

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ УПРАВЛІННЯ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри

Олександр ЛИТВИНЕНКО
«_____» _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)
ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

Тема: Програмний засіб передпольотного контролю стану пілота

Виконавець: _____ Владислава
НОСОВСЬКА

Керівник: _____ Олексій ГЛАЗОК

Нормоконтролер: _____ Євгеній ГУПОТА

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук та технологій

Кафедра комп'ютеризованих систем управління

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо професійна програма «Системне програмування»

Форма навчання денна

ЗАТВЕРДЖУЮ

Завідувач кафедри

Олександр ЛИТВИНЕНКО

« _____ » _____ 2023 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

Носовської Владислави Іванівни

1. Тема кваліфікаційної роботи Програмний засіб передпольотного контролю стану пілота

затверджена наказом ректора від «28» серпня 2023 р. № 1494/ст

2. Термін виконання роботи (проєкту): з 02.10.2023 р. по 31.12.2023 р.

3. Вихідні дані до роботи (проєкту): платформа AWS, середовище розробки Visual Studio, редактор діаграм Microsoft Visio, текстовий процесор Microsoft Word.

4. Зміст пояснювальної записки: вступ, теоретичні аспекти передпольотного контролю стану пілота, вимоги до передпольотного контролю стану пілота, проєктування програмного засобу передпольотного контролю стану пілота, реалізація програмного засобу, висновки.

5. Перелік обов'язкового графічного (ілюстративного) матеріалу:

1) Діаграма варіантів використання програмного засобу пілотом;

2) Діаграма варіантів використання програмного засобу лікарем;

3) Робота програмного засобу (схема алгоритму);

4) Робота аналізатора (схема алгоритму).

6. Календарний план-графік

| № п/п | Етапи виконання кваліфікаційної роботи | Термін виконання | Відмітка про виконання |
|-------|--|---------------------------|------------------------|
| 1 | Ознайомитись з матеріалами за темою роботи, проаналізувати існуючі рішення. Підготувати текст першого розділу | 02.10.2023 -09.10.2023 | |
| 2 | Визначити основні вимоги до ПЗ. Обрати та обґрунтувати вибір інструментів, технологій та мови програмування | 10.10.2023 -15.10.2023 | |
| 3 | Створити структуру програмного засобу. Підготувати текст другого розділу | 16.10.2023 -20.10.2023 | |
| 4 | Визначити функціональні вимоги і архітектурний дизайн ПЗ. Розробити алгоритми аналізу передпольотного стану. Підготувати текст третього розділу | 20.10.2023 -31.10.2023 | |
| 6 | Створити базу даних та інтерфейс для програмного засобу | 01.11.2023 -15.11.2023 | |
| 7 | Написати та відкомпілювати програмний код. Оформити текст четвертого розділу | 16.11.2023 -31.11.2023 | |
| 8 | Завершити оформлення пояснювальної записки. Оформити графічний матеріал | 01.12.2023 -07.12.2023 | |
| 9 | Оформити доповідь та презентацію. Підготовка до захисту | 08.12.2023 -25.12.2023 | |
| 11 | Захист кваліфікаційної роботи | 26.12.2023 -31.12.2023 | |

7. Дата отримання завдання: «02» жовтня 2023 р.

Керівник кваліфікаційної роботи _____

Олексій ГЛАЗОК

Завдання прийняв до виконання _____

Владислава НОСОВСЬКА

РЕФЕРАТ

Пояснювальна записка до кваліфікаційного проєкту «Програмний засіб передпольотного контролю стану пілота» містить 87 с., 18 рис., 1 табл., 25 літературних джерел, 1 додаток.

Ключові слова: ПРОГРАМНИЙ ЗАСІБ, СИСТЕМА ПЕРЕДПОЛЬОТНОГО КОНТРОЛЮ СТАНУ ПІЛОТА, АЛГОРИТМИ АНАЛІЗУ, МОВА ПРОГРАМУВАННЯ *PYTHON*, *AWS*, *MAS*.

Об'єктом дослідження є процес аналізу передпольотного стану пілота.

Предметом дослідження є програмний засіб передпольотного контролю стану пілота.

Метою кваліфікаційної роботи є розробка програмного засобу передпольотного контролю стану пілота.

Методи проєктування – розробка програмного засобу на мові програмування *Python* з використанням високорівневої бібліотеки *NumPy*, проєктування схеми бази даних, алгоритмів аналізу та прийняття рішень, тестування розробленого засобу.

Результатом виконання кваліфікаційної роботи є розроблений програмний засіб передпольотного контролю стану пілота, який призначений для аналізу передпольотного стану пілота та запобігає можливим аваріям через людські помилки. Програмний засіб спрямований на постачання пілотам, диспетчерам та лікарям доступу до даних про фізичний та психологічний стан пілота перед вильотом.

Результати проєктування рекомендується використовувати в авіаційній сфері для підвищення рівня безпеки польотів та зниження ризику людських помилок.

ЗМІСТ

| | |
|---|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ | 6 |
| ВСТУП..... | 7 |
| РОЗДІЛ 1 ТЕОРЕТИЧНІ АСПЕКТИ ПЕРЕДПОЛЬОТНОГО КОНТРОЛЮ | 10 |
| 1.1. Психофізіологічні аспекти пілотажу | 10 |
| 1.2. Огляд існуючих рішень | 16 |
| 1.3. Основні завдання передпольотного контролю | 19 |
| 1.4. Виявлення фізичних та психологічних проблем пілота | 25 |
| 1.5. Висновки до розділу | 28 |
| РОЗДІЛ 2 ВИМОГИ ДО СИСТЕМИ ПЕРЕДПОЛЬОТНОГО КОНТРОЛЮ | 30 |
| 2.1. Вимоги до системи | 30 |
| 2.2. Вибір інструментів та технологій | 33 |
| 2.3. Вибір мови програмування та інструментів для розробки | 35 |
| 2.4. Обґрунтування вибору технологій та бібліотек для реалізації | 36 |
| 2.5. Висновки до розділу | 38 |
| РОЗДІЛ 3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАСОБУ | 39 |
| 3.1. Введення до проєктування | 39 |
| 3.2. Структура програмного засобу контролю передпольотного стану | 40 |
| 3.3. Архітектурний дизайн | 42 |
| 3.4. Функціональні вимоги | 49 |
| 3.5. Алгоритми аналізу фізичного та психологічного здоров'я пілотів | 50 |
| 3.6. Висновки до розділу | 58 |
| РОЗДІЛ 4 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАСОБУ | 60 |
| 4.1. Загальний опис програмного засобу | 60 |
| 4.2. Взаємодія з сервісами <i>Amazon</i> | 69 |
| 4.3. Інтерфейс користувача | 76 |
| 4.4. Рекомендації до програмного засобу | 80 |
| 4.5. Висновки до розділу | 82 |
| ВИСНОВКИ | 84 |
| СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ | 86 |
| ДОДАТОК А | 89 |



ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

| | | |
|-------------|---|---|
| <i>API</i> | – | <i>Application Programming Interface</i> |
| <i>MAS</i> | – | <i>Medical Analysis System</i> |
| <i>DMS</i> | – | <i>Document Management System</i> |
| <i>SIS</i> | – | <i>Security Information System</i> |
| <i>UI</i> | – | <i>User Interface</i> |
| ЕЕГ | – | Електроенцефалограма |
| <i>HTTP</i> | – | <i>HyperText Transfer Protocol</i> |
| <i>AWS</i> | – | <i>Amazon Web Services</i> |
| МРТ | – | Магнітно-резонансна томографія |
| КТ | – | Комп'ютерна томографія |
| НМ | – | Нейрона мережа |
| <i>IAM</i> | – | <i>AWS Identity and Access Management</i> |
| <i>EHR</i> | – | <i>Electronic health record</i> |



ВСТУП

Актуальність теми. Авіація завжди вважалася сферою вищого ступеня відповідальності і надзвичайної складності, де навіть найдрібніші помилки можуть мати серйозні наслідки для безпеки і життя людей. Авіаційна безпека завжди була в центрі уваги та досліджень у галузі авіації. Забезпечення надійності та безпеки польотів є найважливішим завданням для керівництва авіаційних підприємств, а також для авіаційних організацій та владних структур кожної країни. Забезпечення безпеки польотів є однією з найважливіших задач в авіаційній сфері, і стан пілота має вирішальне значення для успішного виконання місії. У рамках здійснення польотів, пілоти знаходяться в ситуаціях, де вони мають бути максимально уважними та фізично готовими для виконання завдань. Їхній стан, як фізичний, так і психологічний, може впливати на їхню продуктивність та безпеку польоту.

Один з основних аспектів забезпечення безпеки авіаційних польотів – це передпольотний контроль стану пілота. Цей контроль передбачає оцінку різних аспектів стану пілота, включаючи фізичні показники, рівень стресу, втому та зосередженості. Враховуючи важливість цього аспекту, розвиток програмних засобів для ефективного передпольотного контролю стану пілота стає актуальною задачею у галузі авіації. Більш того, контроль стану пілота перед польотом вимагає високої уваги та точності, оскільки від нього залежить безпека польоту та життя пасажирів і екіпажу.

Сучасні процеси передпольотного контролю стану пілотів використовують стандартизовані підходи, які не враховують особливості та індивідуальні ризики кожного пілота, таких як медична історія, генетичні особливості та інші унікальні аспекти здоров'я кожного пілота. Також не вирішена проблема більш активного залучення самих пілотів до цього процесу, в той час як вони б могли, використовуючи сучасні інформаційні технології, самостійно надавати та моніторити дані про своє здоров'я.

Ця кваліфікаційна робота присвячена розробці та аналізу програмного засобу передпольотного контролю стану пілота. У контексті постійного технологічного



розвитку та зростання обсягу даних, які можуть бути використані для аналізу, ця робота розглядає можливості використання сучасних технологій, таких як аналіз медичних даних та нейронні мережі, для покращення передпольотного контролю стану пілота. Даний програмний засіб покликаний використовувати сучасні технології, включаючи нейронні мережі та обробку даних, для об'єктивної оцінки фізичного і психологічного стану пілота перед польотом.

Мета і завдання кваліфікаційної роботи. Метою кваліфікаційної роботи є розробка програмного засобу для аналізу передпольотного контролю стану пілота, що полягає в розробці та аналізі програмного засобу, який допоможе підвищити безпеку польотів та запобігти можливим аваріям через людські помилки.

Згідно з метою проекту було визначено такі критерії розробки:

- розробка та імплементація програмного засобу з ручним інтерфейсом для пілотів та лікарів;

- відображення медичних показників пілота;

- розробка аналізу зчитаних показників;

- розробка алгоритму роботи програми.

Розробка програми включає в себе реалізацію таких функцій:

- розробка головного меню програмного засобу;

- демонстрація зчитаних медичних даних;

- вивід результатів аналізу медичних даних;

- збереження вхідної та вихідної інформації на сервер.

Об'єкт і предмет проектування. Об'єкт проектування – процес аналізу передпольотного стану пілота. Предмет проектування – програмний засіб передпольотного контролю стану пілота.

Методи проектування. Для розробки програмного засобу використовується платформа *Amazon Web Services (AWS)*, що пропонує найкращу у своєму класі підтримку повнофункціональних сервісів для обробки даних.

Аналіз передпольотного стану пілота виконується з використанням бібліотек: *Pandas*, *NumPy* та *SciPy*, а також бібліотек для машинного навчання та аналітики: *TensorFlow*, *PyTorch*.



Для надійного зберігання даних застосовано керовану *NoSQL* базу даних – *Amazon DynamoDB*.

Для розробки програмного засобу було обрано мову програмування *Python*.

Практичне значення отриманих результатів. Розроблений програмний засіб в даній кваліфікаційній роботі дозволяє користувачу:

- інтуїтивно-зрозуміло користуватися програмним засобом;
- вивести всю необхідну інформацію для обробки в медичних цілях;
- отримати результат аналізу передпольотного стану пілота.

Засіб передпольотного контролю пілота покращує оцінку аналізу медичних даних та забезпечує аналіз та обробку великого обсягу даних, що в свою чергу сприяє більш якісному контролю та допуску пілотів до польоту. Програмний засіб надає можливість зберігати, зчитувати, аналізувати та виводити медичні показники як лікарям, так і пілотам, оптимізуючи використання засобу, адже це є важливим фактором для медичного персоналу враховуючи економію часу на медичний огляд.

Апробація отриманих результатів. Результати кваліфікаційної роботи було представлено на міжнародній науково-технічній конференції «Інтелектуальні технології лінгвістичного аналізу» та IV Міжнародній науково-практичній конференції «*Researching Advanced Horizons of Global Progress: Challenges and Innovative Concepts*» (13-15 грудня 2023 року).

Результати дослідження дозволяють зробити висновок про доцільність імплементації розробленого програмного засобу та можливість його успішного впровадження в авіаційній сфері з метою підвищення рівня безпеки польотів та зниження ризику людських помилок.



РОЗДІЛ 1

ТЕОРЕТИЧНІ АСПЕКТИ ПЕРЕДПОЛЬОТНОГО КОНТРОЛЮ

1.1. Психофізіологічні аспекти пілотажу

У сучасній авіаційній галузі безпека є пріоритетною метою. Під час польотів, велику роль у забезпеченні безпеки виконує не лише технічний стан повітряних суден, але і психофізіологічний стан пілота. Передпольотний контроль стану пілота дозволяє переконатися у його працездатності та здатності приймати рішення під час польоту. У даній роботі ми детально розглянемо теоретичні аспекти передпольотного контролю стану пілота та психофізіологічні аспекти пілотажу, звертаючи увагу на важливість цих аспектів для безпечної авіаційної діяльності.

Авіаційна безпека – це одна з найважливіших складових сучасної авіації. Кожен політ повинен бути максимально безпечним для пасажирів, екіпажу, вантажу та для навколишнього середовища. Для досягнення цієї мети, пілоти повинні бути не лише досвідченими і висококваліфікованими виконавцями, але й мати добрий психофізіологічний стан. Передпольотний контроль стану пілота є ключовим елементом забезпечення цієї безпеки [3].

Психофізіологічний стан пілота є ключовим фактором, що визначає його здатність ефективно виконувати свої обов'язки під час польоту. Цей стан включає в себе два основні аспекти: фізичний і психічний, які взаємодіють між собою і впливають на поведінку та прийняття рішень пілота в різних ситуаціях під час польоту.

Фізична підготовка пілота є однією з ключових складових його фізичного стану. Пілот повинен бути у відмінній фізичній формі, оскільки він стикається з фізичними навантаженнями під час маневрів, зльотів і посадок. Інтенсивність цих фізичних навантажень може бути дуже високою, і пілот повинен мати достатню фізичну витривалість, щоб впоратися з ними. Це включає в себе здатність



витримувати перевантаження та інші фізичні напруження, які можуть виникнути під час польоту.

Стан здоров'я пілота є ще однією важливою фізичною складовою його психофізіологічного стану. Перед кожним польотом пілоти проходять медичний огляд, під час якого перевіряються різні аспекти їхнього здоров'я. Це включає в себе вимірювання артеріального тиску, серцевої частоти, огляд зору та слуху, а також виявлення будь-яких медичних проблем, які можуть впливати на здатність пілота керувати літаком [11].

Втома і недостатній сон можуть значно впливати на фізичний стан пілота. Довгі польоти, перехід через часові зони та неправильний графік сну можуть викликати втому, що знижує реакцію на подразники та концентрацію. Важливо дотримуватися регулярного режиму сну та відпочинку, щоб забезпечити оптимальний фізичний стан пілота.

Психічна стійкість пілота – це його здатність ефективно впоратися з стресом і тиском під час польоту. Цей аспект включає в себе вміння приймати рішення в критичних ситуаціях, контролювати страх і тривогу, а також залишатися спокійним і зосередженим в будь-яких умовах. Психічна стійкість є важливою як у звичайних польотах, так і в надзвичайних ситуаціях, коли пілот повинен вміти приймати швидкі та правильні рішення [25].

Пілот повинен бути здатним ефективно сприймати і обробляти велику кількість інформації під час польоту. Це включає в себе роботу з показниками на приладах, зв'язок з контрольним центром, взаємодію з екіпажем та багато іншого. Якщо пілот не може ефективно обробляти інформацію, це може призвести до непорозумінь, помилок та навіть аварій.

Пілоти часто стикаються зі стресовими ситуаціями, такими як погані погодні умови, турбулентність, технічні неполадки або несподівані ситуації. Здатність пілота ефективно управляти стресом і залишатися зосередженим є критичною для безпеки польоту. Стрес може впливати на прийняття рішень та реакцію на небезпеку, тому важливо мати механізми для його подолання.



Фізичні та психічні аспекти психофізіологічного стану пілота взаємодіють між собою і можуть суттєво впливати на дії та рішення пілота під час польоту.

Наприклад, фізична втома може призвести до зниження концентрації та реакції на подразники, що може підвищити ризик помилок. Також, емоційний стан пілота може впливати на його сприйняття і обробку інформації.

Однак важливо зазначити, що ця взаємодія може бути як позитивною, так і негативною. Наприклад, певний рівень стимуляції або емоційної активності може підвищити фізичну і психічну готовність пілота до реагування на надзвичайні ситуації. Однак, якщо стрес чи тривога стають занадто великими, це може призвести до негативного впливу на рішення та дії пілота [13].

Психофізіологічний стан пілота є складною системою, що об'єднує фізичні та психічні аспекти його здоров'я та емоційного стану. Ці аспекти взаємодіють між собою і впливають на здатність пілота керувати літаком та приймати рішення під час польоту. Важливо підтримувати оптимальний фізичний стан через фізичну підготовку, правильний відпочинок та медичний огляд. Крім того, розвивати психічну стійкість, навички управління стресом та сприйняття та обробки інформації є важливими аспектами для підтримання безпеки в авіаційній галузі. Враховуючи ці аспекти, пілоти можуть забезпечити ефективну та безпечну роботу в повітрі.

Фізичний стан пілота – це характеристика його загального стану здоров'я, фізичної підготовки та фізичної витривалості. Пілот повинен бути у відмінній фізичній формі, оскільки керування літаком може бути досить важким і вимагати значного фізичного напруження. Крім того, пілот повинен бути здатний витримувати тривалі періоди сидіння в незручних позах та впливи гравітації під час маневрів.

Психічний стан пілота визначає швидкість та якість сприйняття і обробки інформації, прийняття рішень та реакції на стресові ситуації. Під час польоту пілот повинен бути відповідальним, концентрованим та зосередженим. Відволікання або неспроможність зосередитися можуть призвести до серйозних помилок.

Емоційний стан пілота також має важливе значення. Страх, тривога або негативні емоції можуть вплинути на рішення та дії пілота. На відміну від фізичного



стану, емоційний стан більш важко контролювати, але важливо навчитися ефективно впоратися з ними під час польоту [14].

Передпольотний контроль стану пілота – це комплекс заходів та процедур, спрямованих на оцінку та забезпечення його фізичного, психічного та емоційного стану перед кожним польотом. Цей контроль включає в себе наступні аспекти:

Перед кожним польотом пілоти піддаються медичному огляду. Це включає в себе перевірку фізичного стану, вимірювання артеріального тиску, серцевої частоти та інших показників. Медичний огляд допомагає виявити будь-які медичні проблеми, які можуть впливати на здатність пілота керувати літаком. Медичний огляд пілотів перед кожним польотом є однією з найважливіших процедур для забезпечення безпеки авіаційної діяльності.

Данна процедура дозволяє виявити будь-які медичні проблеми або стани, які можуть негативно впливати на фізичний і психічний стан пілота під час польоту. Медичний огляд включає в себе ряд складних та докладних процедур, спрямованих на оцінку загального здоров'я пілота, його фізичних параметрів і можливих медичних ризиків.

Метою медичного огляду є забезпечення того, щоб пілоти, які керують повітряними суднами, були у фізичному стані, який дозволяє їм виконувати свої обов'язки безпечно та ефективно. Основні цілі медичного огляду включають в себе:

- визначення фізичного стану. Проведення огляду і вимірювання показників, таких як артеріальний тиск, серцева частота, температура тіла та інші фізіологічні параметри, щоб визначити загальний фізичний стан пілота;

- виявлення медичних проблем. Визначення наявності або ризику розвитку медичних станів, які можуть впливати на здатність пілота керувати літаком, таких як серцево-судинні захворювання, захворювання дихальних шляхів, епілепсія, діабет, захворювання нирок і багато інших;

- визначення здатності пілота до виконання льотної роботи. Оцінка здатності пілота витримувати підвищені навантаження, виконувати різні маневри, включаючи екстремальні, та реагувати на надзвичайні ситуації;



– попередження несприятливих подій. Виявлення медичних факторів, які можуть впливати на безпеку польоту і запобігання негативним наслідкам.

Процес медичного огляду пілота дуже докладний і включає в себе кілька важливих етапів:

– історія хвороби та анамнез. Пілоти надають докладну інформацію про свою історію хвороби та медичний анамнез. Це включає в себе інформацію про попередні захворювання, лікування, операції та будь-які інші медичні події у їхньому житті;

– фізичний огляд. Лікар проводить фізичний огляд, вимірювання артеріального тиску, пульсу, дихальної частоти, а також перевірку зору та слуху;

– лабораторні дослідження. В певних випадках можуть бути назначені лабораторні аналізи, такі як аналізи крові та сечі, для виявлення патологічних змін або показників хвороб;

– інструментальні дослідження. Іноді можуть бути назначені додаткові інструментальні дослідження, такі як електрокардіографія (ЕКГ), рентген, комп'ютерна томографія (КТ) або магнітно-резонансна томографія (МРТ) для отримання більш детальної інформації [6];

– оцінка психічного стану. Психічний стан пілота також оцінюється, включаючи виявлення ознак тривоги, депресії або інших психічних проблем.

На підставі всіх проведених процедур і досліджень лікар робить висновок про здатність пілота до літання. Цей висновок може бути наступними:

– придатний до польотів. Якщо пілот не має жодних медичних обмежень та його фізичний та психічний стан відповідає вимогам, то він вважається здатним до літання;

– обмежена здатність до виконання польотів. У деяких випадках пілот може мати певні обмеження, наприклад, щодо типу літака, на якому він може працювати, або щодо медичних обмежень, які вимагають регулярного медичного спостереження.

Медичний огляд є ключовим елементом для забезпечення безпеки польоту. Він допомагає виявити медичні стани або ризики, які можуть впливати на здатність пілота керувати літаком. Навіть незначні медичні проблеми, такі як високий



артеріальний тиск, можуть призвести до серйозних наслідків, якщо вони не виявлені та не враховані. Тому медичний огляд має велике значення для запобігання нещасним випадкам та аваріям в авіації [12].

Медичний огляд пілотів перед кожним польотом є необхідною процедурою для забезпечення безпеки авіаційної діяльності. Він дозволяє виявити медичні стани або ризики, які можуть впливати на фізичний і психічний стан пілота під час польоту. Результати медичного огляду визначають здатність пілота до виконання льотної роботи і допомагають попередити можливі медичні небезпеки для безпеки польоту. Тому медичний огляд є важливою частиною системи контролю безпеки в авіації і допомагає забезпечити безпечну авіаційну діяльність.

Психологічна оцінка пілота включає в себе оцінку його психічного стану та емоційного стану. Психолог може проводити спеціальні тести та опитування, щоб виявити ознаки стресу, тривоги або інших психічних проблем [50].

Достатній відпочинок і правильне харчування також мають велике значення для пілотів. Недостатній сон або неправильне харчування можуть призвести до втоми та погіршення фізичного та психічного стану.

Пілоти проходять систематичне тренування та підготовку, щоб підтримувати свої навички та знання на високому рівні. Це включає в себе симулятори польотів, навчальні курси та інші форми навчання.

Передпольотний контроль стану пілота та увага до психофізіологічних аспектів пілотажу мають високе значення для безпеки польоту. Якщо пілот має фізичні або психічні проблеми, він може бути нездатним до виконання своїх обов'язків належним чином, що може призвести до аварій або нещасних випадків. Також, навіть якщо пілот відчуває фізичну втому або стрес, його здатність до прийняття рішень та реакції на непередбачені ситуації може бути обмежена [5].

Передпольотний контроль стану пілота та психофізіологічні аспекти пілотажу є надважливими факторами для забезпечення безпеки пасажирів та екіпажу в авіаційній галузі. Фізичний, психічний та емоційний стан пілота можуть суттєво впливати на його здатність керувати літаком та приймати рішення. Передпольотний контроль та ретельна підготовка допомагають виявити та вирішити проблеми, що



можуть виникнути під час польоту, тим самим забезпечуючи безпечну авіаційну діяльність [3].

1.2. Огляд існуючих рішень

У сфері авіації забезпечення благополуччя та оптимальної роботи пілотів має першорядне значення. Інтеграція передових технологій відкрила нові можливості для моніторингу та контролю стану пілота в режимі реального часу. У цьому тексті розглядаються деякі відомі інструменти та системи, призначені для досягнення цієї мети, а також обговорюються обмеження, пов'язані з цими досягненнями.

NeuroVigil – інноваційна система, заснована на технології ЕЕГ (електроенцефалографії). Її впровадження характеризується як значний прогрес у сфері моніторингу здоров'я пілотів перед польотами. На фотознімку (рис. 1.1) показана спеціальна гарнітура, що використовується для цієї системи. Гарнітура оснащена вбудованими датчиками ЕЕГ, які реєструють мозкову активність пілота.



Рис. 1.1. Гарнітура системи *NeuroVigil*

Використання технології ЕЕГ має першорядне значення для комплексної оцінки різних аспектів психічної рівноваги та загального когнітивного стану пілота.



Основна перевага системи *NeuroVigil* полягає у можливості проводити оцінку когнітивних функцій у реальному часі. Вивчаючи складні патерни мозкових хвиль, вдається оцінити психологічне самопочуття пілота, а також спостерігати за роботою психіки під час прийняття пілотом рішень під час польоту. Це особливо важливо для підтримки зосередженості пілотів, сконцентрованості у виконанні складних завдань, особливо коли вони стикаються зі сценаріями високого стресу. Система *NeuroVigil* добре розпізнає рівень стресу, який відчувають пілоти. Ця здатність досягається шляхом аналізу відхилень у моделях мозкових хвиль, тісно пов'язаних із реакцією на стрес. Виявлення підвищеного стресу має першочергове значення, оскільки це дозволяє авіаційним фахівцям швидко втрутитися або надати підтримку пілотам, коли це необхідно, ефективно пом'якшуючи ризики, пов'язані з підвищеним стресом під час польоту. Крім того, система демонструє здатність виявляти потенційні когнітивні порушення. Порівнюючи мозкову активність у реальному часі з встановленими базовими шаблонами, вона може викликати сповіщення у випадках виявлення порушень. Такі відхилення можуть означати когнітивні проблеми, що виникають через такі фактори, як втома, хвороба або інші змінні, які можуть поставити під загрозу здатність пілота приймати обґрунтовані рішення.

Тим не менш, важливо визнати певні обмеження, пов'язані з моніторингом на основі ЕЕГ. Точна інтерпретація даних ЕЕГ є важким завданням.

Помилкова тривога чи помилкова інтерпретація даних потенційно можуть спровокувати зайве занепокоєння чи навіть паніку. Тому розробка високоточних алгоритмів і методологій інтерпретації даних має першочергове значення. Підводячи підсумок, *NeuroVigil*, система заснована на технології ЕЕГ, це значний стрибок уперед у сфері моніторингу передпольотного контролю стану пілота. Вона надає інформацію про когнітивні функції та рівень стресу в реальному часі, що значно підвищує безпеку авіації. Однак важливо визнати та вирішити вищезазначені обмеження та проблеми, щоб бездоганно інтегрувати ці технології в авіаційну практику, зберігаючи при цьому конфіденційність та автономію пілотів.

Force Cognitive Operations Gear Pack (FCOG) – це інноваційна переносна система, яка використовує штучний інтелект (ШІ) для моніторингу та оцінки



когнітивного стану пілотів та іншого персоналу в режимі реального часу. Система заснована на різноманітних датчиках, включаючи датчики ЕЕГ для вимірювання активності мозку, датчики серцевого ритму та датчики відстеження очей. Дані, зібрані цими датчиками, передаються в алгоритм ШІ, який аналізує їх на наявність ознак когнітивних порушень, стресу та інших факторів, які можуть вплинути на продуктивність (рис. 1.2).

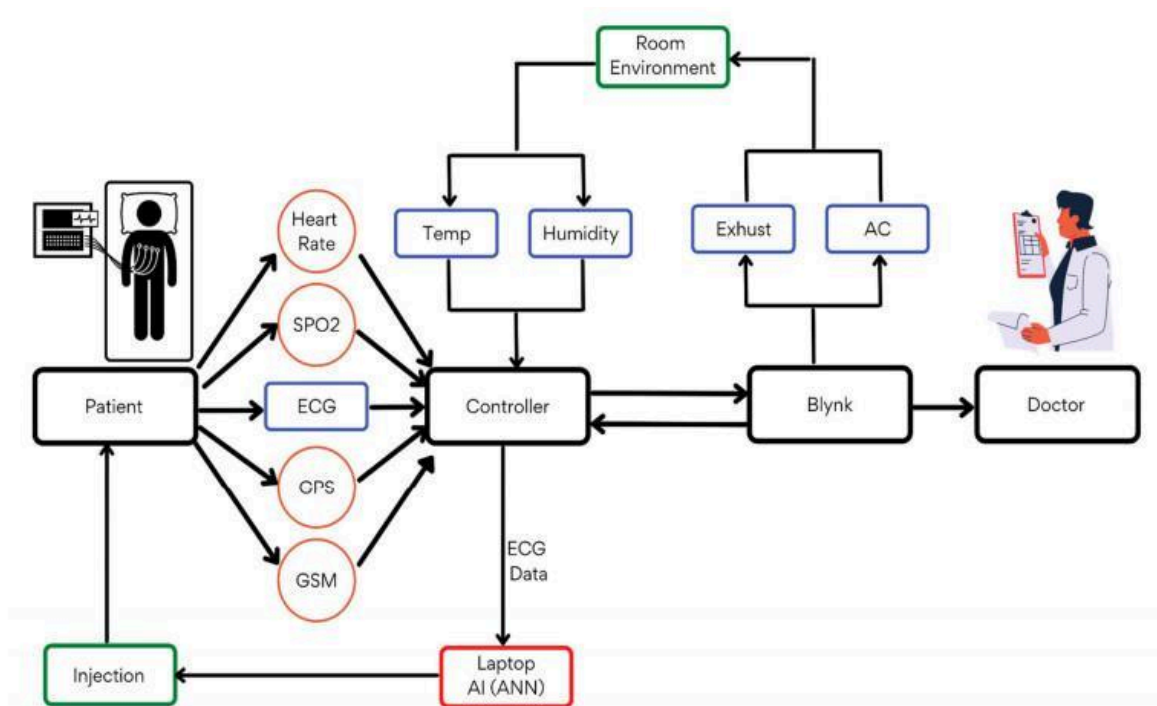


Рис. 1.2. Функціональна схема роботи системи *FCOG*

Система *FCOG* має низку переваг перед традиційними методами моніторингу стану пілотів. По-перше, вона малогабаритна, зручна у носінні та застосуванні, придатна для тривалого використання. По-друге, вона забезпечує зворотний зв'язок у режимі реального часу щодо когнітивного стану, який можна використовувати для виявлення та пом'якшення проблем до того, як вони розвинуться настільки, що почнуть впливати на якість пілотування. По-третє, алгоритм штучного інтелекту має можливість навчатися та вдосконалюється, що з часом робить систему більш точною. Система *FCOG* все ще знаходиться на стадії вдосконалення, але вона має потенціал революціонізувати моніторинг пілотів і підвищити безпеку авіації. Забезпечуючи зворотний зв'язок щодо когнітивного стану в режимі реального часу, система може допомогти пілотам виявити та пом'якшити проблеми до того, як вони



вплинуть на продуктивність. Це може призвести до покращення процесу прийняття рішень, зменшення кількості помилок і підвищення загальної безпеки [4].

Розумний шолом *NASA* – розробка для астронавтів. Хоча вона може не мати прямого зв'язку з традиційною авіацією, вона служить переконливим прикладом того, як подібні технології можуть знайти застосування в авіаційній промисловості. Ця система, встановлена на шоломі, була спеціально розроблена для нагляду за самопочуттям астронавтів під час їхніх космічних місій, охоплюючи моніторинг життєво важливих показників, рівня кисню та включення дисплеїв доповненої реальності. Ця інноваційна технологія підкреслює універсальність інструментів моніторингу за межами наземної авіації.

Тим не менш, незважаючи на величезний потенціал і успіхи, досягнуті в області пілотних програмних засобів моніторингу здоров'я, необхідно визнати кілька важливих обмежень. Процес збору та передачі конфіденційних фізіологічних даних від пілотів викликає серйозні занепокоєння щодо конфіденційності та безпеки. Необхідно ретельно керувати зберіганням і передачею таких даних, щоб запобігти будь-яким потенційним порушенням або неправильному використанню, які можуть мати значні наслідки.

Крім того, в контексті авіації надійність і довговічність обладнання для моніторингу виділяються як першочергові міркування. Виникнення будь-якої несправності або збою в роботі приладів управління під час польоту може призвести до тяжких наслідків. Таким чином, виробники повинні переконатися, що їхнє обладнання є не лише технологічно передовим, але й достатньо міцним, щоб витримати складні умови авіаційного середовища.

1.3. Основні завдання передпольотного контролю

Передпольотний контроль став важливим етапом в авіаційній безпеці, забезпечуючи аналіз медичних даних пілотів перед кожним польотом. Основні завдання цього контролю включають в себе використання передових технологій



обробки та аналізу даних для забезпечення максимальної надійності та точності в оцінці фізичного стану пілота.

У цьому контексті необхідно розглянути ключові аспекти передпольотного контролю, включаючи використання нейронних мереж для обробки медичних даних, високий рівень аналізу для виявлення аномалій, візуалізацію результатів за допомогою графіків, аналіз показників та надання рекомендацій щодо подальшого контролю або обмежень. Також доцільно визначити функціональність передпольотного контролю і те, як система працює для забезпечення безпеки та ефективності авіаційних процесів.

Нейронна мережа виступає як потужний інструмент для аналізу великого обсягу різноманітних медичних даних пілотів, забезпечуючи ефективну та комплексну обробку інформації. Детальніше розглянемо цей процес:

- збір інформації. Система отримує різноманітні дані від пілотів, включаючи результати лабораторних аналізів, історії захворювань, інформацію про фізичний стан, дані з медичних сенсорів та інші параметри;

- структурування та нормалізація. Отримані дані проходять процес структурування та нормалізації для уніфікації формату та забезпечення їхньої консистентності;

- використання нейронної мережі. Нейронна мережа використовується для аналізу цього різноманітного обсягу даних (рис.1.3). Мережа визначає патерни та взаємозв'язки між різними медичними параметрами з використанням *Amazon EMR* – керована кластерна платформа для обробки й аналізу величезних обсягів даних, враховуючи їхні взаємодії;

- пошук аномалій. Система автоматично виявляє аномалії або несподівані відхилення від типових патернів, що може свідчити про потенційні проблеми зі здоров'ям пілота;

- генерація зведеної інформації. На основі результатів нейронного аналізу формується зведена інформація, яка включає ключові висновки та показники стану здоров'я пілота;



– комплексний погляд на здоров'я пілота. Отриманий комплексний аналіз дозволяє системі визначити загальний стан здоров'я пілота, враховуючи всі аспекти його медичного профілю.

В результаті цього процесу система забезпечує інтегрований та глибокий аналіз медичних даних пілотів, дозволяючи здійснювати передпольотний контроль з високою ефективністю та точністю.

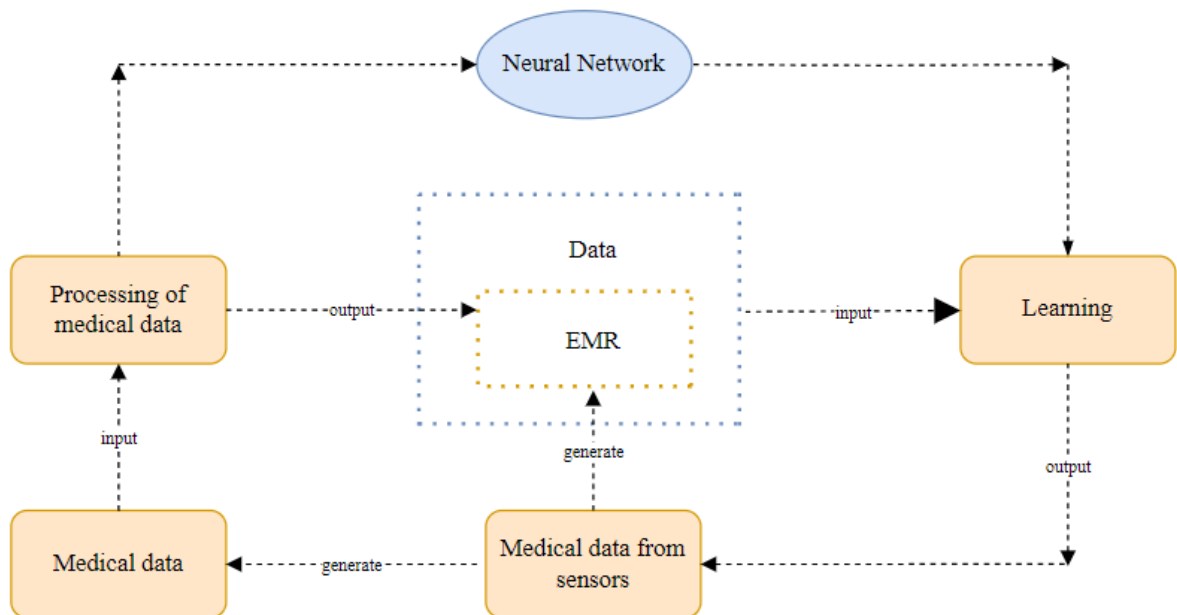


Рис. 1.3. Функціональна схема використання нейронних мереж для аналізу

Збір інформації в рамках передпольотного контролю є важливим етапом, що передбачає комплексний збір та узагальнення різноманітних даних з різних джерел. Основною метою цього етапу є створення повного та об'єктивного медичного образу пілота. Розглянемо цей процес докладніше:

– система отримує дані лабораторних досліджень, такі як аналіз крові, аналіз сечі, біохімічні показники та інші. Наприклад, рівень цукру в крові, кількість (щільність) елементів крові тощо;

– зібрана інформація про медичну історію пілота, включаючи раніше перенесені хвороби, хірургічні втручання, алергії та інші медичні події;

– дані щодо фізичного стану, такі як зріст, вага, артеріальний тиск, пульс, зорові та слухові показники, а також інші параметри, які характеризують фізичну форму пілота;



– інформація, зібрана з носимих пристроїв або сенсорів, таких як фітнес-трекери, які можуть включати в себе дані про рівень активності, якість сну, серцевий ритм тощо;

– інші параметри: збір інших медичних параметрів, таких як рівень кисню в крові, показники дихальної системи, терморегуляція та інші, в залежності від специфіки передпольотного аналізу.

Наприклад, якщо пілот має історію серцевих захворювань, система зосереджується на медичних параметрах, пов'язаних із серцево-судинною системою, таких як артеріальний тиск, пульс, електрокардіограма. Також враховуються дані з аналізів крові, які можуть вказувати на ризик серцевих ускладнень. Усі ці дані інтегруються для створення повного здоров'я пілота.

Після збору інформації, отримані дані піддаються процесам структурування та нормалізації. Ці етапи необхідні для того, щоб забезпечити єдність формату та консистентність даних, що є важливим для подальшого аналізу та використання. Розглянемо цей процес докладніше:

– структурування даних. Дані організовуються у визначений та зрозумілий спосіб, щоб забезпечити їхню систематизацію та легкий доступ для подальшого обробки. Наприклад, утворення таблиць чи баз даних для різних типів інформації;

– нормалізація. Забезпечення єднання структури даних шляхом приведення їх до стандартного формату. Це може включати усунення дублікатів, виправлення помилок, а також приведення однакових типів даних до єдиного стандарту;

– уніфікація формату. Всі дані приводяться до єдиного формату для забезпечення їхньої спільності та зручності в подальшому використанні. Наприклад, приведення дат до єдиного стандарту, використання однакових одиниць вимірювання тощо.

Припустимо, що дані про фізичний стан пілота були зібрані з різних джерел і вони мають різний формат та структуру. Система проводить структурування, вирівнює дані за певними параметрами, такими як вага, зріст, артеріальний тиск, інтегруючи їх у єдиний набір даних.



Після цього застосовується нормалізація, щоб усунути можливі дублікати, виправити помилки та привести всі числові значення до єдиного стандарту. Це забезпечує консистентність та точність даних перед подальшим аналізом системою передпольотного контролю.

Після структурування та нормалізації даних, система використовує навчену нейронну мережу для аналізу цього різноманітного обсягу інформації. Розглянемо цей процес:

- система використовує попередньо навчену нейронну мережу, яка проходила через етапи навчання на попередніх даних. Навчання включає в себе подання мережі великої кількості різноманітних даних для визначення складних взаємозв'язків та патернів;

- нейронна мережа визначає складні патерни та взаємозв'язки між різними медичними параметрами. Наприклад, вона може виявляти, як зміни в показниках артеріального тиску впливають на інші показники, такі як серцевий ритм чи рівень кисню в крові;

- мережа враховує взаємодії між різними параметрами, дозволяючи системі здійснювати більш точний та комплексний аналіз стану здоров'я пілота. Наприклад, вона може виявити, як вплив стану фізичного здоров'я на результати лабораторних аналізів.

Якщо у навчанні мережі використовувалися дані про пілотів з різними захворюваннями та фізичними характеристиками, нейронна мережа може автоматично визначити, які показники є ключовими при виявленні конкретних станів здоров'я. Наприклад, вона може виявити, що певні комбінації показників свідчать про підвищений ризик певних захворювань чи можуть вказувати на загрозу для безпеки політів.

Після аналізу медичних даних за допомогою нейронної мережі, система автоматично виявляє аномалії або несподівані відхилення від типових патернів. Розглянемо цей процес детальніше:

- система використовує ретельно налаштовані алгоритми для автоматичного виявлення аномалій в медичних даних. Ці алгоритми порівнюють поточні дані з



попередньо вивченими зразками і визначають, якщо спостерігається надто значуще відхилення;

- система має базу даних типових патернів та нормальних значень для різних параметрів здоров'я. Вона використовує цю інформацію для порівняння з поточними даними та визначення, чи вони входять в межі норми;

- якщо система виявляє аномалію, вона генерує сигнали або повідомлення, які вказують на можливі проблеми зі здоров'ям пілота. Наприклад, велике відхилення від норми в показниках може свідчити про несподівані стани чи ризики.

Наприклад, якщо у пілота різко зросла частота серцевих скорочень або знизилася концентрація кисню в крові, система може вважати це аномалією та визначити, що ці відхилення від типових значень можуть бути ознакою негайного стану чи потенційної загрози для безпеки польоту.

Після виявлення аномалій в медичних даних, система генерує зведену інформацію на основі результатів нейронного аналізу. Розглянемо цей процес більш детально:

- система проводить аналіз результатів, отриманих під час нейронного аналізу медичних даних. Цей аналіз включає в себе інтерпретацію виявлених аномалій, патернів та взаємозв'язків між різними параметрами здоров'я;

- на основі результатів аналізу формуються ключові висновки, які вказують на стан здоров'я пілота. Ці висновки можуть включати в себе виявлені медичні проблеми, тенденції та прогнози;

- система об'єднує ключові висновки та показники стану здоров'я, формуючи зведену інформацію. Це може бути подано у формі звіту, документу, тощо;

- система враховує контекст інформації, враховуючи історію здоров'я пілота та інші фактори, що можуть впливати на його стан.

Наприклад, якщо система виявила, що пілот має високий рівень стресу та низький рівень кисню в крові, зведена інформація може включати в себе висновок, що ці фактори можуть призвести до зниження фізичної витривалості та загрози для безпеки польоту.

Розглянемо комплексний погляд на здоров'я пілота:



– після отримання результатів нейронного аналізу та генерації зведеної інформації, система проводить загальний огляд аналізу, враховуючи всі виявлені аспекти та показники;

– система інтегрує всі виявлені медичні проблеми, тенденції та показники, враховуючи інформацію з різних джерел медичного профілю пілота;

– аналізується вплив різних факторів на стан здоров'я пілота. Наприклад, якщо аналіз показав зниження фізичної витривалості, система може розглядати можливі причини, такі як стрес, недостатній відпочинок або фізичні травми;

– на основі інтеграції всієї інформації формується комплексний погляд на здоров'я пілота. Це включає в себе оцінку загального стану здоров'я та можливих ризиків.

Наприклад, якщо система виявила, що пілот має високий рівень стресу, низький рівень кисню в крові, а також історію серцевих захворювань, комплексний погляд може вказувати на загрозу для серцево-судинної системи та рекомендації щодо подальших медичних обстежень та обмежень в польотах.

Використовуючи нейронні мережі та високий рівень аналізу, система ефективно збирає, обробляє та визначає стан здоров'я пілотів. Візуалізація даних, глибокий аналіз показників та комплексний огляд дозволяють системі генерувати точні результати та надавати обґрунтовані рекомендації. Такий підхід забезпечує високий рівень ефективності та точності в передпольотному контролі стану пілотів.

1.4. Виявлення фізичних та психологічних проблем пілота

Виявлення фізичних та психологічних проблем пілота є критично важливим процесом для забезпечення безпеки авіаційної діяльності. Пілоти, які керують повітряними суднами, мають бути в найкращому фізичному і психічному стані, оскільки вони несуть відповідальність за безпеку пасажирів і екіпажу, а також за безпеку самого літака. Виявлення та вирішення будь-яких фізичних або психологічних проблем може мати серйозний вплив на здатність пілота виконувати свої обов'язки та приймати рішення під час польоту.



Фізичні проблеми можуть включати в себе різноманітні медичні стани, травми або обмеження, які можуть впливати на фізичну здатність пілота керувати літаком. Деякі з основних фізичних проблем, які важливо виявити, включають:

- серцево-судинні захворювання;
- проблеми з дихальною системою;
- проблеми з органами зору та слуху;
- невідповідна фізична активність;
- медикаментозні препарати.

Хвороби серця, такі як артеріальна гіпертензія, стенокардія, аритмії та інші серцево-судинні захворювання, можуть підвищити ризик серйозних подій під час польоту [24].

Пілоти повинні мати нормальну функцію легень та дихальних шляхів, оскільки під час польоту вони стикаються з низьким рівнем кисню та можливими проблемами з тиском. Процедури, такі як спірометрія, допомагають виявити обмеження функції легень.

Гарний зір та слух є важливими для забезпечення безпеки польоту. Очікування та реагування на важливі сигнали на борту літака вимагає доброго зору та слуху, а регулярні перевірки гостроти зору та слуху допомагають виявити проблеми.

Фізичний стан пілота має відповідати вимогам польоту. Занадто низька фізична активність або занадто висока маса тіла можуть призвести до втоми та зниження реакційної здатності. Деякі ліки можуть впливати на фізичну та психічну здатність пілота. Важливо перевірити, чи має пілот дозвіл на приймання певних медикаментів та як вони можуть впливати на його здатність керувати літаком.

Психологічний стан пілота є не менш важливим аспектом безпеки польоту. Стани тривоги, депресії, стресу, а також інші психологічні проблеми можуть впливати на прийняття рішень та реакцію на стресові ситуації під час польоту. Деякі з психологічних проблем, які важливо виявити, включають:

- пілоти можуть стикатися зі стресом і тривожністю внаслідок надмірної відповідальності, непередбачуваних ситуацій або втоми. Ці стани можуть впливати на ясність розуму та прийняття рішень;



– психічні розлади, такі як депресія, біполярний розлад або шизофренія, можуть мати серйозний вплив на психічний стан пілота та здатність керувати літаком;

– використання алкоголю та наркотиків є неприпустимим під час польоту. Психоактивні речовини можуть впливати на реакцію, мислення та прийняття рішень;

– недостатній сон та перевтома можуть призвести до зниження уваги, концентрації та швидкості реакції.

Для виявлення фізичних та психологічних проблем пілота застосовуються різні методи та інструменти. Ось кілька з них:

- медичний огляд;
- психологічний огляд;
- анкети та інтерв'ю;
- аналіз інцидентів і аварій;
- спостереження за поведінкою.

Перед кожним польотом пілоти проходять медичний огляд, який включає в себе фізичний огляд, лабораторні та інструментальні дослідження. Цей огляд допомагає виявити фізичні проблеми. А психологічні тести та спеціальні оцінки можуть виявити психологічні стани, такі як тривога, депресивні стани, стрес і інші розлади [9].

Під час психологічного огляду пілотам можуть задавати питання та проводити інтерв'ю для оцінки їхнього психічного стану та емоцій.

Вагомим фактором є вивчення подій, аварій або інцидентів, пов'язаних з пілотами, може виявити ряд чинників, які впливають на їхню діяльність та безпеку. Оскільки психологічні стани можуть бути виявлені через спостереження за поведінкою пілота в різних ситуаціях.

Якщо виявлені фізичні або психологічні проблеми пілота, існують певні правила та процедури для їхнього вирішення:

- медична консультація та лікування;
- психологічна підтримка та консультування;



- тимчасова зупинка літання;
- підготовка до повернення.

У разі виявлення фізичних проблем, пілоти можуть бути направлені на медичну консультацію та лікування. Лікарі можуть надавати рекомендації щодо подальших дій та можливого обмеження у виконанні обов'язків пілота. Психологічні проблеми можуть вимагати психологічної підтримки та консультування. Пілотам можуть надавати психологічну допомогу для управління стресом та іншими проблемами.

Виявлення фізичних та психологічних проблем пілота є невід'ємною частиною безпеки в авіації. Ретельний медичний і психологічний огляд, який включає в себе різні методи та інструменти, дозволяє виявити проблеми та приймати вчасні заходи для їх вирішення. Такий підхід сприяє підтриманню безпеки в авіації та запобіганню можливим небезпекам і аваріям [8].

1.5. Висновки до розділу

В даному розділі були розглянуті теоретичні аспекти передпольотного контролю стану пілота, які мають надзвичайно важливе значення для безпеки авіаційної діяльності. Детально розглянуті аспекти включають в себе психофізіологічний стан пілота, виявлення фізичних та психологічних проблем, а також медичний огляд.

Проаналізовано та описано принципи роботи існуючих рішень проблеми передпольотного контролю стану пілота. Ці системи аналізують фізичні показники, такі як серцевий ритм, температура тіла та інші, для оцінки стану пілота. Виявлено їхні переваги та недоліки. Існуючі технології передпольотного контролю стану пілота постійно вдосконалюються, а впровадження нових рішень вимагає збалансованого підходу між безпекою, ефективністю та витратами.

Психофізіологічний стан пілота є надзвичайно складним явищем, яке включає в себе фізичні та психічні аспекти його здоров'я та емоційного стану. Оптимальний психофізіологічний стан є ключем до ефективності та безпеки в авіаційній



діяльності. Виявлення та оцінка цього стану виконується за допомогою різноманітних методів, таких як психофізіологічні тести, оцінка рівня стресу та втоми, а також спостереження за психофізіологічними реакціями пілота під час польотів. Фізичні та психологічні проблеми пілота можуть суттєво впливати на його здатність керувати літаком і приймати рішення під час польоту. Отже, виявлення цих проблем є ключовим елементом передпольотного контролю.

Медичний огляд є важливою частиною передпольотного контролю та забезпечує безпеку на борту літака. Виявлення фізичних обмежень, хвороби серця, порушень дихання та інших медичних проблем є надзвичайно важливим завданням для лікарів і авіаційних властей. Лікарі проводять ретельні обстеження, вимірюють артеріальний тиск, серцевий ритм та інші параметри для визначення здатності пілота до літання [7]. Виявлення фізичних та психологічних проблем, а також психофізіологічний моніторинг пілотів допомагають забезпечити їхню здатність до безпечного керування літаком і зменшити ризик нещасних випадків в авіації.

Підсумовуючи, можна зробити висновок, що передпольотний контроль стану пілота є невід'ємною частиною безпеки авіаційної діяльності. Існуючі рішення передпольотного контролю стану вимагають ряд покращень та модифікацій у зв'язку з постійним та швидким темпом розвитку технологій.

Тому розробка програмного засобу, що контролює передпольотний стан пілота з урахуванням психологічних та фізіологічних зчитаних даних є актуальною та своєчасною для розвитку авіаційної індустрії та покращення послуг. Важливою перевагою засобу є підвищення рівня безпеки для екіпажу та пілота, зручне використання як медичними працівниками так і пілотами, зменшення ризику впливу людського фактору на процес аналізу та збільшення ефективності повітряних польотів.



РОЗДІЛ 2

ВИМОГИ ДО СИСТЕМИ ПЕРЕДПОЛЬОТНОГО КОНТРОЛЮ

2.1. Вимоги до системи

Вибір медичної апаратури для передпольотного контролю стану базується на визначених стандартах та вимогах, викладених у офіційному документі «Про затвердження Правил медичного забезпечення польотів державної авіації України» [16]. Цей документ встановлює норми та процедури, які регулюють використання медичної техніки та обладнання для забезпечення безпеки та ефективності польотів.

Програмне забезпечення також грає ключову роль у системі передпольотного контролю стану пілота, оскільки воно відповідає за збір, аналіз та збереження даних, а також визначення результатів контролю.

Вимоги до програмного забезпечення повинні враховувати такі аспекти:

- надійність і стійкість;
- конфіденційність даних;
- автоматизація процесу обробки даних;
- інтеграція з іншими системами;
- забезпечення архівування даних;
- моніторинг та нагляд.

Програмне забезпечення повинне бути надійним та стійким до помилок для уникнення спотворення результатів контролю. Так як, вимоги повинні передбачати високий рівень конфіденційності даних пілота, оскільки це особиста та медична інформація.

Програмне забезпечення має спрощувати обробку даних та автоматично визначати результати контролю для здатності інтеграції з іншими системами авіаційної безпеки для обміну інформацією та подальшого аналізу.

Вхідні дані для аналізу стану, які рекомендуються для включення в систему передпольотного контролю (табл. 2.1).



Вхідні дані для аналізу стану

| Категорія | Параметр | Опис |
|------------------------|--------------------------------|---|
| Фізичні параметри | Серцевий ритм | Міра діяльності серця (рівень фізичного зусилля та стресу) |
| Фізичні параметри | Кров'яний тиск | Вимірювання артеріального тиску, показник фізичного стану |
| Фізичні параметри | Температура тіла | Вимірювання температури тіла |
| Фізичні параметри | Рівень кисню в крові | Моніторинг кисневого насичення для визначення рівня оксигенізації |
| Психологічні параметри | Рівень стресу | Аналіз біомаркерів стресу (кортизол) для оцінки психологічного стану |
| Психологічні параметри | Рівень адреналіну | Кількості адреналіну в крові (індикатор рівня збудження) |
| Психологічні параметри | Електроенцефалографія | Вимірювання електричної активності мозку для оцінки когнітивних функцій та емоцій |
| Когнітивні параметри | Рівень уваги | Вимірювання концентрації та уваги пілота під час роботи. |
| Когнітивні параметри | Швидкість реакції | Визначення часу реакції на зовнішні подразники. |
| Когнітивні параметри | Сприйняття інформації | Оцінка здатності пілота сприймати інформацію |
| Візуальні параметри | Параметри зору | Оцінка остроти зору, кольоророзрізнення |
| Візуальні параметри | Реакція на світлові подразники | Аналіз реакції на зміни світла та інших візуальних подразників |



Також, програмне забезпечення може включати можливості моніторингу та нагляду за станом пілота під час контролю.

Впровадження системи передпольотного контролю стану пілота передбачає визначення і дотримання технічних стандартів, які регулюють розробку та виробництво обладнання та програмного забезпечення. Ці технічні стандарти включають вимоги, які мають бути виконані для забезпечення якості та надійності системи передпольотного контролю стану пілота:

- калібровка та перевірка обладнання;
- стандарти безпеки;
- стандарти обробки та збереження даних;
- системи контролю якості;
- сертифікація продукції;
- сумісність із сучасними технологіями.

Технічні стандарти повинні визначати процедури калібрування та перевірки медичних приладів. Це необхідно для того, щоб забезпечити точність вимірювань та перевірити, що обладнання функціонує належним чином. Технічні стандарти також повинні визначати правила обробки та збереження медичних даних. Це необхідно для забезпечення цілісності та конфіденційності цих даних. А вимоги до стандартів безпеки мають на меті гарантувати, що обладнання та програмне забезпечення відповідають нормам безпеки. Це допомагає уникнути можливих ризиків для пілотів та забезпечити їхню безпеку під час використання системи передпольотного контролю.

Вимоги до стандартів можуть передбачати системи контролю якості, які перевіряють відповідність обладнання та програмного забезпечення встановленим стандартам, що допомагає гарантувати те, що система передпольотного контролю працює належним чином та надійно. Більш того, деякі технічні стандарти можуть вимагати процедури сертифікації продукції перед введенням її в експлуатацію, що допомагає забезпечити те, що обладнання та програмне забезпечення відповідають встановленим вимогам та стандартам якості. Додатково, важливо враховувати можливість оновлення та сумісності обладнання та програмного забезпечення із



сучасними технологіями. Це допомагає підтримувати систему передпольотного контролю в актуальному стані та готовою до майбутнього розвитку.

Дотримання цих вимог до технічних стандартів є обов'язковим для забезпечення ефективності, безпеки та надійності системи передпольотного контролю стану пілота.

2.2. Вибір інструментів та технологій

Для аналізу медичних даних та розробки програмного засобу передпольотного контролю стану пілота використовуються різноманітні бібліотеки та інструменти. Ось перелік деяких з них, які допомагають в ефективному обробленні та аналізі даних.

Pandas – це потужна бібліотека для аналізу та маніпуляції даних в табличному форматі. *Pandas* ідеально підходить для обробки та аналізу медичних даних, оскільки вони часто подаються у вигляді таблиць. Його зручний інтерфейс дозволяє ефективно взаємодіяти з даними та виконувати складні операції [18].

NumPy та *SciPy* – це бібліотеки для виконання математичних операцій та наукових обчислень. Вони надають широкий спектр функцій для статистичного аналізу, лінійної алгебри та обробки сигналів, що є важливим для обробки медичних даних.

TensorFlow та *PyTorch* – це фреймворки для машинного навчання та глибинного навчання. Вони дозволяють створювати, навчати та впроваджувати нейронні мережі та інші моделі глибинного навчання. Вибір між *TensorFlow* та *PyTorch* може залежати від конкретних вимог та вподобань розробника. *TensorFlow* від *Google* широко використовується у великих проектах, в той час як *PyTorch* від *Facebook* володіє простішим та більш гнучким інтерфейсом.

Amazon DynamoDB – це повністю керована *NoSQL* база даних від *Amazon Web Services*. Вона розроблена для масштабованих та високодоступних додатків з надійністю та низькою затримкою. *DynamoDB* є ідеальним вибором для систем передпольотного контролю через свою здатність масштабуватися відносно великої



кількості даних, швидкості доступу та гнучкості сховища даних. Вона дозволяє зберігати та отримувати дані, що важливо для медичних записів, забезпечуючи високий рівень надійності та продуктивності [17]. Приклад використання користувачем створеної бази даних *Amazon DynamoDB* наведено на рис. 2.1.

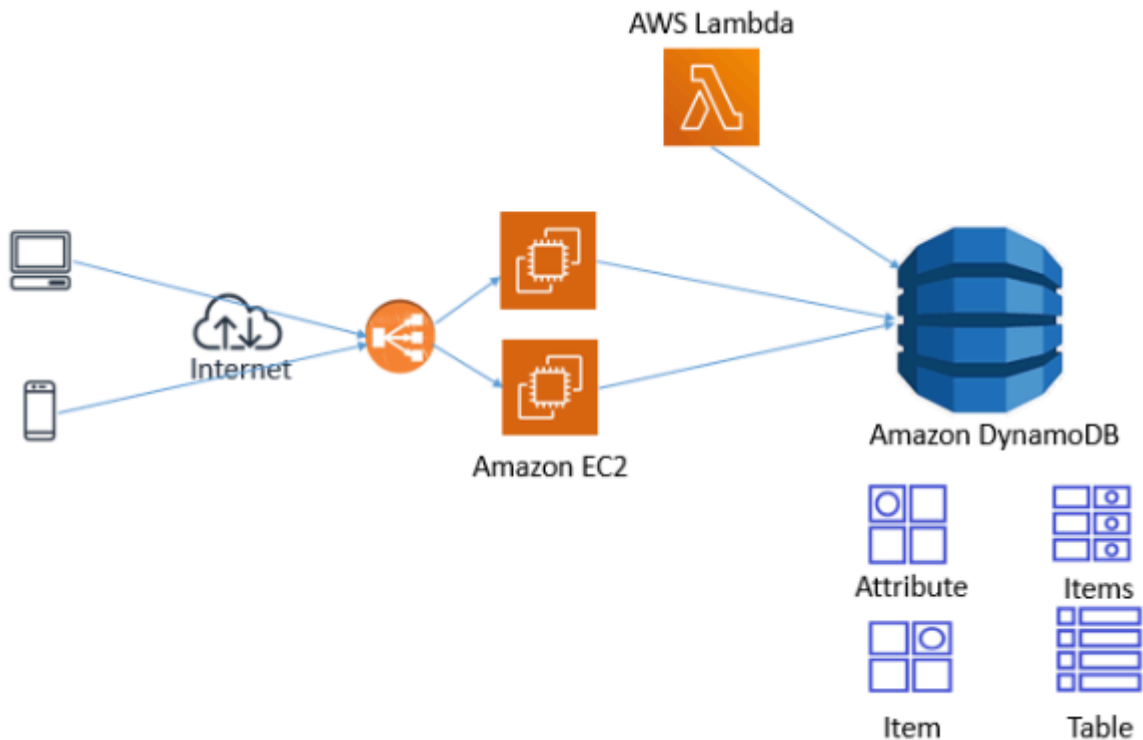


Рис. 2.1. Приклад використання *Amazon DynamoDB*

React – це бібліотека для створення динамічного та відзивчивого веб-інтерфейсу. Дозволяє створювати модерний та зручний інтерфейс для взаємодії з системою.

Prometheus – це система для моніторингу та збору метрик систем. Забезпечує зручний моніторинг функціональності та продуктивності програмного засобу.

NLTK та *SpaCy* – це бібліотеки для обробки та аналізу текстової інформації. Використовуються для роботи з медичними текстами та забезпечення точного аналізу.

Обрані бібліотеки та інструменти спрямовані на забезпечення високої якості та ефективності розробки, а також на високий рівень надійності та безпеки системи передпольотного контролю стану пілота.

2.3. Вибір мови програмування та інструментів для розробки

Для розробки програмного засобу передпольотного контролю стану пілота була вибрана комбінація різних мов програмування та інструментів, спрямована на забезпечення високої продуктивності, швидкості розробки та надійності системи. Основні технології, які увійшли в стек, включили в себе багато засобів для розробки.

JavaScript – використовується для розробки фронтенд-частини веб-додатків, дозволяючи створювати інтерактивний та відзвучивий веб-інтерфейс для користувачів системи передпольотного контролю пілота. Використання *JavaScript* дозволяє розробникам створювати функціональність, що реагує на дії користувача, такі як натискання кнопок та взаємодія з формами.

Python – обрано для бекенд-розробки програми передпольотного контролю. Ця мова програмування визначається своєю простотою та широким спектром бібліотек для обробки даних. *Python* дозволяє легко працювати з медичними даними, аналізувати їх та виконувати обчислення для системи передпольотного контролю. Має багатий набір функцій, які роблять його потужною мовою для розробки складних програм. Він підтримує багатопотоковість, мережу та підключення до баз даних, що робить його ідеальним вибором для створення програм корпоративного рівня [26].

CSS та *SCSS* – *CSS* використовується для стилізації веб-інтерфейсу та надання йому привабливого та професійного вигляду. Розширення *SCSS* дозволяє створювати стилі більш організовано та полегшує розробку та збереження консистентного стилізованого вигляду.

TypeScript – використовується разом із *JavaScript* для створення фронтенд-додатків, додаючи статичну типізацію та полегшуючи уникнення помилок на етапі розробки. Використання *TypeScript* підвищує продуктивність розробників та сприяє підтримці великих проектів.

HTML – використовується для створення структури веб-сторінок та розмітки, визначаючи структуру документа та його вміст. Використання *HTML* дозволяє



створювати сторінки з різноманітними елементами, які об'єднуються в єдиний користувацький інтерфейс системи передпольотного контролю.

Ці мови програмування та інструменти взаємодіють для створення повноцінної системи передпольотного контролю стану пілота, допомагаючи забезпечити високий рівень функціональності та ефективності для потреб медичної галузі.

2.4. Обґрунтування вибору технологій та бібліотек для реалізації

Комбінація вибраних бібліотек та інструментів у поєднанні з обраною хмарною інфраструктурою *AWS* та іншими сервісами надає ряд важливих переваг для реалізації системи передпольотного контролю стану пілота (рис. 2.2).

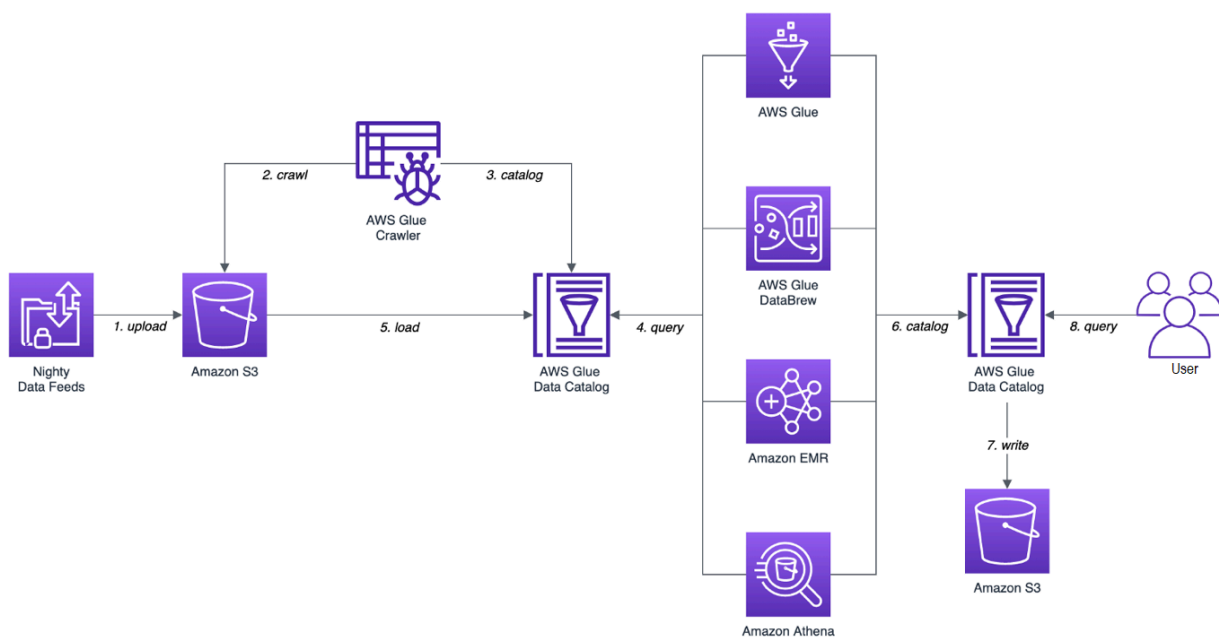


Рис. 2.2. Діаграма роботи обраних *AWS* сервісів

AWS Glue та *Apache Spark* – використання *AWS Glue* для автоматизованого екстракції та трансформації даних з різних джерел. *Apache Spark* використовується для паралельної обробки великих обсягів даних, забезпечуючи швидкий та ефективний аналіз [19].

Medical Analysis System (MAS) – вирізняється рядом інноваційних підходів у сфері аналізу та обробки медичних даних. Використовуючи службу *Amazon*

Comprehend Medical, система виявляє ключові медичні компоненти та точно ідентифікує ризикові чинники. Особливий акцент робиться на високій точності та надійності отриманих результатів.

На початковому етапі *AWS Data Pipeline* автоматично отримує медичні дані з визначеного джерела – *Amazon S3*. Система використовує розроблені алгоритми для індивідуальної обробки медичних записів кожного пілота, виділяючи ключові медичні параметри та ризикові фактори. На етапі моделювання ризиків використовується імовірнісна модель, розроблена для програми. *AWS Data Pipeline* автоматично викликає цей модуль, використовуючи дані, що пройшли попередні кроки аналізу. Результати аналізу, такі як ідентифіковані ризики захворювань, зберігаються відповідно до визначеної структури даних – базі даних *Amazon DynamoDB*. *AWS Data Pipeline* допомагає автоматизувати цей процес. Таким чином, впровадження індивідуально розроблених кроків аналізу даних у контексті *AWS Data Pipeline* дозволяє системі ефективно використовувати потужності хмарної інфраструктури для точного та персоналізованого аналізу медичних даних пілотів.

Щоб забезпечити надійність та продуктивність, система використовує *AWS CloudWatch* для моніторингу ключових параметрів, таких як час виконання, витрати ресурсів і точність моделі.

Для розробки та аналізу використовується *AWS Quicksight* для візуалізації інформації, отриману в результаті аналізу *MAS*.

За допомогою *AWS SDK for JavaScript* встановлено взаємодію із спеціалізованими сервісами *AWS*, використовуючи розроблені методи взаємодії, спрямовані на підтримку унікальної функціональності системи *MAS*.

AWS Glue DataBrew використовується для ефективного вирішення завдань візуальної підготовки та очищення даних перед подальшим аналізом. Це включає в себе визначення правил очищення даних, об'єднання таблиць або нормалізацію інформації для однорідності даних у системі *MAS*.

Інтеграція *AWS CloudWatch* в систему дозволяє не тільки моніторити, але й автоматично реагувати на зміни в продуктивності *MAS*. Завдяки цьому налаштовано



тривалість виконання аналітичних операцій та встановлено автоматичні сповіщення при зменшенні продуктивності.

2.5. Висновки до розділу

Розділ 2 присвячений вибору технологій, структури та розгортанню програмного засобу передпольотного контролю стану пілота. В цьому розділі обґрунтовано важливість вибору правильних технологій для фронтенду та бекенду, а також розглянуто основні компоненти системи та їхню взаємодію.

Були визначені ключові вимоги до системи передпольотного контролю стану пілота, а також сформовано перелік необхідних вхідних параметрів для глибшого аналізу показників пілота. Ці вимоги та параметри визначають основу для розробки програмного засобу, що гарантує надійний трекінг фізичного та психологічного стану пілота перед виконанням польотів. Описаний підхід враховує особливості пілотської діяльності та забезпечує збір та обробку важливих даних для максимальної безпеки та оптимального функціонування повітряного транспорту.

Визначено, що використання *JavaScript*, *Python*, *CSS*, *SCSS*, *TypeScript* та *HTML* дозволяє створити потужний та інтерактивний веб-інтерфейс для користувачів системи передпольотного контролю. Також *AWS* вибрано як інфраструктуру для розгортання, оскільки вона надає надійність, масштабованість та безпеку.

Були розглянуті переваги використання таких сервісів *AWS*, як *Amazon Transcribe Medical* та *Amazon Comprehend Medical* для аналізу медичних даних, а також *DynamoDB* та *Amazon S3* для збереження даних. Також надано докладний опис структури програмного засобу передпольотного контролю та розглянуто, як різні компоненти системи взаємодіють між собою.

Загалом, розділ 2 допомагає зрозуміти, як вибір технологій та структура програмного засобу сприяють досягненню основних цілей системи та забезпечують її надійність, продуктивність та безпеку.



РОЗДІЛ 3

ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАСОБУ

3.1. Введення до проєктування

У цьому підрозділі ми розглянемо важливий етап проєктування програмного засобу, спрямованого на передпольотний контроль стану пілота. Відповідно до визначених мети та завдань, ми визначимо ключові аспекти, які потребують уваги в ході розробки програми передпольотного контролю.

Головною метою проєктування програмного засобу передпольотного контролю стану пілота є створення високоефективного та надійного інструмента для медичного обстеження пілотів перед польотом. Ця система повинна забезпечувати надійну обробку та аналіз медичних даних, щоб визначити стан пілота та його придатність до політів. Метою є також зменшення обсягу ручного адміністрування та паперової роботи з метою прискорення процесу передпольотного контролю та полегшення навантаження на медичних фахівців.

Проєктування системи передпольотного контролю включає в себе наступні основні завдання:

- 1) створення зручного та інтуїтивного інтерфейсу, який дозволить пілотам та медичним фахівцям взаємодіяти з системою для збору медичних даних;
- 2) розробка функціоналу для аналізу медичних даних, виявлення ключових медичних компонентів та ідентифікації ризикових чинників;
- 3) розробка системи безпеки, яка гарантує конфіденційність медичних даних пілотів та дотримання вимог щодо захисту особистої інформації;
- 4) взаємодія з різними хмарними службами та сервісами *Amazon Web Services (AWS)* для збереження та аналізу даних;
- 5) забезпечення можливості масштабування системи для обробки великих обсягів даних та забезпечення високої продуктивності.



У процесі проєктування програмного засобу передпольотного контролю стану пілота, ми будемо звертати увагу на такі ключові аспекти:

- зручність користування;
- автоматизація;
- безпека;
- масштабованість;
- інтеграція.

Інтерфейс системи повинен бути легким для використання як пілотами, так і медичними фахівцями. Автоматизація системи має значно зменшити ручну роботу та спростити обробку медичних даних.

Захист конфіденційності та інтеграції медичних даних є важливим аспектом. Так як, система повинна бути готовою до масштабування для вирішення зростаючих потреб у передпольотному контролі та забезпечення зв'язку з існуючими сервісами та хмарними рішеннями для ефективного використання ресурсів.

3.2. Структура програмного засобу передпольотного стану

Програмний засіб передпольотного контролю стану пілота розроблений з використанням сучасних технологій та інструментів для забезпечення надійності, продуктивності та безпеки.

Структура даного програмного засобу відзначається наявністю різноманітних компонентів та модулів, що спрямовані на оптимізацію операцій з великим обсягом даних, що будуть надходити до програми з датчиків, та забезпечення зручного доступу до програми для лікарів та пілотів (рис. 3.1).

Для покращення продуктивності та забезпечення надійності фронтенду використовуються різні фреймворки та бібліотеки, такі як *React*, *Angular* або *Vue.js*. Ці фреймворки допомагають у створенні складних інтерфейсів та управлінні станом додатку. Користувачі взаємодіють з ПЗ через веб-браузери та інші клієнтські пристрої, де вони можуть виконувати різні дії та вводити медичні дані.



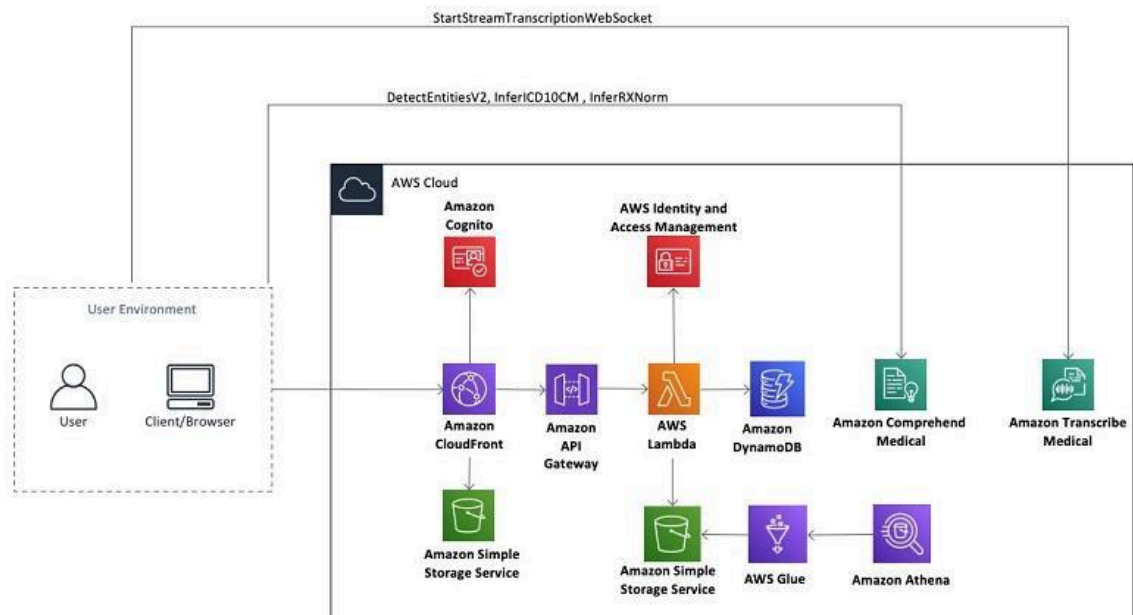


Рис. 3.1. Діаграма сполучення сервісів AWS

Бекенд-частина ПЗ розроблена з використанням мови програмування *Python*. *Python* вибраний через його популярність та здатність легко обробляти та аналізувати медичні дані.

У бекенді можуть використовуватися фреймворки та бібліотеки для створення *API*, обробки запитів та взаємодії з базою даних. Наприклад, *Django* або *Flask* для створення веб-сервісів.

Для зберігання медичних даних та інших важливих інформацій використовуються сервіси *Amazon DynamoDB* та *Amazon S3*. *DynamoDB* використовується як реляційна база даних для структурованих даних, а *S3* – для збереження та обміну файлами [20].

Для розробки та розгортання ПЗ використовується платформа *Amazon Web Services (AWS)* та середовище розробки *Visual Studio*. *AWS* надає потужні інструменти для створення, розгортання та масштабування додатків в хмарі.

AWS Identity and Access Management (IAM) – Для керування доступом до ресурсів *AWS* використовуються *Amazon Cognito* та *AWS IAM*. *Amazon Cognito* використовується для управління ідентифікацією користувачів, а *AWS IAM* – для налаштування прав доступу.

AWS Content Delivery Network – використовується для прискорення доставки вмісту ПЗ користувачам з різних регіонів світу.

Amazon API Gateway – використовується для створення та управління *API* для ПЗ, надаючи доступ до його функціональності через *HTTP* запити.

Amazon Comprehend Medical – Для видобуття та аналізу медичної інформації з неструктурованого тексту використовується сервіс *Amazon Comprehend Medical*. Цей сервіс допомагає витягти ключові дані, такі як медичні стани, ліки, дозування та інше, та пов'язати їх з медичними онтологіями.

Користувачі можуть зберігати сесії, позначати їх та аналізувати результати. Для збереження використовується *Amazon DynamoDB* та *Amazon S3*, а для аналізу – *Amazon Athena*.

Програмний засіб відкритий для розширення та інтеграції з іншими системами, такими як системи електронної медичної документації (*EHR*), додавання постійного зберігання даних, розбудова аналітики та інші розширення для поліпшення досвіду користувачів.

Ця структура програмного засобу передпольотного контролю стану використовує сучасні технології та хмарну інфраструктуру для надання користувачам медичних послуг надійних та продуктивних інструментів для обробки та аналізу медичних даних.

3.3. Архітектурний дизайн

У цьому розділі розглянемо загальну архітектурну концепцію системи передпольотного контролю стану пілота та обґрунтуємо вибір конкретних архітектурних рішень. А також докладно опишемо взаємодію між основними компонентами системи.

Архітектурний дизайн системи передпольотного контролю стану пілота базується на модульній структурі, що враховує різні аспекти системи та роль різних компонентів. Основною метою цієї архітектури є забезпечення високої ефективності та надійності системи, а також легкості розширення та підтримки.



Система передпольотного контролю складається з наступних ключових компонентів:

User interface (UI) – Цей компонент відповідає за взаємодію з пілотами та медичними фахівцями. Він забезпечує зручний та інтуїтивний інтерфейс для збору медичних даних та взаємодії з системою.

Medical Analysis System (MAS) – здійснює аналіз транскрипцій та інших медичних даних, виявляє ключові медичні компоненти та ідентифікує ризикові чинники. Вона використовує службу *Amazon Comprehend Medical* для цього.

Document Management System (DMS) – забезпечує збереження медичних даних та їх обробку. Вона використовує *Amazon DynamoDB* для збереження та організації даних та *Amazon S3*.

Security Information System (SIS) – забезпечує конфіденційність та безпеку медичних даних. Вона використовує служби *Amazon Cognito* та *AWS Identity and Access Management (IAM)* для керування доступом.

Хмарні послуги *AWS* – Ця складова відповідає за інтеграцію з хмарними сервісами *Amazon Web Services* для забезпечення масштабованості, продуктивності та доступності системи.

У даній системі використовується мікросервісна архітектура, де кожен компонент працює як окремий сервіс і має власну функціональність (рис. 3.2).

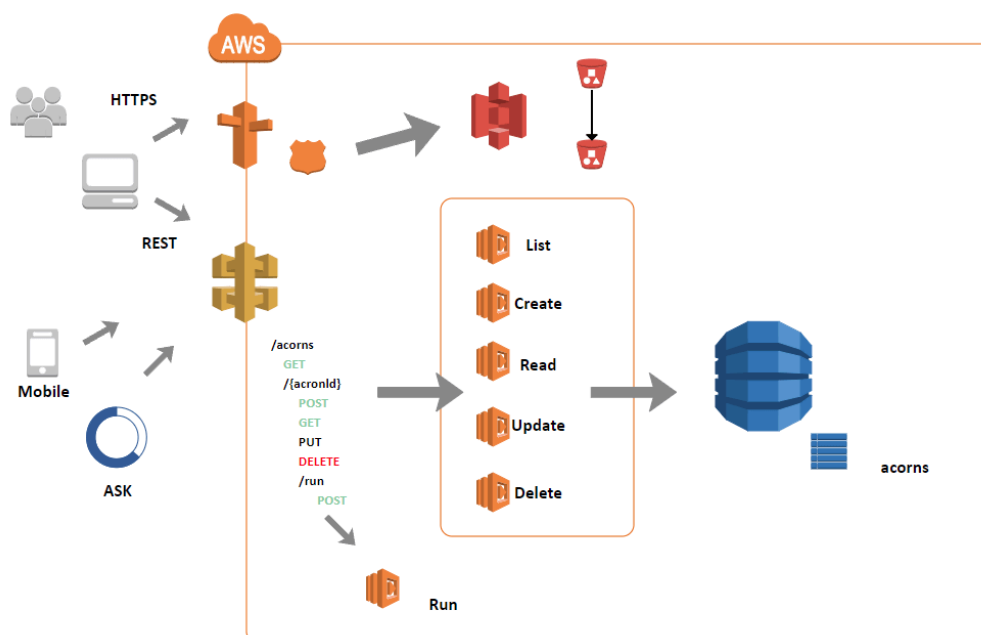


Рис. 3.2. Мікросервісна архітектура

Основні переваги цієї архітектури включають:

- масштабованість;
- легкість у розробці та підтримці;
- зменшення впливу помилок;
- легка заміна та оновлення;
- інтеграція *AWS*.

Кожен сервіс може бути масштабованим окремо, в залежності від навантаження, що дозволяє ефективно вирішувати зростаючі потреби та може бути незалежно розроблений і підтриманий, спрощуючи процеси розробки та обслуговування. Помилки в одному сервісі не впливають на інші, що підвищує надійність системи, а заміна або оновлення окремих сервісів стає набагато простішою, не впливаючи на решту системи.

Для підвищення масштабованості та надійності використовується хмарні сервіси *Amazon Web Services (AWS)*. Це дозволяє інтегрувати систему з такими рішеннями, як *Amazon DynamoDB* для збереження даних. Основні переваги використання *AWS* включають:

- масштабованість та продуктивність, оскільки *AWS* надає можливість ефективно масштабувати систему для вирішення зростаючих потреб;
- забезпечення надійності та доступності, так як *AWS* гарантує високий рівень надійності та доступності даних та сервісів;
- зменшення трудовитрат у наслідок використання хмарних рішень допомагає уникнути великих трудовитрат на підтримку фізичної інфраструктури.

Ця архітектура спроектована з урахуванням можливості розширення системи. Це означає, що можна ефективно впоратися зі зростаючими потребами у передпольотному контролі та обслуговувати більше користувачів. Можна додавати нові мікросервіси або масштабувати існуючі, щоб забезпечити високу продуктивність та якість обслуговування. Гнучкість такої архітектури дозволяє змінювати систему відповідно до змінюючихся потреб користувачів та ринку.



Взаємодія між ключовими компонентами системи передпольотного контролю стану пілота відбувається наступним чином:

- інтерфейс користувача (*UI*) надає засоби для взаємодії пілотів та медичних фахівців з системою. Він передає дані на систему та отримує аналіз медичних даних для відображення на інтерфейсі;

- система збереження та управління даними (*DMS*) відповідає за збереження аналізів, рекомендацій та інших медичних даних в *Amazon DynamoDB* та *Amazon S3*;

- система безпеки та ідентифікації (*SIS*) забезпечує конфіденційність медичних даних та керування доступом користувачів;

- хмарні послуги *AWS* використовуються для інтеграції з хмарними рішеннями та забезпечення масштабованості та продуктивності.

Зручність використання засобу пілотом, полягає в повній незалежності від лікаря, що означає те, що він самостійно може виконувати передпольотний аналіз за допомогою зчитаних даних з датчиків у результаті якого генерується звіт про рішення на дозвіл чи заборону польоту; переглядати власну історію включаючи всі зчитані данні та завантажені документи до системи (рис. 3.3).



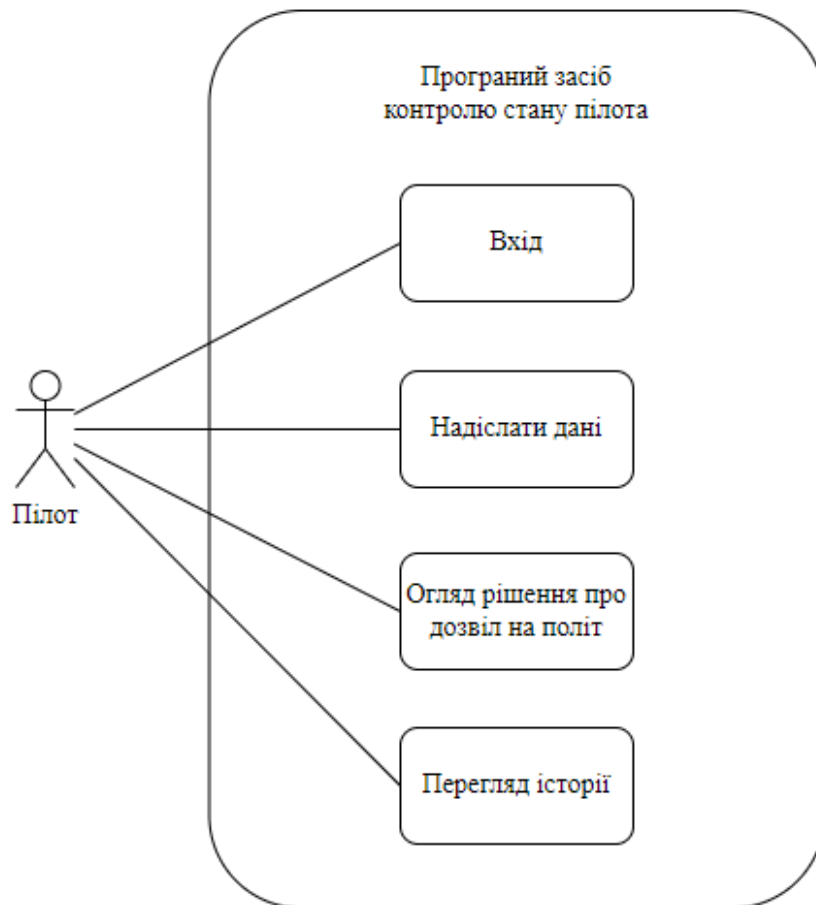


Рис. 3.3. Діаграма варіантів використання програмного засобу пілотом

В свою чергу лікарі мають більш розширені можливості використання програмного засобу: приймати участь у аналізі, надсилати персоналізовані рекомендації та змінювати історію пілота (рис. 3.4).



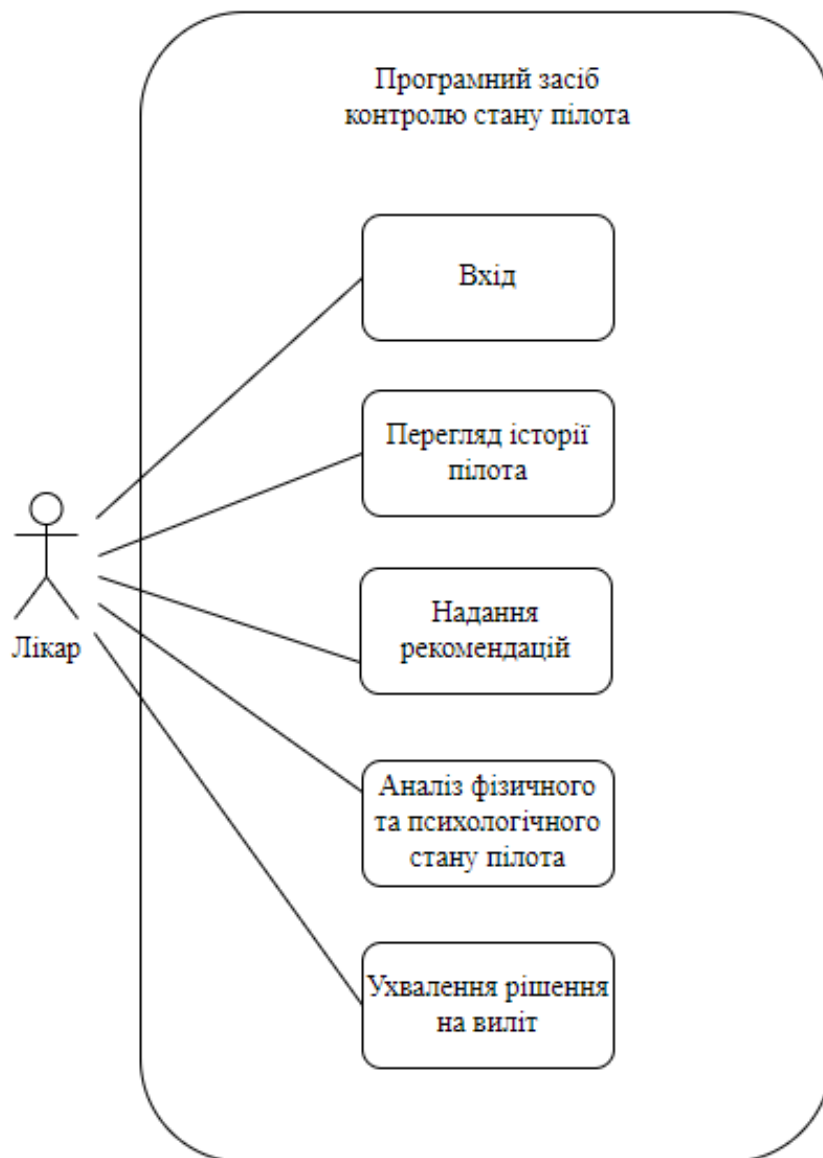


Рис. 3.4. Діаграма варіантів використання програмного засобу лікарем

За допомогою зручного інтерфейсу пілоти та лікарі мають швидкий доступ до засобу і основних функцій, що націлені на відображення та аналіз зчитаних даних, перегляд історії зареєстрованих пілотів, надання рекомендацій системою у наслідку аналізу показників (рис. 3.5).

Архітектурний дизайн системи передпольотного контролю стану пілота розрахований на те щоб забезпечити високу продуктивність та надійність обробки медичних даних перед польотом пілотів (рис. 3.6).

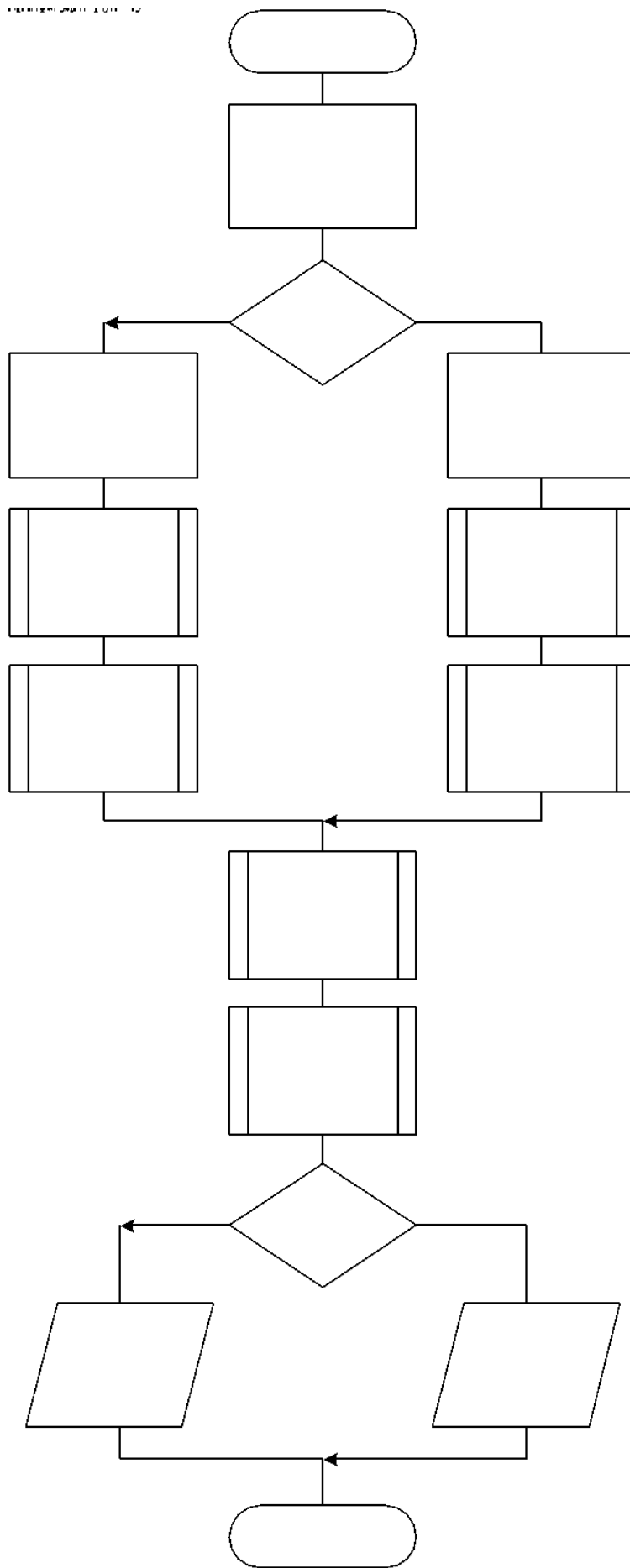


Рис. 3.5. Схема алгоритму роботи програмного засобу



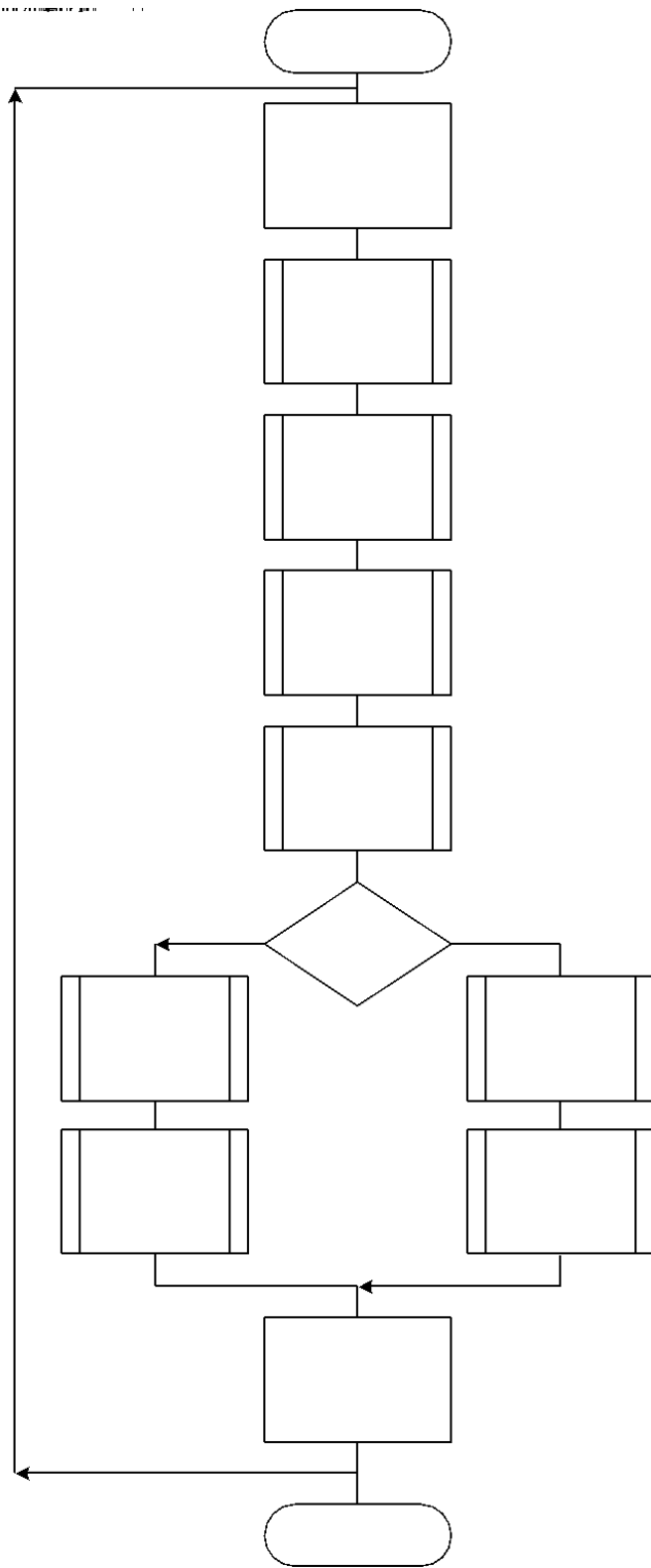


Рис. 3.6. Схема алгоритму роботи аналізатора

Опишемо загальний алгоритм роботи:

– отримання даних: медичні дані пілота, включаючи результати аналізів, фізичний стан та інші медичні параметри. *MAS* отримує дані з медичних джерел, таких як електронні медичні записи, датчики здоров'я;



– попередня обробка даних: виконується обробка даних для усунення шуму, заповнення відсутніх значень, стандартизації формату та підготовки даних для аналізу;

– аналіз медичних даних: *MAS* використовує алгоритми аналізу, включаючи машинне навчання для оцінки стану здоров'я пілота;

– генерація результату: *MAS* генерує оцінку загального стану пілота на основі виявлених факторів та медичних параметрів;

– прийняття рішення про дозвіл польотів – система аналізу встановлює критерії, які визначають, чи дозволено пілоту виконувати польоти, на основі оцінки його здоров'я;

– вивід рішення: рішення виводиться для користувача через інтерфейс користувача, а також може бути відправлено відповідним авіаційним органам та іншим зацікавленим сторонам.

3.4. Функціональні вимоги

Функціональні вимоги є ключовою частиною процесу розробки програмного засобу передпольотного контролю стану пілота, оскільки вони визначають, як програма повинна функціонувати з точки зору користувача та системи в цілому. Розглянемо основні функціональні вимоги:

– аналіз медичних даних: система повинна здійснювати аналіз медичних даних для визначення ключових медичних компонентів та ідентифікації ризикових чинників за допомогою служби *Amazon Comprehend Medical*. Результати аналізу повинні бути представлені користувачам у зручному форматі, який дозволить їм визначити медичний стан та придатність пілота до політів;

– збереження медичних даних: система має забезпечувати медичних консультацій та результатів аналізу. Для збереження даних використовуються служби *Amazon DynamoDB* та *Amazon S3*;

– безпека та ідентифікація: система повинна гарантувати конфіденційність медичних даних пілотів та дотримання вимог щодо захисту особистої інформації.



Для керування доступом та ідентифікації користувачів використовуються служби *Amazon Cognito* та *AWS Identity and Access Management (IAM)*;

– інтеграція з *Amazon Web Services*: система повинна взаємодіяти з хмарними сервісами *AWS* для забезпечення масштабованості, продуктивності та доступності. Взаємодія включає збереження даних в *Amazon S3*, а також аналіз збережених сесій за допомогою *Amazon Athena*;

– сесії: користувачі мають можливість зберігати сесії, позначати їх та пов'язувати з медичними фахівцями та пацієнтами. Збережені сесії можуть бути шукані та відновлені для подальшого використання;

– пошук та перегляд збережених сесій: користувачі можуть шукати збережені сесії за ідентифікаторами медичних фахівців, пацієнтів та сесій. Результати пошуку відображаються у табличному форматі, а користувачі можуть переглядати зведення сесій;

– аналітика через *Amazon Athena* та *Amazon Quicksight*: збережені сесії можуть бути аналізовані за допомогою *Amazon Athena*, яка надає можливість виконувати аналітичні запити до даних. Опційно може використовуватися *Amazon Quicksight* для створення інтерактивних панелей та звітів.

Зазначені функціональні вимоги визначають операції та можливості програмного засобу передпольотного контролю стану пілота та відповідають основним цілям даного проекту, оскільки саме вони надають основу для подальшого проектування засобу та розробки надійно-функціонуючої системи з урахуванням потреб користувачів.

3.5. Алгоритми аналізу фізичного та психологічного здоров'я пілотів

У цьому підрозділі надано детальний опис розроблених алгоритмів для програмного забезпечення передпольотного контролю стану пілота. Кожен алгоритм враховує унікальні характеристики пілота, що дозволяє проводити комплексний та індивідуалізований аналіз.



Алгоритм аналізу серцевого ритму та фізичної активності розроблено для глибокого розуміння та об'єктивного оцінювання фізичного стану пілота. Підставою для цього аналізу є постійний моніторинг серцевого ритму та реального вимірювання рівня фізичної активності за допомогою вбудованих датчиків. Датчики електрокардіограми фіксують електричну активність серця, а акселерометри та гіроскопи відстежують рухову активність пілота [21].

Частотний аналіз серцевого ритму, використовуючи методи частотного аналізу, визначає основні характеристики серцевого ритму, такі як частота та амплітуда [15].

Часовий аналіз виконує аналіз фізичної активності, вимірюванням інтервалів між серцевими скороченнями, що дозволяє отримати детальну інформацію про динаміку змін у ритмі серця [15]:

```
class PhysicalActivityAnalyzer:
```

```
    def __init__(self, accelerometer_data, gyroscope_data):
```

```
        self.accelerometer_data = accelerometer_data
```

```
        self.gyroscope_data = gyroscope_data
```

```
    def integrate_accelerometer_data(self):
```

```
        integrated_data = integrate_accelerometer(self.accelerometer_data)
```

```
        return integrated_data
```

```
    def classify_physical_activity(self, integrated_data):
```

```
        activity_type = classify_activity(integrated_data)
```

```
        return activity_type
```

```
accelerometer_data = get_accelerometer_data_from_sensor()
```

```
gyroscope_data = get_gyroscope_data_from_sensor()
```

```
activity_analyzer = PhysicalActivityAnalyzer(accelerometer_data, gyroscope_data)
```

```
integrated_data = activity_analyzer.integrate_accelerometer_data()
```

```
activity_type = activity_analyzer.classify_physical_activity(integrated_data)
```

Алгоритм відрізняється від існуючих тим, що враховує індивідуальні особливості серцево-судинної системи пілота. Заснований на природному серцевому ритмі, алгоритм враховує фізіологічних особливостей кожного пілота.



Алгоритм забезпечує негайний та високоякісний аналіз фізичного стану пілота, що дозволяє ефективно реагувати на будь-які зміни, пов'язані зі стресом або втомою. Застосування часового та частотного аналізу надає можливість детально вивчати динаміку серцевого ритму та рівень фізичної активності.

Алгоритм використовує сучасні методи обробки сигналів, включаючи фільтрацію, частотний та часовий аналіз. Враховуючи індивідуальні особливості, він забезпечує високу точність аналізу, а адаптивність робить його ефективним і надійним для визначення фізичного стану пілота в режимі реального часу.

Можливість взаємодії з іншими алгоритмами та системами контролю робить його ключовим елементом передпольотного контролю стану пілота, забезпечуючи безпеку та ефективність в авіаційних умовах.

Алгоритм для аналізу когнітивних параметрів пілота розроблено для комплексної оцінки рівня уваги, швидкості реакції та сприйняття інформації. Його етапи включають збір вхідних даних, використання стандартизованих тестів та моніторинг психічного навантаження за допомогою систем ЕЕГ:

```
class CognitiveAnalyzer:
```

```
    def __init__(self, attention_test, reaction_time_test, perception_test, eeg_data):
```

```
        self.attention_test = attention_test
```

```
        self.reaction_time_test = reaction_time_test
```

```
        self.perception_test = perception_test
```

```
        self.eeg_data = eeg_data
```

```
    def collect_input_data(self):
```

```
        input_data = {
```

```
            "attention": self.attention_test(),
```

```
            "reaction time": self.reaction_time_test(),
```

```
            "perception": self.perception_test(),
```

```
            "eeg_data": self.eeg_data
```

```
        }
```

```
        return input_data
```

```
    def normalize_data(self, input_data):
```



```

    normalized_data = some_normalization_function(input_data)
    return normalized_data

def cognitive_profile(self, normalized_data):
    cognitive_profile = create_profile(normalized_data)
    return cognitive_profile

cognitive_tester = CognitiveAnalyzer(attention_test, reaction_time_test,
perception_test, eeg_data)

input_data = cognitive_tester.collect_input_data()
normalized_data = cognitive_tester.normalize_data(input_data)
cognitive_profile = cognitive_tester.cognitive_profile(normalized_data)

```

Основні етапи алгоритму:

- збір вхідних даних: результати стандартизованих тестів, таких як тести уваги, швидкості реакції та сприйняття інформації, формують основу вхідних даних;
- тести уваги: оцінюється концентрація та увага пілота під час виконання завдань, спрямованих на тривалий збір уваги;
- тести швидкості реакції: визначається час реакції на різні подразники, які оцінюють загальний рівень реакційної швидкості пілота;
- тести сприйняття інформації: оцінка здатності пілота сприймати та обробляти вхідну інформацію, включаючи завдання на розпізнавання образів;
- нормалізація даних: отримані дані нормалізуються для стандартизації результатів та подальшого аналізу;
- формування когнітивного профілю: зібрані дані використовуються для створення когнітивного профілю пілота, який відображає рівень уваги, швидкість реакції та сприйняття інформації.

Використання наведеного алгоритму:

```

def perform_attention_test():
    attention_score = 42
    return attention_score

def perform_reaction_time_test():
    reaction_time_score = 56

```



```

    return reaction_time_score
def perform_perception_test():
    perception_score = 73
    return perception_score
eeg_data = get_eeg_data_from_sensor()
cognitive_analyzer = CognitiveAnalyzer(perform_attention_test,
perform_reaction_time_test, perform_perception_test, eeg_data)
input_data = cognitive_analyzer.collect_input_data()
normalized_data = cognitive_analyzer.normalize_data(input_data)
profile = cognitive_analyzer.cognitive_profile(normalized_data)
print("Cognitive Profile:")
print(profile)

```

Використані дані:

- рівень уваги: вимірює концентрацію та увагу пілота під час роботи;
- швидкість реакції: визначає час реакції на зовнішні подразники;
- сприйняття інформації: оцінка здатності пілота сприймати та обробляти інформацію.

Алгоритм враховує індивідуальні особливості психічної діяльності кожного пілота, щоб адаптувати аналіз до його унікальних потреб та можливостей. Інтеграція оцінки психічного навантаження робить його більш повним та реалістичним.

Алгоритм є ключовим компонентом передпольотного контролю, оскільки дозволяє ефективно оцінювати когнітивні параметри та рівень психічного навантаження пілота. Його унікальність полягає в здатності адаптуватися до індивідуальних особливостей, покращуючи об'єктивність та достовірність результатів аналізу.

Алгоритм синтезу фізичних та психологічних даних розроблено на високорівневому інтегрованому рівні, спрямованому на глибоке поєднання результатів фізичного та психологічного аналізу для отримання комплексного та вичерпного портрета стану пілота. Під час розробки алгоритму використовуються



багатовимірні моделі, які дозволяють враховувати взаємозв'язки та взаємовплив між фізичними та психологічними аспектами його функціонального стану.

Робота алгоритму починається з ініціалізації, а саме:

- створення екземпляра класу *PhysicalPsychologicalSynthesizer*;
- передача об'єктів фізичного та психологічного аналізаторів у конструктор:

Наступний приклад коду описує процес ініціалізації:

```
class PhysicalPsychologicalSynthesizer:  
    def init(self, physical_analyzer, psychological_analyzer):  
        self.physical_analyzer = physical_analyzer  
        self.psychological_analyzer = psychological_analyzer
```

Далі ми отримуємо дані викликом методів *analyze_physical_data* та *analyze_psychological_data* для отримання відповідних фізичних та психологічних даних. Наведемо приклад коду, що отримує зчитані дані:

```
def synthesize_data(self):  
    physical_data = self.physical_analyzer.analyze_physical_data()  
    psychological_data = self.psychological_analyzer.analyze_psychologi
```

Завершуємо алгоритм створенням синтезованих даних, об'єднуючи обрані параметри з фізичного та психологічного аналізів:

```
synthesized_data = {  
    "heart_rate": physical_data["heart_rate"],  
    "blood_pressure": physical_data["blood_pressure"],  
    "stress_level": psychological_data["stress_level"],  
    "adrenaline_level": psychological_data["adrenaline_level"],
```

Унікальність цього алгоритму полягає в його здатності адаптуватися до індивідуальних характеристик кожного пілота. Він не лише об'єднує дані, але і враховує їхню взаємодію та вплив на загальний стан пілота. Такий підхід допомагає уникнути ситуацій, де ізольований аналіз може призвести до неточних висновків та не враховувати всю комплексність фізичного та психологічного здоров'я пілота.

Алгоритм прийняття рішень на основі аналізу є критичним етапом у визначенні придатності пілота до виконання польотних завдань. Використовуючи



інтегрований аналіз фізичного, психологічного та когнітивного здоров'я, цей алгоритм формує об'єктивне рішення щодо продовження чи припинення польоту. Опис алгоритму прийняття рішень:

1) нормалізуємо та стандартизуємо дані, щоб усі дані, отримані з різних датчиків та тестів, проходять процес нормалізації для порівняння та стандартизації. В свою чергу, використовуючи комбінацію традиційної стандартизації та інтелектуального підходу до нормалізації, за рахунок застосування штучного інтелекту, наступним фрагментом коду враховуємо особливості конкретних даних:

```
def ai_blend_normalization(data):
    normalized_data = []
    for column in np.array(data).T:
        standardized_column = (column - np.mean(column)) / np.std(column)
        ai_adjusted_column = np.where(standardized_column < 0,
standardized_column + 1, standardized_column - 0.5)
        blended_column = 0.7 * standardized_column + 0.3 * ai_adjusted_column
        normalized_data.append(blended_column)
    return np.array(normalized_data).T
data = [
    (heart_rate, blood_pressure, body_temperature, oxygen_level, stress_level,
adrenaline_level, eeg, attention_level, reaction_speed, perception, vision_params,
light_reaction)]
normalized_data = ai_blend_normalization(data)
```

2) формування комплексний підхід, використовуючи багатовимірні моделі для об'єднання фізичних, психологічних та когнітивних аспектів здоров'я пілота;

3) враховуємо взаємозв'язки, для того щоб рішення алгоритму враховувало не лише окремі параметри, але і їхню взаємодію та вплив одних параметрів на інші.

Таким чином, алгоритм не просто враховує окремі показники, але і з'єднує їх взаємодію, надаючи повний зоровий на стан пілота. Це дозволяє точно та об'єктивно визначити його здатність до продовження або припинення польоту, забезпечуючи високий рівень безпеки та ефективності у повітряному просторі.



Зчитування фізичних показників є критично важливим елементом системи передпольотного контролю, спрямованої на забезпечення достовірних фізіологічних даних про пілота. Методи зчитування взаємодіють з фізичними сенсорами, включаючи сенсори серцевого ритму, термометри та тискоміри, забезпечуючи точне та надійне зчитування даних. Основні етапи включають в себе:

– взаємодія з фізичними сенсорами: активна взаємодія з різноманітними фізичними сенсорами через спеціальні інтерфейси та пристрої для оптимального отримання фізіологічних показників. Наступний код ініціалізує наведену взаємодію:

```
class PhysicalSensors:  
def __init__(self):  
self.heart_rate_sensor = HeartRateSensor()  
self.thermometer = Thermometer()  
self.barometer = Barometer()  
def interact_with_sensors(self):  
self.heart_rate_sensor.read_data()  
self.thermometer.read_data()  
self.barometer.read_data()  
def real_time_reading(self):  
while True:  
self.interact_with_sensors()  
time.sleep(1)  
def normalize_and_filter_data(self, raw_data):  
normalized_data = normalization_function(raw_data)  
filtered_data = filtering_function(normalized_data)  
return filtered_data  
def transfer_to_analysis_system(self, processed_data):  
analysis_system.receive_data(processed_data)
```

– зчитування в режимі реального часу: постійне та миттєве зчитування фізичних параметрів забезпечує актуальність даних, що є важливим для швидкого та точного реагування на зміни у стані пілота;



– нормалізація та фільтрація даних: отримані дані піддаються процесу нормалізації та фільтрації для видалення будь-яких артефактів чи шумів, забезпечуючи консистентність та точність інформації, і в кінці передаються до системи;

– передача у систему аналізу: чисті та нормалізовані дані передаються у систему аналізу стану пілота для подальшого використання.

Особливість цього алгоритму полягає в його здатності забезпечувати стабільну та високоточну передачу даних в режимі реального часу до системи аналізу, у якій всі дані перед початком аналізу спочатку будуть підлягати процесам нормалізації та фільтрації, враховуючи різноманіття фізичних сенсорів на тілі пілота та забезпечуючи ефективність передпольотного контролю.

3.6. Висновки до розділу

У розділі 3 було розглянуто проектування програмного засобу передпольотного контролю стану пілота. Цей розділ надає загальне уявлення про ключові аспекти проектування та визначає важливі параметри системи.

Були визначені основні цілі та завдання проектування системи передпольотного контролю стану пілота. Головною метою є створення високоефективного та надійного інструменту для медичного обстеження пілотів перед польотом. Основні завдання включають розробку інтерфейсу користувача, автоматизовану транскрипцію, аналіз медичних даних, забезпечення безпеки даних та інтеграцію з існуючими сервісами.

У розділі 3.2 була описана архітектурна концепція системи передпольотного контролю. Вибір мікросервісної архітектури дозволяє розробляти окремі компоненти системи, що спрощує розширення та обслуговування. Використання хмарних сервісів *Amazon Web Services (AWS)* дозволяє легко інтегрувати систему з надійними та масштабованими рішеннями для збереження, аналізу даних та забезпечення безпеки.



Система не лише дозволяє отримати висновки щодо фізичного стану, але й визначає ризики та надає конкретні рекомендації для їх врегулювання. Подальший розділ функціональних вимог чітко визначає процеси аналізу та робить акцент на безпеці та конфіденційності оброблюваної медичної інформації. Застосування інтеграції з *Amazon Web Services* додає масштабованість та доступність до системи, забезпечуючи надійність та ефективність в її функціонуванні.

У підрозділі 3.4 були детально визначені функціональні вимоги до програмного засобу, охоплюючи операції та можливості для користувачів. Сценарії використання були описані для кожної функції, і їх відповідність основним цілям системи була обґрунтована.

У підрозділі 3.5 розроблено та описано унікальні алгоритми аналізу показників пілота, що мають забезпечити високоякісний аналіз зчитаних показників, а саме: алгоритм аналізу серцевого ритму та фізичної активності; частотний аналіз серцевого ритму, що використовує використовуючи методи частотного аналізу; часовий аналіз, котрий виконує аналіз фізичної активності; алгоритм для аналізу когнітивних параметрів, розробка якого включає комплексну оцінку рівня уваги, швидкості реакції та сприйняття інформації; алгоритм синтезу психофізіологічних даних пілота, що спрямований на поєднання результатів фізичного та психологічного аналізу для отримання комплексного та фінального рішення про стану пілота. Насамперед, алгоритм прийняття рішень на основі аналізу є найважливішим етапом у визначенні придатності пілота до польотів. Додатково, застосовано розроблений алгоритм зчитування фізичних показників, що забезпечує зчитування та надсилання до системи фізіологічних даних пілота.



РОЗДІЛ 4

РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАСОБУ

4.1. Загальний опис програмного засобу

При розгляді проекту та його ключових файлів необхідно зрозуміти організацію ресурсів та їх призначення. Від цього залежить ефективність розробки та подальшого розгортання програмного засобу передпольотного контролю стану пілота. Наш наступний крок – детальний розгляд файлової структури та функціональності кожного компонента проекту.

Файли програмного проекту організовані у наступні каталоги:

Bin – в цьому каталозі містяться виконавчі файли та скрипти, які можуть пов'язані з автоматизацією певних процесів у проекті.

Lambda – цей каталог містить функції *AWS Lambda* або інші ресурси, які пов'язані з обробкою функцій в хмарних сервісах *AWS*.

Lib – цей каталог містить бібліотеки та модулі, які використовуються в проекті.

Public – у цьому каталозі знаходяться ресурси та файли, доступні для загального використання, наприклад, статичні файли, які обслуговуються веб-сервером.

Src – цей каталог містить вихідний код програмного засобу. Він включає файли, які відповідають за логіку додатку, користувацький інтерфейс, інші складові програми.

Test – в цьому каталозі розміщуються тести та скрипти для тестування програмного засобу.

Деякі важливі файли, що входять до складу програмного проекту:

.eslintrc.js – цей файл встановлює правила лінтингу (перевірки на дотримання кодового стилю) для проекту. Він допомагає забезпечити, що код відповідає вимогам до якості.



.prettierrc – цей файл містить конфігурацію для інструменту *Prettier*, який допомагає формувати код так, щоб він був більш читабельним і структурованим.

cdk.json – цей файл містить конфігурацію *AWS Cloud Development Kit (CDK)*, яка використовується для створення і розгортання інфраструктури.

package.json – цей файл містить інформацію про залежності проекту та скрипти для його збірки та запуску.

tsconfig.json – цей файл встановлює налаштування для *TypeScript*, якщо проект використовує цю мову програмування.

yarn.lock – файл *yarn.lock* містить інформацію про версії залежностей, встановлених за допомогою менеджера пакетів *Yarn*.

Компонент *AnalysisPane* служить для аналізу результатів зчитаних даних, і він містить ряд підкомпонентів та функціонал для взаємодії з результатами аналізу. Компонент *AnalysisPane* забезпечує структурування та аналіз результатів показників, включаючи категоризацію результатів, додавання нових елементів та редагування вже існуючих. Він допомагає організувати та працювати з великою кількістю результатів у зручний спосіб.

Опис компонента *AnalysisPane* та його складових:

- це кореневий компонент, що відповідає за аналіз зчитаних даних;
- він містить список категорій, які служать як розділи для результатів;
- викликається для кожної категорії для аналізу кожного показника.

Опис функції *ResultTable* та її складових:

- відповідає за відображення зчитаних результатів для певної категорії;
- включає список результатів, що відносяться до цієї категорії;
- надає можливість додавання нових елементів до цієї категорії;
- має можливість видалення результатів та редагування обраних концепцій.

Опис функції *ResultRow*:

- відповідає за відображення окремого рядка результату або концепції в межах категорії;
- надає можливість вибирати концепцію зі списку, видаляти результат та змінювати атрибути.



Компонент *TranscriptLine* використовується для відображення кожного рядка зчитаних даних.

Він приймає рядок *chunk*, результати *Comprehend*, список включених категорій *enabledCategories*, інформацію про можливість редагування *enableEditing*, а також функції для початку аналізу та зчитування даних. Функція *combineSegments* призначена для зведення результатів *Comprehend* до нормалізації неперекриваючихся діапазонів із списком відповідних результатів. Використовується стилізація за допомогою класів *classMap* та компонентів *Chakra-UI*, таких як *EditableClass*, *EditableInputClass*, *EditablePreviewClass*, *BoxClass*.

Компонент *App* є тією частиною додатка, що відповідає за визначення, чи користувач авторизований чи ні.

Використовується глобальний стан за допомогою *React*-хука *useState* для збереження інформації про те, чи користувач авторизований. Використовує *React*-хук *useEffect* для встановлення слухача, який викликає функцію *onLoad* при завантаженні компонента.

Метод *onLoad* перевіряє, чи існує поточна сесія користувача за допомогою *Auth.currentSession()*. Якщо користувач має сесію, то встановлюється значення *isAuthenticated* в *true*, що вказує на те, що користувач автентифікований. Якщо не існує поточного користувача або відбулася якась помилка, то виводиться повідомлення про помилку.

Після завершення перевірки автентифікації, встановлюється значення *isAuthenticating* в *false*, що дозволяє відображати основний контент додатка.

Компонент обгорнутий умовною логікою, яка виводить основний контент (елементи *AppContext.Provider* і *Routes*) лише тоді, коли *isAuthenticating* встановлено в *false*.

Для оцінки стану пілота використовуємо різноманітні вхідні дані. Важливо враховувати як фізичні, так і психологічні параметри. Розглянемо список вхідних фізичних параметрів для аналізу стану пілота:

- серцевий ритм;
- кров'яний тиск;



- температура тіла;
- рівень кисню в крові.

Також наведемо список психологічних параметрів, що будуть оцінюватись у аналізі:

- рівень стресу (аналіз біомаркерів стресу, таких як кортизол);
- рівень адреналіну (кількості адреналіну в крові);
- електроенцефалографія (електричної активності мозку).

Наступні когнітивні параметри враховуватимуться до оцінки передпольотного стану під час аналізу:

- рівень уваги (вимірювання концентрації);
- швидкість реакції (час реакції на зовнішні подразники);
- сприйняття інформації (здатність пілота сприймати та обробляти інформацію).

Алгоритм аналізу в свою чергу включатиме також наведені нижче візуальні параметри:

- параметри зору (оцінкаостроти зору, кольоророзрізнення);
- реакція на світлові подразники (аналіз реакції на зміни світла).

Отримання фізичних та психологічних даних пілота здійснюється за допомогою різних датчиків та пристроїв. Давайте розглянемо можливий сценарій отримання даних з деяких типових сенсорів. Багато біометричних пристроїв, таких як пульсометри чи датчики кров'яного тиску, можуть підтримувати стандартні протоколи бездротового з'єднання, такі як *Bluetooth* або *ANT+*. Використаємо бібліотеку *bluepy* для роботи з *Bluetooth*.

Наступний код ініціалізує використання відповідних бібліотек *Python* для отримання даних через протоколи:

```
def connect_to_sensor(device_address):  
    device = bluepy.btle.Peripheral(device_address)  
    return device  
  
def read_sensor_data(device):  
    # Зчитайте дані з датчика
```




```
data = device.readCharacteristic(handle)  
print(f"Received data: {data}")  
sensor_address = '00:11:22:33:44:55'  
sensor_device = connect_to_sensor(sensor_address)  
read_sensor_data(sensor_device)
```

Транслявання даних може відбуватися різними способами, в залежності від обраної технології та пристрою. Однак, основний принцип полягає в отриманні даних з датчика та подальшому передаванні їх до програми, яка обробляє та аналізує ці дані.

Наступний фрагмент коду показує логіку транслявання даних:

```
def start_server():  
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)  
server_socket.bind(('localhost', 12345))  
server_socket.listen()  
client_socket, client_address = server_socket.accept()  
print(f"Connected to {client_address}")  
while True:  
data = client_socket.recv(1024)  
if not data:  
break  
print(f"Received data: {data.decode('utf-8')}")  
client_socket.close()  
start_server()
```

Наведемо приклад коду, що виконує реалізацію на рівні сокетів і описує сервер-клієнтський підхід з використанням серверної частини, що приймає дані від датчиків і надсилає їх в програму, яка виконує аналіз:

```
def send_data_to_server(data):  
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)  
client_socket.connect(('localhost', 12345))  
client_socket.sendall(data.encode('utf-8'))
```



```
client_socket.close()
# Зразок використання
sensor_data = "Heart Rate: 75 bpm"
send_data_to_server(sensor_data)
```

Для вимірювання серцевого ритму використовуємо датчик пульсометра або екг-монітор. Пристрій може бути підключений до пілота за допомогою бездротового з'єднання, такого як *Bluetooth* або *ANT+*, або ж за допомогою фіксованих датчиків, які вбудовані в пілотський екіпаж. Даний фрагмент коду ініціює отримання даних серцевого ритму:

```
def get_heart_rate():
    nearby_devices = bluetooth.discover_devices(duration=8, lookup_name=True)
    if nearby_devices:
        device_address, device_name, _ = nearby_devices[0]
        print(f"Connecting to {device_name} ({device_address})...")
        sock = bluetooth.BluetoothSocket(bluetooth.RFCOMM)
        sock.connect((device_address, 1))
        while True:
            data = sock.recv(1024)
            heart_rate = int.from_bytes(data, byteorder='big')
            print(f"Heart Rate: (heart_rate) bpm")
            time.sleep(1)
        sock.close()
    else:
        print("No nearby Heart Rate Monitor found.")
    get_heart_rate()
```

Для вимірювання кров'яного тиску використовуємо датчик тонометра. Пристрій також може бути підключений за допомогою бездротового з'єднання або ж використовувати фіксовані датчики.



Для вимірювання температури тіла використовуємо температурний датчик, який може бути вбудований у пілотський костюм чи підключений окремо. Додатково наведемо лістинг коду, що отримує дані температури тіла та кров'яного тиску

```
get_blood_pressure():  
# Simulation of received data  
    systolic_pressure = random.randint(90, 120)  
    diastolic_pressure = random.randint(60, 80)  
    print(f"Blood Pressure: {systolic_pressure}/{diastolic_pressure} m  
get_body_temperature():  
# Simulation of temperature  
body_temperature = random.uniform(36.5, 37.5)  
print(f"Body Temperature: {body_temperature} °C")  
# Start a function to get simulated data  
while True:  
    get_blood_pressure()  
    get_body_temperature()  
    time.sleep(i)
```

Алгоритм перевірки фізичного стану пілота є комплексною системою, розробленою для об'єктивної оцінки його здоров'я перед виконанням повітряних місій. В основу алгоритму вбудовані ряд індивідуалізованих фізичних аналітичних алгоритмів, які оцінюють ключові фізіологічні параметри.

Один із ключових компонентів – аналіз серцевого ритму. Алгоритм взаємодіє з датчиком серцевого ритму, отримуючи дані, які потім аналізуються на предмет відхилень від норми. Унікальність полягає в тому, що враховується індивідуальний серцевий ритм пілота, а не обмежуючись загальними стандартами.

Другий аспект – аналіз тиску крові, що включає в себе оцінку систолічного та діастолічного тиску. Алгоритм визначає, чи значення перевищують норму, і при необхідності генерує попередження про високий тиск.



Третій компонент – аналіз температури тіла. Алгоритм взаємодіє з датчиком термометру, аналізуючи значення. Підвищення температури в порівнянні з рекомендованими величинами спричиняє генерацію попередження.

Четвертий аспект – аналіз рівня кисню в крові. Датчик рівня кисню надає дані, які алгоритм аналізує, визначаючи, чи не виникає низький рівень кисню.

Завершальна фаза алгоритму – інтеграція індивідуальних фізичних аналітичних алгоритмів. Алгоритм розглядає взаємозв'язки між фізіологічними показниками, формуючи комплексну картину фізичного стану пілота.

Унікальність цього алгоритму виявляється у врахуванні індивідуальних характеристик кожного пілота, що робить оцінку більш точною та об'єктивною. Отримані результати можуть використовуватися для прийняття рішень щодо придатності пілота для виконання польотів.

Алгоритм аналізу психологічних параметрів пілота спрямований на визначення його психологічного стану через оцінку рівнів стресу, адреналіну та застосування електроенцефалографії.

Алгоритм аналізу психологічних параметрів є критично важливою складовою системи передпольотного контролю, спрямованою на об'єктивне оцінювання здатності пілота виконувати повітряні місії. Його функціональність включає аналіз різноманітних показників, таких як рівень стресу, концентрація адреналіну та дані електроенцефалографії. Важливою рисою є взаємодія цього алгоритму з іншими частинами системи передпольотного контролю, що дозволяє створювати комплексні оцінки стану пілота.

Однією з ключових особливостей цього алгоритму є врахування індивідуальних особливостей психологічного стану пілота та його адаптація до стресових ситуацій. Наприклад, алгоритм враховує високотехнологічний аналіз даних ЕЕГ, спрямований на визначення рівня активності альфа-хвиль у мозку пілота. Це дозволяє виявляти можливі відхилення в психічному стані, такі як когнітивна втома, що може виникнути під час тривалого польоту.

Такий інтегрований підхід забезпечує фізіологічний аналіз та підвищує об'єктивність визначення придатності пілота для польоту. Узгодженість з іншими



алгоритмами, які аналізують фізичний та когнітивний стан, створює повноцінний комплекс, спроможний надавати високоточні рекомендації забезпечення безпеки та ефективності місії.

Вказаний фрагмент коду виконує загальний алгоритм аналізу даних та прийняття рішення:

```
class ComprehensiveHealthAnalyzer:  
    def analyze_health(self):  
        physical_analyzer = PhysicalHealthAnalyzer(**self.physical_data)  
        psychological_analyzer = PsychologicalStateAnalyzer(**self.psychological_data)  
        psychological_result = psychological_analyzer.analyze_psychological_state()  
        cognitive_analyzer = CognitiveAnalyzer(**self.cognitive_data)  
        cognitive_result = cognitive_analyzer.analyze_cognitive_parameters()  
        decision = {"flight_approval": True, "reasons": []}  
        if physical_result["status"] == "Abnormal":  
            decision["flight_approval"] = False  
            decision["reasons"].append("Abnormal physical health.")  
            if psychological_result["stress_level"] == "High" or  
psychological_result["adrenaline_level"] == "High":  
                decision["flight_approval"] = False  
                decision["reasons"].append("High psychological stress or adrenaline  
levels.")  
            if cognitive_result["attention_level"] == "Low" or  
cognitive_result["reaction_time"] > 1.0:  
                decision["flight_approval"] = False  
                decision["reasons"].append("Low cognitive performance.")  
        return decision
```

Цей комплексний алгоритм передпольотного контролю стану пілота спрямований на оцінку його фізичного, психологічного та когнітивного стану для забезпечення безпеки та ефективності польоту.



Перший етап включає аналіз фізичного стану, який оцінює параметри, такі як серцевий ритм, артеріальний тиск та температура тіла. Якщо отримані дані вказують на аномалії, наприклад, високий тиск чи низьку температуру тіла, алгоритм приймає рішення про непридатність пілота до польоту з обґрунтуванням.

Другий етап включає аналіз психологічного стану, який оцінює рівень стресу, кількість адреналіну та результати електроенцефалографії. Якщо виявлені високий рівень стресу чи адреналіну, або аномалії в ЕЕГ, це також може призвести до відхилення від придатності пілота до польоту.

Останній етап включає аналіз когнітивних параметрів, таких як рівень уваги та швидкість реакції. Якщо виявлені низький рівень уваги чи повільна реакція, алгоритм може призначити пілота непридатним для польоту.

Загальний аналіз об'єднує результати трьох етапів та приймає об'єктивне рішення щодо придатності пілота. Унікальність алгоритму полягає в його здатності інтегрувати різноманітні дані та враховувати комплексні взаємозв'язки між різними аспектами фізичного, психологічного та когнітивного здоров'я пілота, що забезпечує об'єктивність та безпеку при прийнятті рішень щодо його готовності до польоту.

4.2. Взаємодія з сервісами *Amazon*

Взаємодія з сервісами *Amazon* – це важлива частина багатьох додатків, особливо тих, які розгортаються на *Amazon Web Services (AWS)* або використовують *AWS*-послуги. Для взаємодії з різними сервісами *Amazon* будемо використовувати бібліотеку *aws-amplify*, яка надає спрощений спосіб роботи з *AWS*-сервісами [22]. Ось кілька кроків, які допоможуть почати взаємодіяти з *Amazon*-сервісами:

- встановлення *aws-amplify*: спочатку потрібно встановити *aws-amplify* за допомогою *npm* або *yarn*;
- налаштування *AWS*: перед використанням *aws-amplify* потрібно налаштувати доступ до сервісів *AWS*. Це включає в себе створення файлу конфігурації *AWS* та надання необхідних дозволів для додатка;



– імпорт та конфігурація *Amplify*: у додатку потрібно імпортувати *aws-amplify* і налаштувати його з використанням конфігурації *AWS*;

– взаємодія з сервісами *AWS*: тепер можемо взаємодіяти з *AWS*-сервісами за допомогою *aws-amplify*. Наприклад, використовуємо *Auth* для керування аутентифікацією користувачів, *API* для взаємодії з *AWS API Gateway* або іншими службами *API*.

Даний код ілюструє роботу програми з сервісом *Auth* для аутентифікації користувачів:

```
import { Auth } from 'aws-amplify';
async function signUp(username, password) {
  try {
    const { user } = await Auth.signUp({
      username,
      password,
    });
    console.log('Registered user:', user);
  } catch (error) {
    console.error('Error registering user:', error);
  }
}
```

Наступний скрипт забезпечує вхід до програми за ім'ям користувача (*username*) та паролем (*password*):

```
async function signIn(username, password) {
  try {
    const user = await Auth.signIn(username, password);
    console.log('Authenticated user:', user);
  } catch (error) {
    console.error('Error signing in:', error);
  }
}
```



Додатковий скрипт перевіряє вихід користувача з програми:

```
async function signOut() {  
  try {  
    await Auth.signOut();  
    console.log('User signed out');  
  } catch (error) {  
    console.error('Error signing out:', error);  
  }  
}
```

Взаємодія з *Amazon DynamoDB* – це ключовий аспект розробки багатьох сучасних додатків, особливо веб-сервісів та мобільних додатків [23]. *DynamoDB* є флексібельною та повністю керованою службою бази даних, розробленою *Amazon Web Services (AWS)*, і вона надає багато переваг для зберігання та отримання даних про пілотів.

База даних в *Amazon DynamoDB* для системи передпольотного контролю стану пілота містить в собі 9 таблиць, що дозволяють зберігати та організувати дані для ефективного використання системи передпольотного контролю (рис. 4.5).



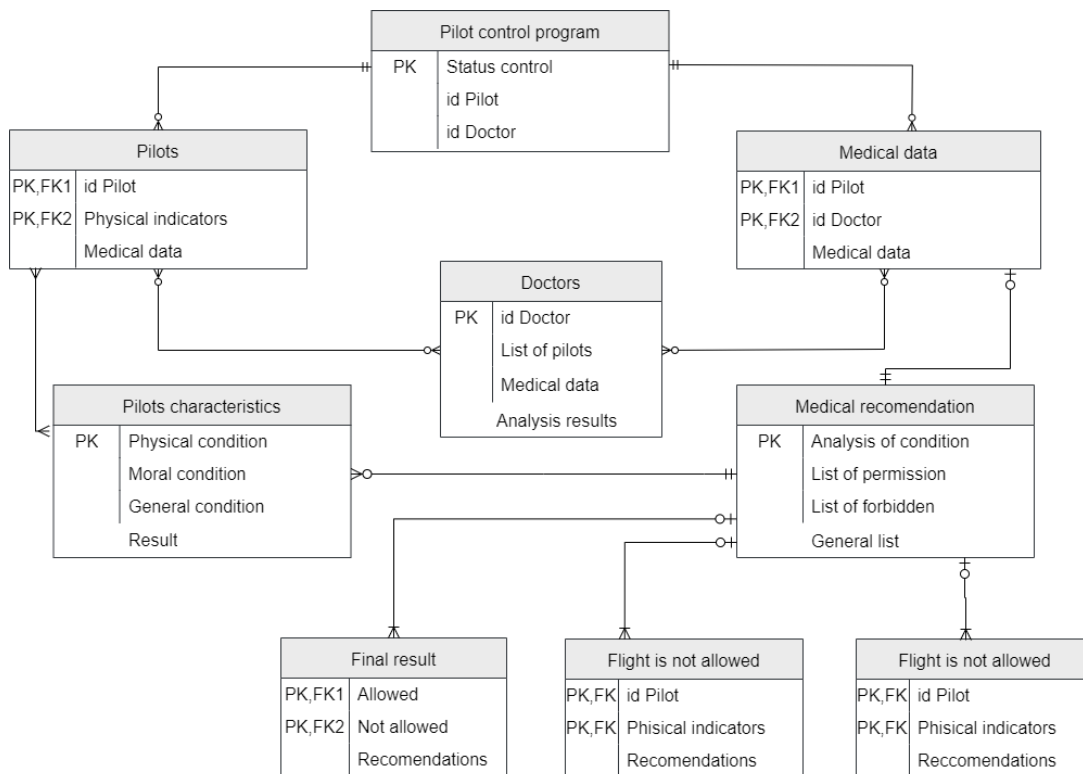


Рис. 4.5. Модель бази даних програмного засобу у *DynamoDB*

Таким чином, наведена база даних може відігравати ключову роль у підтримці та оптимізації функціонування програмного засобу передпольотного контролю стану пілота, оскільки 9 створених таблиць забезпечують зберігання даних про пілотів, лікарів, зчитаних даних, проаналізованих даних та прийняття рішення про допущення чи недопущення до польотів, рекомендацій алгоритмів аналізу та наявних лікарів.

Забезпечення належного рівня захисту даних у програмі передпольотного контролю є критично важливим, враховуючи індивідуальні дані кожного пілота та вміст великого обсягу інформації. Так як, програма передпольотного контролю стану пілота використовує сервіси *AWS* застосуємо *AWS Key Management Service (KMS)* для керування ключами шифрування програми передпольотного контролю. Застосування шифрування у програмі передпольотного контролю визначається необхідністю захистити дані пілота та іншу важливу інформацію від несанкціонованого доступу під час обробки та обміну. Це зроблено з метою забезпечення безпеки польотів, захисту особистих даних та виконання вимог стандартів безпеки в авіаційній галузі.



Симетричне шифрування є швидшим та більш ефективним для обробки великої кількості даних. У програмі передпольотного контролю, де може виникати потреба в обробці оперативної реакції на дані в режимі практичного використання, це є важливим фактором.

Створюємо ключ шифрування ключ симетричного шифрування та отримуємо ідентифікатор створеного ключа.

Надалі забезпечимо шифрування та дешифрування даних з використанням створеного ключа з *AWS KMS*. Наступний код забезпечує шифрування інформації про наявних лікарів:

```
dynamodb = boto3.resource('dynamodb')
table = dynamodb.Table('Doctors')
response = table.query(
    KeyConditionExpression=Key('Occupation').eq('Doctor')
    plaintext_data = b'Confidential data'
    context = {'Purpose': 'DataEncryption'}
    response = kms.encrypt(KeyId=key_id, Plaintext=plaintext_data,
    EncryptionContext=context)
    ciphertext_blob = response['CipherBlob']
    decrypted_data = kms.decrypt(CiphertextBlob=ciphertext_blob,
    EncryptionContext=context)
```

Регулярно оновлення симетричних ключів реалізовано автоматично за допомогою *AWS KMS*, що автоматично обертає ключі на підставі використання на основі розкладу.

Інтегруємо наведені операції шифрування та дешифрування в програму передпольотного контролю, де дані передаються між різними компонентами:

```
def encrypt_data(data):
    response = kms.encrypt(KeyId=key_id, Plaintext=data,
    EncryptionContext=context)
    return response['CipherBlob']
def decrypt_data(ciphert_blob):
```



```
return kms.decrypt(CipherBlob=ciphertext_blob,  
EncryptionContext=context)['Plaintext']
```

Керування доступом до ключа є критично важливим елементом безпеки. Опишемо деталі щодо керування доступом до ключа за допомогою *AWS Identity and Access Management (IAM)* для симетричного ключа шифрування в програмі передпольотного контролю стану пілота.

Створюємо *IAM* роль, яка буде використовуватися для доступу до ключа шифрування:

```
role_name = 'EncryptionRole'  
assume_role_policy_document = {  
    'Version': '01',  
    'Statement': [  
        {'Effect': 'Allow',  
         'Principal': {'Service': 'lambda.amazonaws.com'},  
         'Action': 'sts:AssumeRole' } ]}  
iam.create_role(  
    RoleName=role_name,  
    AssumeRolePolicyDocument=json.dumps(assume_role_policy_document)
```

Періодично створюємо резервні копії ключів програми, таким чином забезпечуючи можливість їх відновлення у випадку втрати або пошкодження.

Поліпшення безпеки шифрування включає в себе використання сучасних криптографічних алгоритмів, застосування коректних режимів шифрування та дотримання високоякісних практик. Нижче приведений код шифрування з використанням бібліотеки *cryptography*:

```
def encrypt_data(data, key):  
    iv = os.urandom(16)  
    cipher = Cipher(algorithms.AES(key), modes.CFB(iv),  
backend=default_backend())  
    encryptor = cipher.encryptor()  
    ciphertext = encryptor.update(data) + encryptor.finalize()
```



```

    encrypted_data = iv + ciphertext
    return encrypted_data
def decrypt_data(encrypted_data, key):
    iv = encrypted_data[:16]
    cipher = Cipher(algorithms.AES(key), modes.CFB(iv),
backend=default_backend())
    decryptor = cipher.decryptor()
    decrypted_data = decryptor.update(encrypted_data[16:]) +
decryptor.finalize()
    return decrypted_data

```

Цей код використовує алгоритм шифрування *AES* в режимі *CFB* (*Cipher Feedback*), який вважається безпечним. Також важливо враховувати, що ключ має бути довгим та випадковим. Застосовані також інші додаткові заходи, такі як випадковий ініціалізаційний вектор (*IV*), що покращує безпеку програмного засобу при шифруванні медичних даних. Згенерували випадковий ініціалізаційний вектор (*IV*), який використовується для покращення та підтримки безпеки шифрування. Ініціалізуємо алгоритм шифрування *AES* в режимі *CFB* з ключем та *IV*. Створюємо об'єкт шифрування та шифруємо дані, додаючи до них *IV*.

В результаті такого алгоритму шифрування ми вказуємо усі дані, які потрібно зашифрувати та генеруємо випадковий 256-бітовий ключ для *AES-256*. Цей код використовує сучасні методи шифрування та дотримується високоточних практик для забезпечення безпеки даних у програмі передпольотного контролю.

Наведений код встановлює зв'язок між політикою доступу до створеної ролі, оскільки політика повинна містити права доступу до ключа шифрування (*AWS Key Management Service - KMS*):

```

policy =
{
    'Version': '01',
    'Statement':
iam.put_role_policy(

```



```
RoleName=role_name,  
PolicyName='EncryptionAccessPolicy',  
PolicyDocument=json.dumps(policy))
```

У даному випадку *arn:aws:kms:region:account-id:key/key-id* – це реальний *ARN* (*Amazon Resource Name*) ключа *KMS*.

Присвоюємо створену роль користувачеві, який буде використовувати цей ключ для шифрування та дешифрування даних:

```
lambda_client.update_function_configuration  
(  
    FunctionName='Admin',  
    Role='arn:aws:iam::account-id:role/YourEncryptionRole'  
)
```

Такий процес шифрування дозволяє забезпечити доступ до симетричного ключа шифрування лише конкретним користувачам системи, які мають необхідні дозволи управління. Керування доступом таким чином допомагає уникнути неправомірного використання ключа і підвищує загальний рівень безпеки програми передпольотного контролю.

4.3. Інтерфейс користувача

У цьому розділі ми розглянемо ключові елементи та функціональність інтерфейсу системи контролю фізичного стану пілота, зокрема спрямовані на аналіз фізичних показників та забезпечення ефективної взаємодії з системою.

Головна сторінка інтерфейсу є стартовою точкою для пілотів. Тут вони можуть виконати запис фізичних показників, завантажити попередньо записані дані або переглядати прикладові фізичні показники. Інтерфейс містить інтегрований датчик для запису фізичних параметрів.

Після завершення запису показників, система встановлює *WebSocket*-з'єднання. Це забезпечує отримання реального аналізу та рекомендацій.

Перший крок для використання додатку – вибір можливості входу до облікового запису лікаря чи пілота. Введення відбувається через обрання ролі та



введення облікових даних. На рис. 4.6 подано вигляд вхідного вікна з інтерфейсом українською мовою.

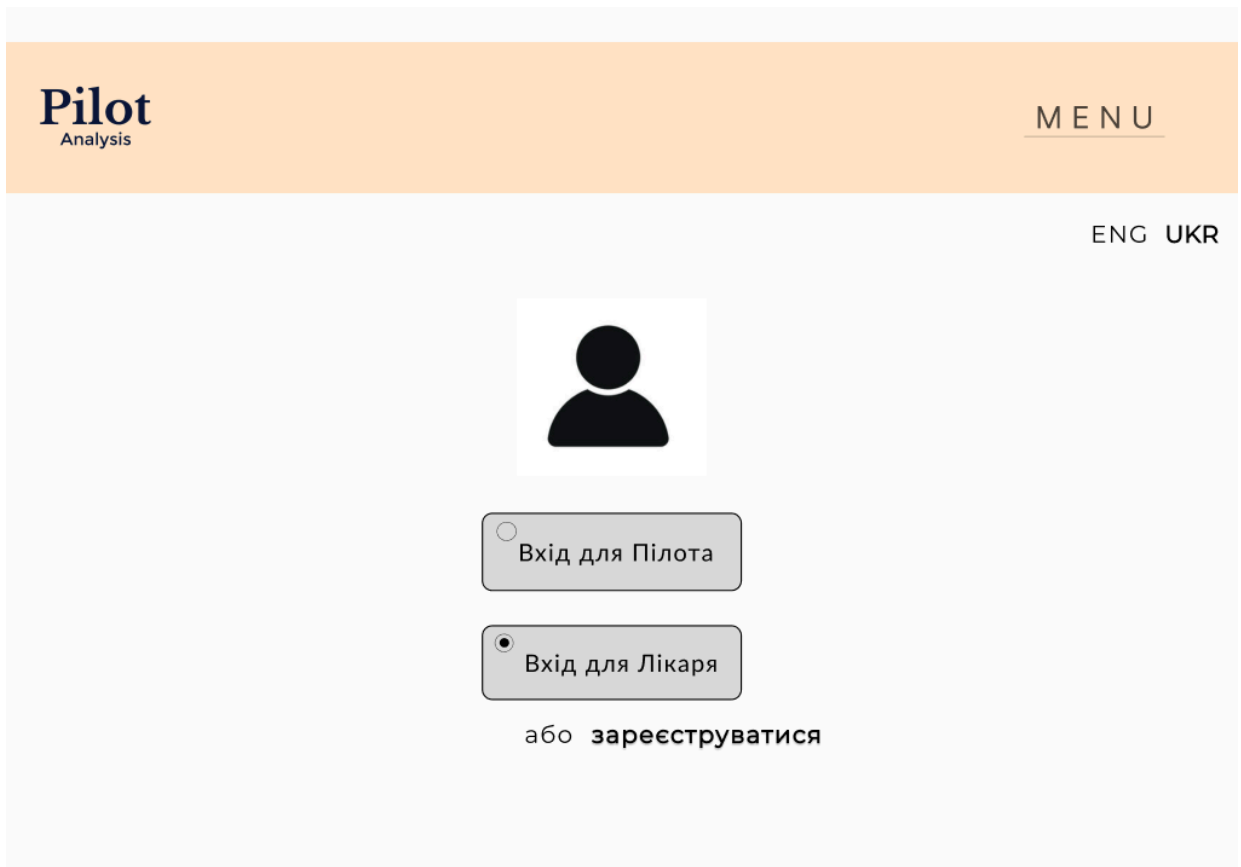


Рис. 4.6. Вхід у додаток (українська версія)

Даний дизайн інтерфейсу забезпечує зручність та ефективність взаємодії пілотів та лікарів із системою контролю фізичного стану. Для розширення кола користувачів розроблено також англійську версію інтерфейсу (рис. 4.7).



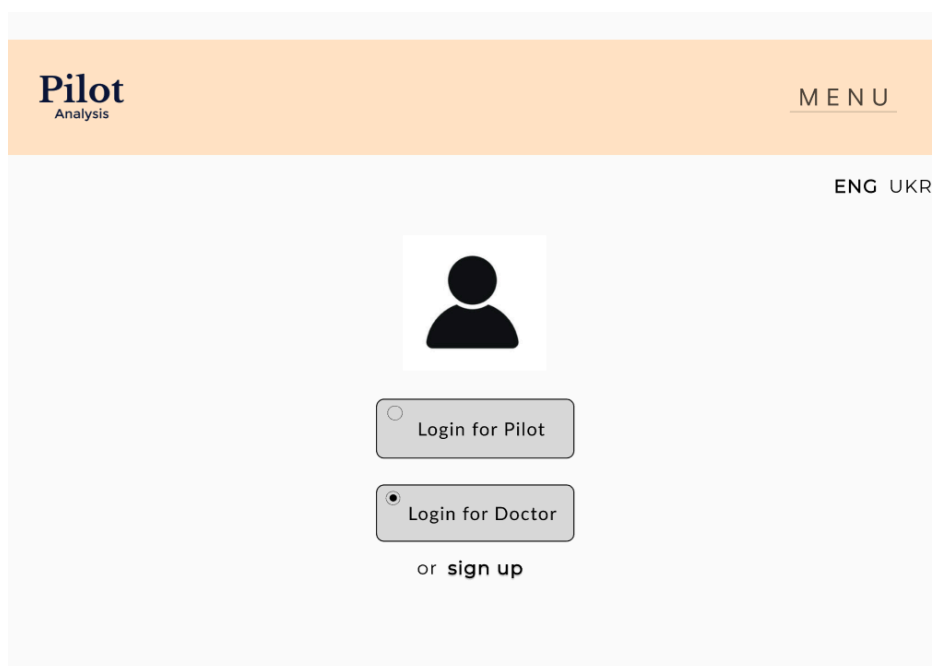


Рис. 4.7. Вхід у додаток (англійська версія)

Якщо обрано роль лікаря, доступний інтерфейс для перегляду та аналізу медичних даних пілотів. Лікар має доступ до інструментів та бібліотек для проведення детального аналізу стану здоров'я пілотів та виявлення ризикових чинників.

Якщо обрано роль пілота, користувачу після входу стає доступний інтерфейс для завантаження медичних даних та перегляду стану. Інформація про медичний стан виводиться у вікно, а також надаються рекомендації та вимоги щодо готовності до польотів.

Запис, трансляція та аналіз фізичних показників стають прозорими та доступними завдяки інтеграції різноманітних функціональних можливостей. За допомогою кнопки *Start Tracking* можемо почати зчитування всіх даних з підключених датчиків (рис. 4.8). Під час оцінки фізичного стану пілота наша система використовує розроблені алгоритми, щоб детально проаналізувати фізичні показники, такі як серцебиття, температура тіла і інші. Розроблені алгоритми порівнюють ці дані з індивідуальними нормами, встановленими для кожного пілота. На основі результатів аналізу робиться висновок щодо того, чи може пілот виконувати льотну роботу.



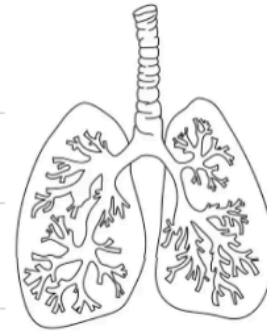
Pilot ID

Heart Rate: 78 beats per minute
 Blood Pressure: 120/80 mmHg
 Body Temperature: 36.8 degrees Celsius
 Blood Oxygen Level: 98%

Stress Level: 25 units
 Adrenaline Level: 0.7 micrograms per deciliter (µg/dL)
 EEG: 30 microvolts (alpha wave amplitude)

Attention Level: 75%
 Reaction Speed: 250 milliseconds
 Cognitive Processing: 85% accuracy

Vision Parameters: 20/20 vision
 Color differentiation: 95%
 Reaction to Light: 150 milliseconds (response time)



Start Tracking

Start Analysis

Рис. 4.8. Опис медичного стану на основі зчитаних даних

У процесі обробки фізичних даних розроблені алгоритми взаємодіють з медичними показниками пілота, щоб оцінити відхилення від стандартних значень. Це включає визначення того, чи відповідає серцебиття конкретного пілота встановленим медичним стандартам безпеки польотів. Такий підхід дозволяє забезпечити об'єктивну оцінку фізичного стану пілота та визначити його придатність для виконання польотів.

На основі результатів аналізу система приймає рішення щодо можливості польоту. Якщо фізичний стан пілота відповідає встановленим нормам, то йому дозволяться польоти.

У протилежному випадку система рекомендує відсторонити пілота від виконання польотних завдань до тих пір, поки його фізичний стан не відновиться до безпечного рівня. Крім цього, базуючись на генерованому аналізі, система демонструє персональні рекомендації (рис. 4.9).



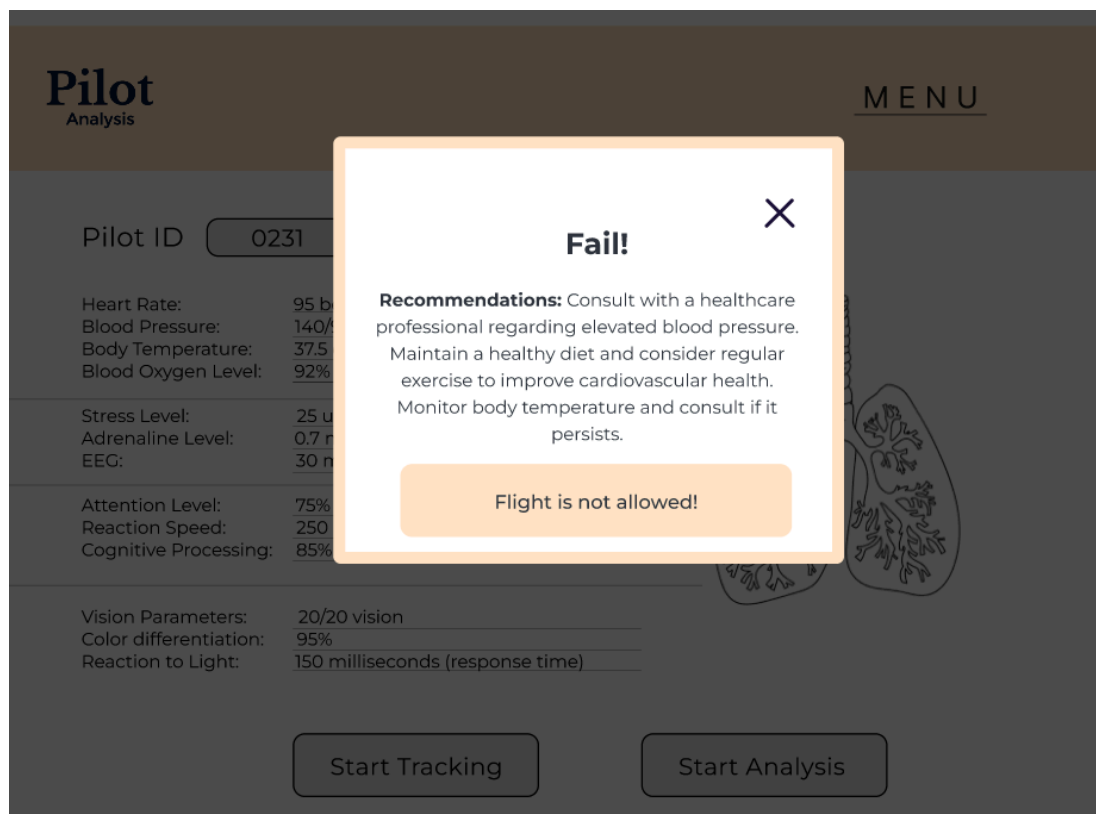


Рис. 4.9. Виліт заборонено, фізичні показники менше за рекомендовані

Система передпольотного контролю надає можливість пілотам та лікарям детально переглядати історію здоров'я пілота. Пілоти мають доступ до результатів усіх попередніх медичних обстежень, включаючи фізичні показники, аналізи та відмітки щодо їхнього стану здоров'я (рис. 4.10).

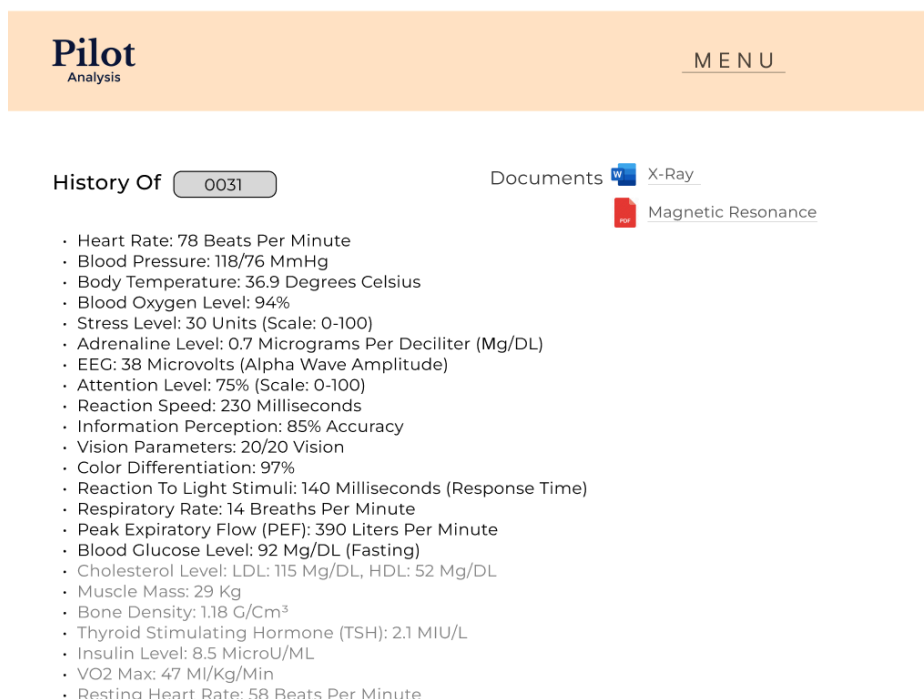


Рис. 4.10. Історія користувача

У випадку, якщо усі фізичні показники пілота знаходяться в межах норми та не виявлено жодних аномалій чи відхилень, система передпольотного контролю надає рішення про дозвіл на виконання польоту (рис. 4.11).

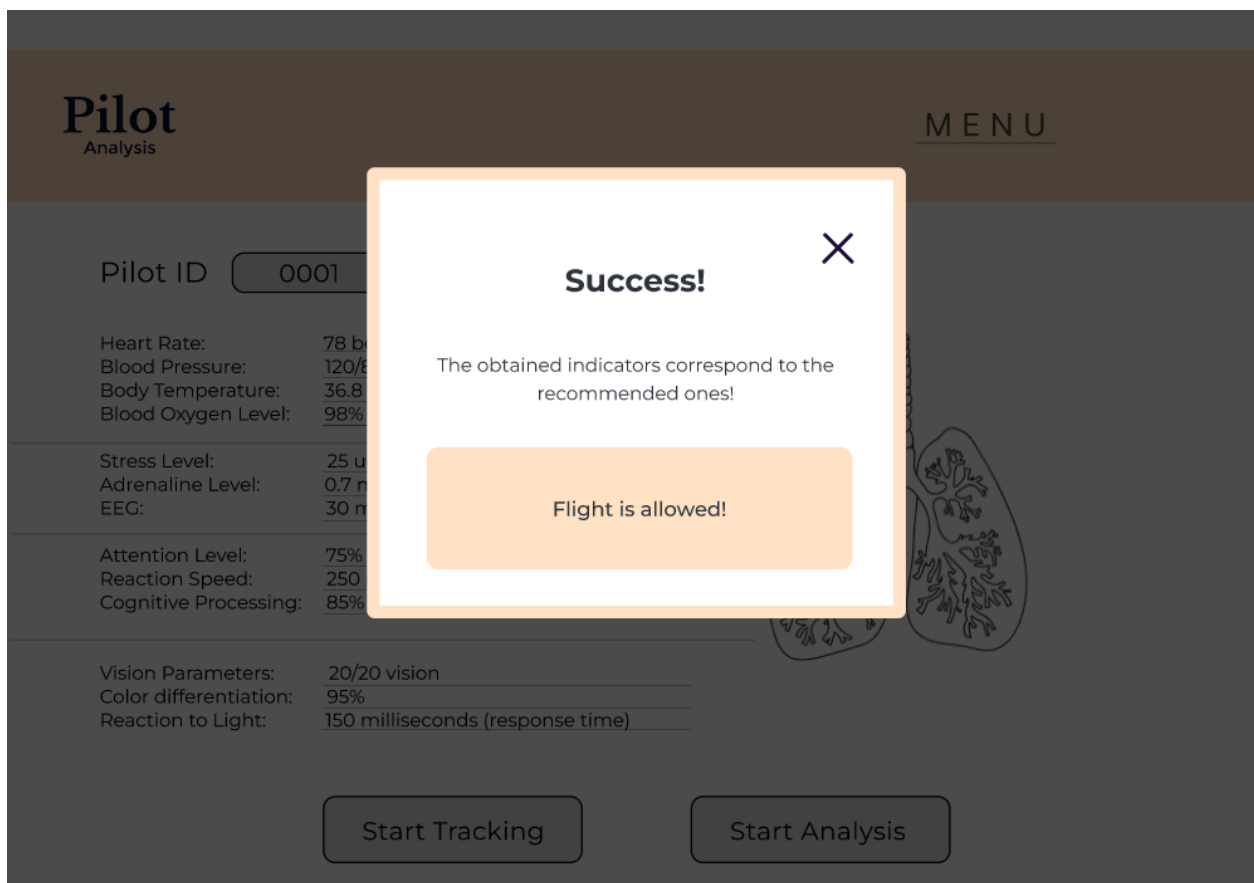


Рис. 4.11. Виліт дозволено

Здоров'я пілота є ключовим фактором при визначенні його готовності до польоту, та у випадку, коли усі фізичні показники відповідають нормам безпеки, система допускає пілота до виконання польотів. Це забезпечує високий рівень безпеки та забезпечує, що пілот може виконувати свої обов'язки з врахуванням його фізичного здоров'я та загального стану.



4.4. Рекомендації до програмного засобу

В даній роботі був описаний програмний засіб, призначений для обробки та аналізу медичних даних в хмарному середовищі, використовуючи сервіси *Amazon Web Services (AWS)*.

Розроблений програмний засіб дозволяє зберігати, аналізувати, інтегрувати та використовувати медичні дані для покращення діагностики, обробки та аналізу медичних записів за допомогою наступних параметрів:

- обробка даних: розроблений програмний засіб дозволяє ефективно обробляти великі обсяги медичних даних та витягати корисну інформацію з них. Використання сервісів *AWS* спрощує обробку даних та полегшує доступ до них;

- масштабованість та надійність: *AWS* надає можливість масштабування ресурсів в залежності від потреб, що дозволяє ефективно використовувати ресурси та забезпечувати надійність обробки даних;

- автоматизація процесів: використання *AWS Lambda* дозволяє автоматизувати процеси створення, налаштування та керування ресурсами, що полегшує адміністрування інфраструктури;

- оптимізація витрат: важливо враховувати оптимізацію витрат, видаляючи зайві ресурси та використовуючи ресурси ефективно.

Рекомендовані наступні кроки для подальшого розвитку програмного засобу контролю передпольотного стану пілота зважаючи на продовжений розвиток проблеми авіаційних польотів:

- розширення функціональності: для подальшого розвитку програмного засобу рекомендується розширювати його функціональність. Можливі напрямки розвитку включають розширення можливостей аналізу медичних даних та інтеграцію з іншими сервісами *AWS*;

- забезпечення безпеки даних: при роботі з медичними даними важливо дотримуватися стандартів безпеки та конфіденційності даних;



– застосування технологій штучного інтелекту для аналізу великих обсягів даних, щоб удосконалити розпізнавати відхилення та попереджень про можливі проблеми.

– постійна робота над підвищенням кібербезпеки та впровадженням механізмів шифрування для захисту конфіденційної інформації;

– інтеграція з сенсорами та пристроями *IoT* для отримання розширеної інформації та покращення збору даних;

– використання *VR* та *AR* для створення іммерсивного середовища передпольотного контролю, що може симулювати різні умови та сценарії;

– дослідження та впровадження систем для виявлення емоційного стану пілота для додаткового аналізу психофізіологічних показників;

– створення мобільного додатку для пілотів, який дозволяє отримувати дані передпольотного контролю та рекомендації в реальному часі;

– розробка системи для збору зворотного зв'язку від пілотів щодо функціоналу та ефективності передпольотного контролю;

– розробка системи візуалізації даних для зручного сприйняття результатів передпольотного контролю;

– додавання функцій розпізнавання обличчя та голосу для автентифікації та визначення емоційного стану пілота.

Загальний опис програмного засобу та висновки надають уявлення про розроблений інструмент для обробки медичних даних та надають рекомендації для подальшого розвитку та використання, адже розроблений в даній роботі програмний засіб легко інтегрується з іншими системами за рахунок міросервісної архітектури. За необхідності можливо збільшити вхідні параметри програми для більш точного прогнозування передпольотного стану.



4.5. Висновки до розділу

У розділі 4 розглянуто та проаналізовано ключові аспекти нашого розробленого програмного засобу і його взаємодію з різними сервісами *Amazon Web Services (AWS)*.

В результаті виконання кваліфікаційної роботи був успішно розроблено програмний засіб передпольотного контролю стану пілота який дозволяє контролювати стан пілота перед вильотом та надає необхідну інформацію для прийняття рішення про допуск пілота до польоту. Цей продукт відіграє важливу роль у підвищенні безпеки авіаційних перельотів.

Програмний засіб успішно інтегрується з різними сервісами *AWS*, включаючи *Amazon S3*, *Amazon Athena*, *Amazon Comprehend Medical* та *AWS Lambda*. Ця інтеграція дозволяє зберігати, обробляти та аналізувати медичні дані з високою ефективністю.

Розроблено захист даних за допомогою *AWS Key Management Service (KMS)*, що надає керування ключами шифрування для захисту конфіденційної інформації пілотів та лікарів. Шифрування даних та зберігання у сховищах даних важливо для захисту від несанкціонованого доступу. Всі ключі зберігаються та управляються в безпечному середовищі *AWS*, зменшуючи ризик витоку ключів.

AWS Lambda успішно використовується для автоматичного створення та керування різними ресурсами, такими як бази даних, таблиці та запити. Це спрощує розгортання та конфігурування системи.

Програмний засіб має значний потенціал для подальшого розвитку. Було визначено та описано майбутні кроки для його поліпшення. Цей розвиток може включати розширення функціональності, інтеграцію з системами авіації та рівня безпеки польотів.

Загалом, розроблений програмний засіб є інструментом для підвищення безпеки авіаційних перельотів, і має значний потенціал для подальшого успішного впровадження та подальшого розвитку.



ВИСНОВКИ

Кваліфікаційна робота є важливим кроком у розробці та впровадженні програмного засобу передпольотного контролю стану пілота, який має велике значення для безпеки авіаційних перельотів. Робота представляє собою результат докладного дослідження, проєктування та реалізації цього програмного засобу, враховуючи низку ключових аспектів.

У даній роботі детально було розглянуто та описано психологічні аспекти пілотажу, оскільки передпольотний контроль вимагає особливої уваги до перевірки психічного та фізичного станів пілота.

Аналізуючи існуючі рішення, що вирішують проблеми передпольотного контролю стану пілотів, прийшли до висновку, що на даний момент стандартизовані підходи аналізу станів полота не враховують рушійних показників для глибокого та чіткого передпольотного аналізу.

Детально обрані інструменти та технології забезпечили високоефективну роботу програмного засобу та глибокий аналіз стану пілота. Мова програмування *Python* добре підходить для реалізації аналітичних алгоритмів, обробки даних та швидкого прототипування.

Зазначені функціональні вимоги встановлюють широкий спектр операцій, дій та можливостей, які передпольотний засіб контролю стану пілота повинен здійснює. Ці вимоги відображають основні завдання та функції програмного засобу, визначаючи його функціональну область дії. Вони визначили рушійні параметри для глибокого аналізу, проєктування та розробки системи, забезпечуючи врахування та задоволення необхідних потреб та очікувань користувачів.

Розробка та впровадження алгоритмів для аналізу фізичного та психологічного стану пілотів забезпечили здатність автоматично оцінювати фізичний та психологічний стан пілота, що оптимізувало працездатність програмного засобу та забезпечила високий рівень безпеки під час польотів.

Системи аналізу можуть виявляють фізичні або психологічні проблеми, що дозволяє передбачати аварійні ситуації. Розроблені алгоритми передпольотного



контролю орієнтовані на індивідуальні особливості кожного пілота, створюючи персоналізовані методи управління та рекомендації щодо підтримки здоров'я.

Однією з основних складових дипломної роботи є сам розроблений програмний засіб, який дозволяє вчасно контролювати та аналізувати стан пілота перед вильотом. Це важливо для забезпечення оптимальних умов для пілота і підвищення безпеки авіаційних перельотів.

Окрему увагу слід приділити інтеграції програмного засобу з сервісами *Amazon Web Services (AWS)*. Вона дозволила ефективно зберігати, обробляти та аналізувати медичні дані пілотів, використовуючи різні *AWS*-сервіси, включаючи *Amazon S3*, *Amazon Athena*, *Amazon Comprehend Medical* та *AWS Lambda*. Ця інтеграція спростила роботу з даними та ресурсами, роблячи систему більш ефективною. Не менш важливим було використання *AWS Lambda* для автоматичного керування ресурсами. Це дозволило спростити розгортання та налаштування програмного засобу та його складових.

Один із головних висновків полягає в виявленні значного потенціалу для подальшого розвитку дипломної роботи. Потенціал включає розширення функціональності програмного засобу та подальшу інтеграцію з системами авіації для підвищення безпеки та зручності управління авіаційними перельотами.

Загальний висновок полягає в тому, що дипломна робота була успішною у всіх аспектах, і її результати мають велике практичне застосування у сфері авіаційної безпеки. Програмний засіб передпольотного контролю стану пілота є важливим інструментом для підвищення безпеки авіаційних перельотів, рекомендований до розширення функціоналу за рахунок зручної структури і має вагомую роль у цій сфері.



СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бойченко С.В., Іванченко О.В. Положення про дипломні роботи (проекти) випускників Національного авіаційного університету. – К.: НАУ, 2017. – 63 с.
2. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення. – 39 с.
3. Endsley M. R., *Handbook of Aviation Human Factors: Second Edition*, 2016. – 12.1-12.22 с.
4. Lee S., Eun Y., Kim H., Kim S., Lee D., *Developing an mHealth app for post-stroke upper limb telerehabilitation: an open source solution*, 2016. – 56 с.
5. Murthi D., Sharma S., *Mobile health platform for post-stroke care at home. Pervasive and Mobile Computing*, 2015. – 11 с.
6. Johnson T., Broach M., Richman J., *An integrated approach to pilot workload assessment using a fusion of eye tracking, fNIRS and performance measures. Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 2017. – 202 с.
7. Hockey G. R. J., *A state control theory of adaptation. In Vigilance: Theory, Operational Performance and Physiological Correlates*, 2016.
8. Rasmussen J., *Skill, rules and knowledge; signals, signs, and symbols, and other distinctions in human performance models*, 2017.
9. Norman D. A., Draper S. W., *User Centered System Design: New Perspectives on Human-Computer Interaction*, 2015.
10. Wickens C. D., *Multiple resources and performance prediction. Theoretical Issues in Ergonomics Science*, 2016. – 198 с.
11. Dekker S., *Human factors and ergonomics in the aviation domain. Journal of Cognitive Engineering and Decision Making*, 2015.
12. Moller J. A., Smith B. L., *Analysis of Pilot Physiology: Information Needs for Design of Future General Aviation Cockpits. Aviation, Space, and Environmental Medicine*, 2016.



13. *The Convention on International Civil Aviation* [Електронний ресурс] – URL: https://www.icao.int/documents/annexes_booklet.pdf.
14. *Junichiro H., Yuda E., Pitfalls of assessment of autonomic function by heart rate variability*. 2019.
15. Про затвердження Правил медичного забезпечення польотів державної авіації України [Електронний ресурс] – URL: <https://zakon.rada.gov.ua/laws/show/z1287-15#Text>
16. *Amazon DynamoDB* [Електронний ресурс] – URL: <https://aws.amazon.com/dynamodb/>
17. *pandas: powerful Python data analysis toolkit* [Електронний ресурс] – URL: <https://pypi.org/project/pandas/>
18. *AWS Glue for Spark and AWS Glue for Ray* [Електронний ресурс] – URL: <https://docs.aws.amazon.com/glue/latest/dg/how-it-works-engines.html>
19. *Amazon S3 storage documentation* [Електронний ресурс] – URL: <https://docs.netapp.com/us-en/bluexp-s3-storage/>
20. Апаратура для реєстрації електрокардіограми [Електронний ресурс] – URL: <https://compendium.com.ua/uk/clinical-guidelines-uk/cardiology-uk/section-5-uk/glava-1-osnovi-elektrokardiografii/>
21. *Amplify Documentation* [Електронний ресурс] – URL: <https://docs.amplify.aws>
22. *Amazon DynamoDB* [Електронний ресурс] – URL: <https://www.techtarget.com/searchaws/definition/Amazon-Dynamo-Database-DDB>
23. Гіпертонія і серцево судинні захворювання [Електронний ресурс] – URL: <https://www.pfizerpro.com.ua/therapy-areas/cardiovascular/hypertension-cardio>
24. *Mental Health and Sleep* [Електронний ресурс] – URL: <https://www.sleepfoundation.org/mental-health>
25. *Features of Python* [Електронний ресурс] – URL: <https://www.prepbytes.com/blog/python/features-of-python/>



Лістинг коду

```
# Алгоритм перевірки фізичного стану
```

```
class Sensor:
```

```
    def __init__(self, name):
```

```
        self.name = name
```

```
    def get_sensor_data(self):
```

```
        # Simulate data retrieval from the sensor
```

```
        return {"value": 0, "unit": ""}
```

```
class PhysicalConditionCheckAlgorithm:
```

```
    def __init__(self, pilot):
```

```
        self.pilot = pilot
```

```
        self.heart_rate_sensor = Sensor("Heart Rate Sensor")
```

```
        self.blood_pressure_sensor = Sensor("Blood Pressure Sensor")
```

```
        self.body_temperature_sensor = Sensor("Body Temperature Sensor")
```

```
        self.oxygen_level_sensor = Sensor("Oxygen Level Sensor")
```

```
    def analyze_heart_rate(self):
```

```
        # Retrieve heart rate data from the sensor
```

```
        heart_rate_data = self.heart_rate_sensor.get_sensor_data()
```

```
        # Analyze heart rate data and assess its significance
```

```
        if heart_rate_data["value"] > 100:
```

```
            print(f"Warning: Elevated heart rate detected ({heart_rate_data['value']}
```

```
bpm).")
```

```
    def analyze_blood_pressure(self):
```

```
        # Retrieve blood pressure data from the sensor
```



```
blood_pressure_data = self.blood_pressure_sensor.get_sensor_data()
```

```
# Analyze blood pressure data and assess its significance
```

```
if blood_pressure_data["systolic"] > 140 or blood_pressure_data["diastolic"] >
```

```
90:
```

```
    print(f"Warning: High blood pressure detected ({blood_pressure_data}).")
```

```
def analyze_body_temperature(self):
```

```
    # Retrieve body temperature data from the sensor
```

```
    body_temperature_data = self.body_temperature_sensor.get_sensor_data()
```

```
# Analyze body temperature data and assess its significance
```

```
if body_temperature_data["value"] > 99.5:
```

```
    print(f"Warning: Elevated body temperature detected  
({body_temperature_data['value']}°F).")
```

```
def analyze_oxygen_level(self):
```

```
    # Retrieve oxygen level data from the sensor
```

```
    oxygen_level_data = self.oxygen_level_sensor.get_sensor_data()
```

```
# Analyze oxygen level data and assess its significance
```

```
if oxygen_level_data["value"] < 95:
```

```
    print(f"Warning: Low oxygen level detected ({oxygen_level_data['value']}%).")
```

```
def integrate_physical_algorithms(self):
```

```
    # Integrate individual physical analysis algorithms
```

```
    # Consider interplay of heart rate, blood pressure, body temperature, and oxygen
```

```
level
```

```
    pass
```



```

def check_individual_physical_indicators(self):
    # Perform individualized analysis of physical indicators
    self.analyze_heart_rate()
    self.analyze_blood_pressure()
    self.analyze_body_temperature()
    self.analyze_oxygen_level()

def check_overall_physical_condition(self):
    # Perform overall analysis of physical condition
    # Integrate individual physical analysis results
    self.integrate_physical_algorithms()

def execute_physical_condition_check(self):
    # Execute the entire physical condition check algorithm
    # Include checks for individual parameters and overall condition
    self.check_individual_physical_indicators()
    self.check_overall_physical_condition()

# Example Usage
# Create a pilot object with relevant physical data
pilot_data = {
    "heart_rate": 78,
    "blood_pressure": {"systolic": 120, "diastolic": 80},
    "body_temperature": 98.4,
    "oxygen_level": 97,
}

pilot = Pilot(**pilot_data)
# Use the PhysicalConditionCheckAlgorithm to analyze the pilot's physical condition
physical_condition_checker = PhysicalConditionCheckAlgorithm(pilot)

```



physical_condition_checker.execute_physical_condition_check(

Алгоритм аналізу психологічних параметрів:

import numpy as np

Define a class for the psychological analysis algorithm

class PsychologicalAnalysisAlgorithm:

def __init__(self):

Initialize any necessary parameters or variables

self.threshold_stress = 70

self.threshold_adrenaline = 0.8

def analyze_psychological_parameters(self, pilot_data):

Analyzes psychological parameters of the pilot.

– pilot_data: Dictionary containing psychological data.

– result: Dictionary containing the analysis result.

result = {}

Extract psychological parameters

stress_level = pilot_data['stress']

adrenaline_level = pilot_data['adrenaline']

eeg_data = pilot_data['eeg']

Analyze individual psychological parameters

stress_analysis = self.analyze_stress(stress_level)

adrenaline_analysis = self.analyze_adrenaline(adrenaline_level)

eeg_analysis = self.analyze_eeg(eeg_data)

Combine individual analyses into an overall assessment

*result['overall_assessment'] = self.combine_analyses(stress_analysis,
adrenaline_analysis, eeg_analysis)*



```
# Check for any specific conditions or triggers
result['recommendation'] =
self.generate_recommendation(result['overall_assessment'])
```

```
return result
```

```
def analyze_stress(self, stress_level):
```

```
    Analyzes the stress level of the pilot.
```

```
    Parameters:
```

```
    – stress_level: Numeric value representing the stress level.
```

```
    Returns:
```

```
    – analysis: String describing the stress analysis.
```

```
    """
```

```
    if stress_level > self.threshold_stress:
```

```
        return 'High stress level. Requires attention.'
```

```
    else:
```

```
        return 'Stress level within normal range.'
```

```
def analyze_adrenaline(self, adrenaline_level):
```

```
    Analyzes the adrenaline level of the pilot.
```

```
    Parameters:
```

```
    – adrenaline_level: Numeric value representing the adrenaline level.
```

```
    Returns:
```

```
    – analysis: String describing the adrenaline analysis.
```

```
    if adrenaline_level > self.threshold_adrenaline:
```

```
        return 'Elevated adrenaline level. Monitor closely.'
```

```
    else:
```

```
        return 'Adrenaline level within normal range.'
```



```

def analyze_eeg(self, eeg_data):
    average_alpha_wave = np.mean(eeg_data['alpha_wave'])
    if average_alpha_wave < 10:
        return 'Low alpha wave activity. Potential cognitive fatigue.'
    else:
        return 'Normal alpha wave activity.'

def combine_analyses(self, stress_analysis, adrenaline_analysis, eeg_analysis):
    if 'High' in (stress_analysis, adrenaline_analysis):
        return 'High Risk'
    else:
        return 'Low to Moderate Risk'

def generate_recommendation(self, overall_assessment):
    if overall_assessment == 'High Risk':
        return 'Ground the pilot due to high psychological risk.'
    else:
        return 'Proceed with pre-flight checks.'

# Example usage
algorithm_instance = PsychologicalAnalysisAlgorithm()
pilot_data = {
    'stress': 75,
    'adrenaline': 0.9,
    'eeg': {'alpha_wave': [8, 9, 10, 11, 12]}
}

result = algorithm_instance.analyze_psychological_parameters(pilot_data)
print(result)

```

1) Загальний алгоритм аналізу даних та прийняття рішення:
class ComprehensiveHealthAnalyzer:



```

def __init__(self, physical_data, psychological_data, cognitive_data):
    self.physical_data = physical_data
    self.psychological_data = psychological_data
    self.cognitive_data = cognitive_data

def analyze_health(self):
    # Аналіз фізичного стану
    physical_analyzer = PhysicalHealthAnalyzer(**self.physical_data)
    physical_result = physical_analyzer.check_physical_health()

    psychological_analyzer =
    PsychologicalStateAnalyzer(**self.psychological_data)
    psychological_result = psychological_analyzer.analyze_psychological_state()
    cognitive_analyzer = CognitiveAnalyzer(**self.cognitive_data)
    cognitive_result = cognitive_analyzer.analyze_cognitive_parameters()
    decision = {"flight_approval": True, "reasons": []}
    if physical_result["status"] == "Abnormal":
        decision["flight_approval"] = False
        decision["reasons"].append("Abnormal physical health.")
        if psychological_result["stress_level"] == "High" or
        psychological_result["adrenaline_level"] == "High":
            decision["flight_approval"] = False
            decision["reasons"].append("High psychological stress or adrenaline
            levels.")
        if psychological_result["eeg_results"] == "Abnormal":
            decision["flight_approval"] = False
            decision["reasons"].append("Abnormal EEG results.")
            if cognitive_result["attention_level"] == "Low" or
            cognitive_result["reaction_time"] > 1.0:
                decision["flight_approval"] = False
                decision["reasons"].append("Low cognitive performance.")
    return decision

```




```
physical_data = {"heart_rate": 75, "blood_pressure": {"systolic": 110, "diastolic":  
70}, "body_temperature": 98.6}  
psychological_data = {"cortisol_level": 60, "adrenaline_level": 90, "eeg_data":  
[0.5, 0.8, 0.3, 0.9, 0.2]}  
cognitive_data = {"attention_level": "Moderate", "reaction_time": 0.8}  
comprehensive_analyzer = ComprehensiveHealthAnalyzer(physical_data,  
psychological_data, cognitive_data)  
decision_result = comprehensive_analyzer.analyze_health()  
print("Flight Approval Decision:")  
print("Can the pilot fly?", decision_result["flight_approval"])  
1) print("Reasons:", decision_result["reasons"])
```

