з відсотками в середині. Нижче лінійна діаграма, з датами та відсотками правильних завдань. В самому низу, види статистик: загальна; розширена.



Рисунок 3.17 – Статистика виконання завдань

ВИСНОВКИ

1. Проведено аналіз предметної галузі. Визначено види вправ, які використовуються для відновлення письмової мови. А саме вправи, спрямовані на стабілізацію та розвиток м'якої моторики пальців та рук і на тренування письмової пам'яті та логіки побудови друкованих слів.

2. Проведено аналіз існуючих додатків, які можна використовувати в процесі реабілітації при афазії для відновлення порушень письмової мови. Були визначені основні недоліки цих додатків, наприклад нестача завдань для розвитку дрібної моторики та інтерфейс, орієнтований тільки на дітей.

3. Виконано проектування додатку: розроблено діаграму варіантів використання, діаграму класів, створено макети форм інтерфейсу.

4. Розроблено програмний додаток мовою C#, який дозволяє відпрацювати шість видів вправ письмової мови. Додаток має дизайн, який враховує особливості користувача. Реалізовано можливість для лікаря та родичів переглядати прогрес пацієнта за допомогою доступу до статистики виконання вправ.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Aphasia [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.hopkinsmedicine.org/health/conditions-and-diseases/aphasia>.
2. FastForWord [Електронний ресурс] – Режим доступу до ресурсу:
<https://logoprognoz.com/metodiki/fastforward/>
3. iName it [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.smartyearsapps.com/iname-it/>
4. CommunicAide: AAC / Speech Therapy [Електронний ресурс] – Режим доступу до ресурсу:
<https://apprecs.com/ios/580971913/communicaide-aac-speech-therapy>
5. Lexico Cognition [Електронний ресурс] – Режим доступу до ресурсу:
<https://lexico.ch/en/lexico-cognition/>
6. What is iOS? [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.makeuseof.com/tag/what-is-ios/>
7. Android [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.android.com/what-is-android/>
8. Microsoft Windows [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.techtarget.com/searchwindowsserver/definition/Windows>
9. Visual Studio [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.crunchbase.com/organization/visual-studio>
10. ANDREW TROELSEN, Philip Japikse, C# 6.0 and the .NET 4.6 Framework / Apress, 2015 – с. 1083.
11. Mark J. Price C# 7 and .NET Core: Modern Cross-Platform Development - Second Edition/ Packt Publishing, 2017 – с. 96.
12. Weil A. Learn WPF C# and the MVVM pattern: Be ready for coding away next week using WPF and MVVM / Arnaud Weil, 2016. – 176 с.
13. Eugene F. Designing in Figma: The complete guide to designing with reusable components and styles in Figma / Fedorenko Eugene, 2020. –164с.
14. Use case diagram [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.smartdraw.com/use-case-diagram/>

15. Waterfall model [Электронный ресурс] – Режим доступа до ресурсу:
<https://uk.myservername.com/what-is-sdlc-waterfall-model>
16. Iterative model model [Электронный ресурс] – Режим доступа до ресурсу:
<https://artoftesting.com/iterative-model>
17. Spiral model [Электронный ресурс] – Режим доступа до ресурсу:
<https://www.geeksforgeeks.org/software-engineering-spiral-model/>
18. Agile model [Электронный ресурс] – Режим доступа до ресурсу:
<https://www.javatpoint.com/software-engineering-agile-model#:~:text=%22Agile%20process%20model%22%20refers%20to,beginning%20of%20the%20development%20process.>

ДОДАТОК А

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО- НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка програмного забезпечення для реабілітації при афазії засобами C#, WPF. Спецчастина: розробка програмного модуля для відпрацювання вправ письмової мови.

Виконав студент 4 курсу
групи ПД-44
Міщук Андрій Юрійович
Керівник роботи

К.т.н. доцент кафедри ІПЗ Золотухіна Оксана Анатоліївна
Київ – 2022

АНАЛОГИ



ПОРІВНЯЛЬНА ТАБЛИЦЯ АНАЛОГІВ

Додатки	Переваги	Недоліки
iName It	Заповнення пропущених літер та складів у словах. Письмо найзвичніших слів - ідеограм списування простих текстів. Є функція озвучки завдань.	Не вистачає озвучення тесту. Немає вправи на відпрацювання рисування ліній, що є елементами літер. Завдання ніяк не прогресують
Fast ForWord	Заповнення в текстах при письмі і пропущених слів, що відпрацьовуються в мовленні. Розкладання підписів під картинками (предметними та сюжетними).	Не вистачає озвучення тесту. Маленький шрифт. Програма не підлаштовується під рівень пацієнта, комусь завдання будуть надто важкі, а комусь навпаки легкі.
Communicaid	Списування слів, фраз та невеликих текстів із встановленням на самоконтроль та самостійне виправлення помилок Є функція озвучки завдань.	Немає вправи на відпрацювання рисування ліній, що є елементами літер. Важкий для сприйняття інтерфейс.
Lexico Cognition	Писання під диктовку слів з звуковою структурою, що поступово ускладнюється, простих фраз, а також окремих складів і літер. Є функція озвучки завдань.	Немає вправи на відпрацювання рисування ліній, що є елементами літер. Завдання орієнтовані тільки на дитячу аудиторію.

3

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи** – ІТ-підтримка процесу реабілітації пацієнтів з афазією шляхом відпрацювання вправ письмовою мови за допомогою програмного забезпечення.
- **Об'єкт дослідження** – виконання вправ по відновленню письмової мови.
- **Предмет дослідження** – програмне забезпечення для виконання вправ по відновленню письмової мови.

4

ТЕХНІЧНІ ЗАВДАННЯ

1. Розробка вправ різної складності та типу для тренування письма українською мовою для пацієнтів при афазії:
 - вправи, спрямовані на стабілізацію та розвиток мілкої моторики пальців та рук;
 - вправи, спрямовані на тренування письмової пам'яті та логіки побудови друкованих слів.
2. Підтримка дизайну, орієнтованого на хворих на афазію – вікові особливості, психологічний стан, можливі фізіологічні проблеми з пальцями та руками.
3. Можливість для лікаря та родичів переглядати прогрес пацієнта за допомогою доступу до статистики виконання вправ.

5

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



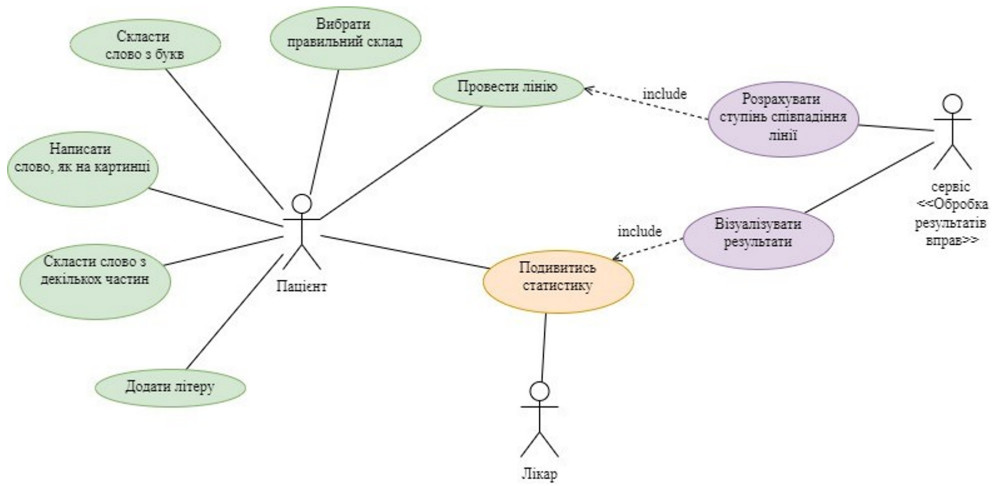
Розробка програмного коду



Проектування та розробка інтерфейсу

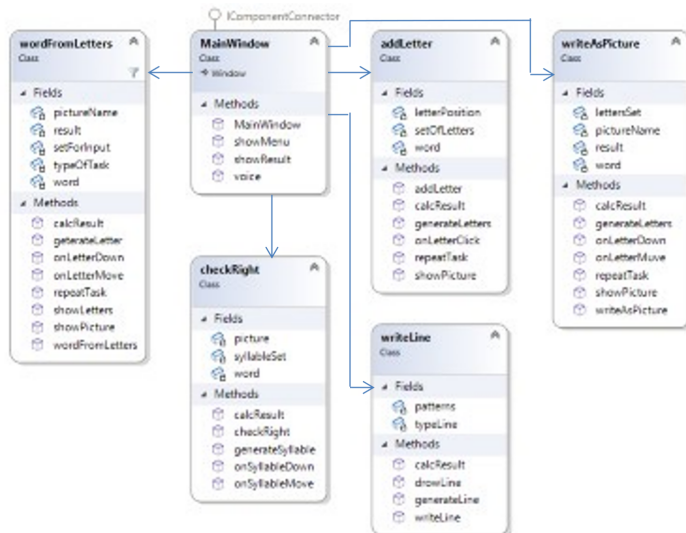
6

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



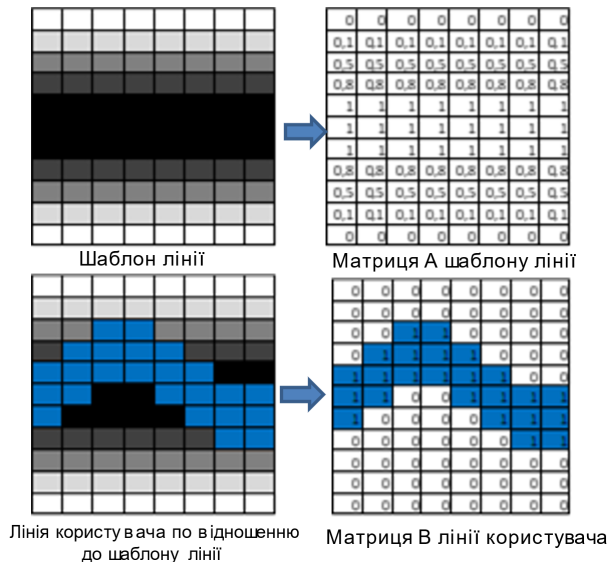
7

ДІАГРАМА КЛАСІВ



8

МОДЕЛЬ ОБРОБКИ РЕЗУЛЬТАТІВ ВИКОНАННЯ ВПРАВИ «ПРОВЕДІТЬ ЛІНІЮ»



Матриця $A = \{a_{i,j}\}$ визначає лінію шаблону:

$a_{i,j} = 1$ для пікселя на лінії шаблону

$a_{i,j} = 0,8$ для пікселя на відстані 1 від лінії шаблону

$a_{i,j} = 0,5$ для пікселя на відстані 2 від лінії шаблону

$a_{i,j} = 0,1$ для пікселя на відстані 3 від лінії шаблону

$a_{i,j} = 0$ для пікселя на відстані 4 та більше від лінії шаблону

Вага лінії шаблону VL :

$$VL = \sum_{i=1}^M \sum_{j=1}^N a_{i,j} \text{ для } \forall a_{i,j} = 1.$$

Матриця $B = \{b_{i,j}\}$ визначає лінію, проведену користувачем:

$b_{i,j} = 1$ для пікселя на лінії користувача

Вага лінії користувача VL_{user} :

$$VL_{user} = \sum_{i=1}^M \sum_{j=1}^N a_{i,j} \text{ для } \forall \{i,j\}: b_{i,j} = 1.$$

Правильність виконання завдання:

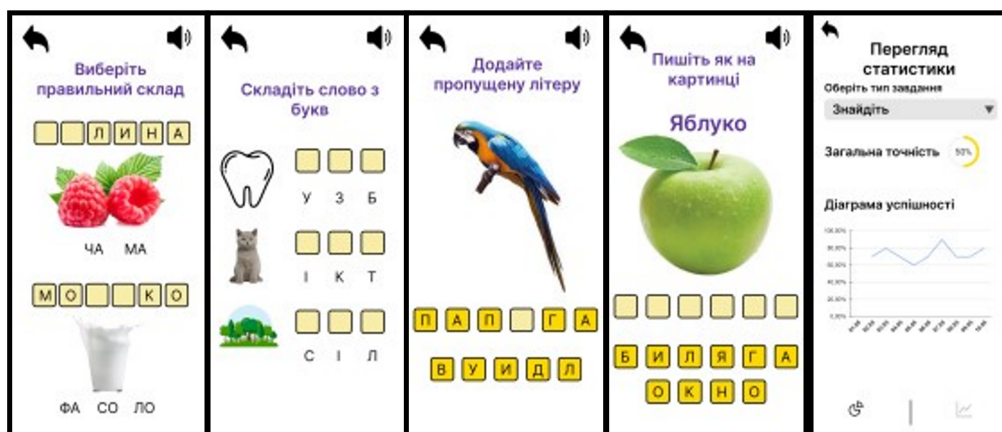
$$R = \frac{VL_{user}}{VL} * 100\%$$

9

ЕКРАННІ ФОРМИ ДОДАТКУ



ЕКРАННІ ФОРМИ ДОДАТКУ



11

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

- Золотухіна О.А. Використання програмного забезпечення в реабілітації порушень письмової мови при афазії/ Золотухіна О.А., Міщук А.Ю. // Науково-технічна конференція застосування програмного забезпечення в інфокомунікаційних технологіях. Збірник тез. 20.04.2022, ДУТ, м.Київ— К.: ДУТ, 2022.— С. 75.
- Міщук А.Ю. Інклюзивний інтерфейс / Діджиталізація науки як виклик сьогодення. 03.06.2022.— С. 183. Подано до друку

12

ВИСНОВКИ

1. Проведено аналіз предметної галузі. Визначено види вправ, які використовуються для відновлення письмової мови.
2. Проведено аналіз існуючих додатків, які можна використовувати в процесі реабілітації при афазії для відновлення порушень письмової мови.
3. Виконано проектування додатку: розроблено діаграму варіантів використання, діаграму класів, створено макет форм інтерфейсу.
4. Розроблено програмний додаток мовою C#, який дозволяє відпрацювати шість видів вправ письмової мови. Додаток має дизайн, який враховує особливості користувача. Реалізовано можливість для лікаря та родичів переглядати прогрес пацієнта за допомогою доступу до статистики виконання вправ.

ДЯКУЮ ЗА УВАГУ!