

## Специфікація загальних вимог до системи ТРП в цілому

Ви- моги код	Вимоги опис
<b>A</b>	<b>Архітектура системи</b>
A1	Кожен компонент системи повинен являти собою окремий сервіс, що взаємодіє з іншими через HTTP / HTTPS протокол.
A2	Для передачі даних між компонентами (сервісами) системи повинні використовуватися POST-запити. Сервіси повинні мати можливість відповідати за всіма адресами свого API також і на GET-запити за допомогою HTTP-відповіді, але лише як повідомлення, що сервіс працює (обов'язково) або для відображення графічного інтерфейсу (якщо таке передбачено для даного сервісу).
A3	Графічні інтерфейси користувача системи повинні являти собою незалежні окремі додатки, які взаємодіють з API-сервісами, що виконують основну програмну логіку та зберігають дані.
A4	Загальний формат передачі даних між частинами системи – JSON (пріоритетний) або XML.
<b>B</b>	<b>Бази даних</b>
B1	Система може використовувати, відповідно до потреб конкретного компоненту, графові, реляційні та орієнтовані на документи бази даних.
B2	Дані онтологічного графу зберігаються в форматі XML, OWL або іншому вбудованому форматі, що передбачений СУБД, що його опрацьовує.
B3	Для зберігання і забезпечення доступу до OWL онтологій використовується база даних, що орієнтована на документи, наприклад MongoDB.
B4	Для роботи з онтологіями у форматі OWL слід застосовувати функції бібліотеки RDFLib (Python), або іншої бібліотеки, яка здатна виділяти набори RDF-трижок з OWL і виконувати запити на мові SPARQL.
B5	Модуль забезпечення роботи інтерфейсу користувача здійснює зберігання технічної інформації (дані про реєстрацію, сесії, поточний діалог, тощо) у реляційній базі даних (наприклад, PostgreSQL).

B6	Забезпечити можливість обміну даними компонентів системи з центральною базою даних (ЦБД) онтологічного типу через відкритий прикладний програмний інтерфейс (далі – API).
B7	Забезпечити можливість внесення інформації до ЦБД системи через свій інтерфейс українською мовою з можливістю використання латинських літер та числових символів.
<b>C</b>	<b>Інтерфейс користувача</b>
C1	Інтерфейс користувача повинен бути графічним, зрозумілим і зручним у використанні.
C2	Забезпечити варіанти графічного інтерфейсу користувача у наступних видах: web-сторінка, що відображається у браузері; бот в існуючій системі обміну повідомленнями (якщо доцільно), мобільний додаток, що підтримує зв'язок з сервісами через API.
C3	Мова інтерфейсу користувача – українська.
<b>D</b>	<b>Мова і термінологія</b>
D1	Назви хвороб і ступенів функціональності, що містяться в системі та її базах даних, повинні відповідати Міжнародній класифікації функціонування, обмежень життєдіяльності та здоров'я.
D2	У компонентах системи допускається вживання стандартних символів латинського (англійського) алфавіту у наступних випадках: <ul style="list-style-type: none"> <li>– наведення імені чи назви мовою оригіналу;</li> <li>– наведення латинських назв;</li> <li>– написання математичних виразів та хімічних формул;</li> <li>– назви брендів, торгівельних марок, компаній;</li> <li>– посилання на веб-ресурси;</li> <li>– e-mail адреси.</li> </ul>
<b>E</b>	<b>Програмний інтерфейс (API)</b>
E1	Кожен агент та сервіс, що входить у систему, повинен мати програмний інтерфейс, що може взаємодіяти з іншими програмами за допомогою HTTP-запитів.
E2	В основу API усіх агентів і сервісів, які входять в дану систему, повинні бути покладені POST-запити.

<b>F</b>	<b>Безпека інформації</b>
F1	Доступ до інтерфейсу адміністратора надається тільки після введення відповідного логіну і паролю.
F2	Доступ до внесення змін до баз даних потребує окремого паролю і окремих прав адміністратора бази даних.
F3	У разі втрати паролю користувач має можливість змінити його, але тільки через відправлення електронного листа, на e-mail, який було вказано ним під час реєстрації.
F4	При обміні повідомленнями між агентами і сервісами системи повідомлення повинні містити пароль і імена сервісів відправника та одержувача.
F5	Програмні файли системи зберігаються на сервері в окремому каталозі, доступ до редагування вмісту якого має лише користувач з правами адміністратора.
F6	Доступ до інтерфейсу адміністратора здійснюється за окремим таємним url, посилання на який не повинно міститись ніде в інтерфейсі користувача (незалежно від статусу авторизації).
F7	Система повинна відповідати вимогам законодавства України «Про захист інформації в інформаційно-телекомунікаційних системах», «Про захист персональних даних».
<b>G</b>	<b>Надійність системи</b>
G1	Система TISP має гарантувати одночасну роботу не менше ніж 100 користувачів без збоїв.
G2	Виникнення помилки при роботі з певним користувачем не повинно призводити до зупинення, зависання чи руйнування всієї системи.
<b>H</b>	<b>Реєстрація і авторизація</b>
H1	Для реєстрації користувачеві достатньо вказати логін, e-mail та пароль.
<b>I</b>	<b>Робота з API</b>
I1	До всіх сервісів системи можна звернутися через їх API при вказуванні відповідних параметрів запиту, що може бути потрібним для емуляції їх роботи в системі в процесі тестування.

## **Б2 ТВ на допоміжні програмні модулі хмарної платформи**

Модуль "Реєстратура" призначений для забезпечення можливості створення та оновлення Електронної медичної картки пацієнта в системі Медичної інформаційної системи (МІС), а також для забезпечення взаємодії з базою даних Національної служби здоров'я України (НСЗУ) [7, 59, 63, 75]. Цей модуль дозволяє внести коди МКХ (Міжнародної класифікації хвороб) та МКФ (Міжнародної класифікації функціонування, інвалідності та здоров'я) для попередньої каталогізації медичної інформації.

При використанні цього модуля можливо додати інформацію про палату, в якій перебуває пацієнт, та про лікаря, який його лікує, для забезпечення зручності та систематизації даних. Крім того, в модулі передбачена таблиця завантаження ліжкового фонду, яка допомагає контролювати доступність ліжок у медичних закладах.

Відображення запланованих та вже проведених подій у зручній таблиці з відображенням формату їх проведення онлайн чи офлайн. Це дозволяє медичному персоналу швидко оцінити та зіставити дані, а також вчасно реагувати на будь-які зміни в графіку обстежень.

Календарна інтеграція створює зручний спосіб організації робочого часу, а також дозволяє планувати обстеження, враховуючи наявність вільних слотів. Це допомагає уникнути конфліктів у графіку та мінімізувати затримки.

Модуль також пропонує розширений функціонал для перегляду та прослуховування різних форматів медичної інформації, таких як зображення, відео, біосигнали та звук. Це дозволяє медичному персоналу аналізувати та інтерпретувати результати обстежень більш ефективно та точно.

Загалом, модуль "Додаткові обстеження" є потужним інструментом у системі здоров'я, який допомагає медичному персоналу краще організувати процес обстежень, зберігати медичні дані та забезпечує зручний доступ до інформації для подальшого аналізу і прийняття рішень.

**Модуль "Експертна система Підтримки прийняття рішень"** включає опис та додаткову інформацію (посилання на публікації) [7,59, 63,70], про при-

значення, можливості, назви модулів та сервісів у яких застосовуються розроблені додаткові оптимізаційні, Особлива функція "Бронювання ліжка в реабілітаційному центрі" дозволяє забезпечити можливість заздалегідь забронювати місце для пацієнта в центрі реабілітації, що допомагає зменшити час очікування та покращити організацію надання послуг.

Також, модуль "**Реєстратура**" забезпечує можливість переведення пацієнта з одного відділення до іншого, а також здійснення оплати за надані медичні послуги. Ці функції роблять процес організації лікування більш гнучким та зручним.

Крім того, модуль дозволяє пацієнту записатись на додаткові консультації та обстеження, що забезпечує йому більш широкий доступ до медичних послуг та можливість більш детального обстеження стану здоров'я.

В цілому, модуль "Пацієнт" інтегрує різноманітні функції, що сприяють ефективній роботі медичного закладу та полегшують отримання пацієнтом якісної та своєчасної медичної допомоги [7, 59, 63,71].

**Модуль "Додаткові обстеження"** розроблений з метою забезпечення можливості вибору пацієнта, призначення додаткових консультацій та обстежень, а також уявлення запланованих та проведених подій в зручній таблиці нижче та в календарі. Головна функція модуля - допомогти медичному персоналу ефективно керувати і наглядати за процесами додаткових обстежень.

Важливою можливістю модуля є можливість вибору пацієнта, що дозволяє зручно знаходити та працювати з інформацією про конкретного пацієнта. Призначення додаткових консультацій та обстежень відбувається в систематизованому і зрозумілому форматі, що полегшує роботу медичного персоналу.

Модуль також надає можливість лінгвістичні, прогностичні, тощо сервіси.

**Модуль "Методичного забезпечення"** передбачає внесення корисної методичної та нормативної інформації для роботи учасників реабілітаційного процесу, медичних працівників та пацієнтів.

**Модуль "Календар"** – це важлива складова системи, яка забезпечує зру-

чний та організований спосіб відображення запланованих подій у реабілітаційному процесі. Його головна мета полягає в тому, щоб допомогти запланувати події щоденно, а також на тиждень, з урахуванням обраного інтервалу дат для різних учасників процесу реабілітації.

Цей модуль забезпечує зручний графічний інтерфейс, який дозволяє користувачам легко оглядати та керувати запланованими подіями. Він показує календарні події на основі різних переглядів, таких як щоденний та тижневий перегляд, що дає можливість збільшити ефективність планування та координації подій.

Кожен учасник реабілітаційного процесу має власний персоналізований календар, де він може бачити лише свої заплановані події, що забезпечує конфіденційність та обмежує доступ до інформації. Це особливо корисно, якщо в процесі реабілітації беруть участь різні фахівці, пацієнти, а також члени сім'ї, і кожен з них має власний розклад дій.

Загалом, модуль "Календар" є потужним інструментом, який допомагає організувати реабілітаційний процес, збільшує його ефективність та допомагає забезпечити гармонійну співпрацю між різними учасниками. Він спрощує планування, координацію та нагляд над подіями, забезпечуючи успішну реалізацію реабілітаційних заходів і сприяючи покращенню стану пацієнтів.

**Модуль "Ефективність"** – це інтегрована система, що дозволяє збирати та аналізувати інформацію про функціональні, психологічні проби та тести, проведені з метою визначення ефективності діяльності, здібностей чи можливостей користувачів. Цей модуль надає зручний і легкий спосіб внесення даних про проведені випробування та зберігання їх в системі [59, 70].

У модулі передбачено структуровані поля для введення різноманітної інформації про функціональні та психологічні проби і тести, що включають результати, дати проведення, методологію, використані засоби, а також інші важливі параметри. Це дозволяє користувачам системи легко вносити дані і зберігати всю потрібну інформацію в одному місці.

Однак головною перевагою цього модуля є можливість перегляду історії

попередньо проведених функціональних, психологічних проб та тестів. Ця функція дозволяє здійснювати аналіз результатів в часовому ракурсі, що в свою чергу може допомогти зрозуміти динаміку змін показників з плином часу, виявити тенденції та зробити висновки щодо ефективності діяльності або змін в психологічних станах користувачів.

**Модуль "Працівники"** має задовольняти декілька важливих вимог для оптимального функціонування. Передусім, цей модуль повинен надавати можливість відображення інформації про працівників згідно з обраними фахівцями та за їхнім прізвищем. Очевидно, що даний функціонал є дуже важливим для забезпечення швидкого та зручного доступу до необхідної інформації про персонал компанії чи організації [59, 65 – 69].

Система повинна дозволяти користувачеві відбирати певні фахівців, використовуючи критерії пошуку, наприклад, область знань, спеціалізацію, рівень освіти або інші характеристики. Після вибору критеріїв, система повинна ефективно фільтрувати базу даних працівників та відображати лише тих, хто відповідає вказаним параметрам.

Забезпечення безпеки та конфіденційності даних також є невід'ємною частиною вимог до цього модуля. Доступ до інформації про працівників повинен бути обмеженим лише для авторизованих користувачів, забезпечуючи, що конфіденційні дані не потраплять в руки несанкціонованих осіб.

Узагалі, вимоги до модуля "Працівники" повинні забезпечувати швидкий, зручний та безпечний доступ до інформації про працівників з можливістю фільтрації та сортування за обраними критеріями, а також забезпечення конфіденційності даних. Це допоможе зробити роботу зі зброєю людських ресурсів більш ефективною та продуктивною..

**Модуль "Відділ кадрів закладу"** має відповідати наступним вимогам щодо свого функціоналу:

Додавання та реєстрація працівників в базі даних трудових ресурсів підприємства (БД ТРП). Це означає, що модуль має забезпечити можливість вводити особисту інформацію про нових працівників, таку як ім'я, прізвище, кон-

тактні дані, посада, дата прийому на роботу тощо. Дані про працівників повинні автоматично синхронізуватися з НСЗУ та зберігатися в БД ТРП.

Внесення графіку роботи працівників та їх функціональних обов'язків. Модуль повинен дозволяти визначити розклад роботи для кожного працівника, включаючи робочі дні, години початку і закінчення робочого дня, а також перерви. Додатково, необхідно мати можливість визначити функціональні обов'язки кожного працівника, що включають його основні обов'язки та можливі додаткові відповідальності.

Можливість передачі функціональних обов'язків з одного працівника на інший на визначений термін. Модуль повинен забезпечити можливість делегування функціональних обов'язків між працівниками з вказанням періоду, на який вони будуть передані. Це дозволить ефективно керувати завданнями та враховувати можливі відсутності або інші обставини.

Загалом, функціонал модуля "Відділ кадрів" має спростити та автоматизувати процеси управління персоналом, дозволяючи зосередитися на ефективному використанні ресурсів підприємства і забезпечити точну і актуальну інформацію про працівників.

**Модуль Адміністратор** – це важлива складова системи, яка має виконувати ряд функцій для ефективного управління закладом. У цьому модулі необхідно врахувати роботу з наступними вкладками та їх функціоналом:

**Сервіси:** Модуль повинен дозволяти адміністратору додавати, редагувати та видаляти інформацію про надані сервіси. Це можуть бути, наприклад, медичні процедури, послуги допомоги, або інші дії, які заклад надає своїм клієнтам.

**Послуги:** Адміністратор повинен мати можливість керувати переліком послуг, які надаються закладом. Це може включати додавання нових послуг, редагування інформації про послуги та їхнє видалення.

**Підрозділи закладу:** Модуль має дозволяти адміністратору керувати структурою закладу, додавати нові підрозділи, видаляти або змінювати існуючі. Це може стосуватися різних підрозділів, таких як відділення, лабораторії, кабінети та інші.



Налаштування посад: Адміністратор повинен мати можливість керувати переліком посад в закладі. Він може додавати нові посади, встановлювати права та обов'язки для кожної посади і змінювати інформацію про вже існуючі.

Мультидисциплінарні команди: Модуль повинен надавати можливість адміністратору формувати та керувати мультидисциплінарними командами. Це можуть бути групи фахівців різних спеціальностей, які співпрацюють над конкретними клієнтами або проектами.

Загалом, Модуль Адміністратор повинен бути зручним та функціональним інструментом для управління різними аспектами діяльності закладу. Він допомагає забезпечити ефективну організацію роботи, оптимізацію процесів та забезпечення безпеки даних та доступу до них.

**Модуль Системного адміністратора** – це важлива закладка системи, яка відповідає за забезпечення можливості внесення інформації про лікувальний заклад і Ліцензійні обсяги працівників. Його функціонал спрямований на допомогу адміністраторам та відповідальним особам у веденні актуальних та точних даних про медичні заклади та кваліфікацію медичного персоналу.

Серед ключових вимог до Модуля Системного адміністратор можна виділити:

Доступ до інформації про лікувальний заклад: Модуль має забезпечити можливість внесення та збереження детальної інформації про медичний заклад. Це може включати такі дані, як назва закладу, його адреса, контактні дані, графік роботи, а також інші реквізити, необхідні для ідентифікації та опису закладу.

Зручний інтерфейс: Важливо, щоб інтерфейс Модуля Системного адміністратора був зрозумілим та зручним для користувачів. Інтуїтивно зрозумілі елементи управління та легка навігація допоможуть прискорити процес внесення та редагування даних.

Захист даних: З урахуванням чутливості інформації, яка зберігається у Модулі Системного адміністратора, важливо забезпечити надійний захист даних. Механізми аутентифікації, ролевого доступу та шифрування допоможуть

уникнути несанкціонованого доступу до інформації.

Звіти та аналітика: Модуль може також надавати можливість генерувати звіти та аналітику зі зібраної інформації. Це дозволить відповідальним особам отримувати всебічну статистику щодо роботи медичних закладів та ефективності персоналу.

Загалом, Модуль Системного адміністратора грає важливу роль у підтримці ефективного функціонування медичних закладів, допомагаючи зберігати актуальну та точну інформацію про заклади та їх персонал. Його функціонал повинен бути надійним, зручним у використанні та забезпечувати адекватний рівень безпеки даних.

**Модуль Реєстрації закладу** є інтегральною частиною системи організації та контролю медичних закладів і має на меті забезпечити ефективне внесення та оновлення інформації про Лікувальний заклад. Цей модуль гарантує безперервний доступ до даних про заклад, який зареєстровано в Національній службі здоров'я України (НСЗУ), а також забезпечує синхронізацію з Базою Даних Телереабілітаційної платформи [62,63].

Серед вимог до Модуля Реєстрації закладу слід виокремити наступні:

Внесення інформації про Лікувальний заклад: Модуль має дозволяти внести в систему всю необхідну інформацію про заклад, таку як його назва, місце знаходження, контактні дані, профіль медичних послуг тощо.

Оновлення інформації: Медичні заклади можуть зазнавати змін, і, отже, модуль повинен давати можливість оновлювати дані про заклад, коли це необхідно. Це може стосуватися зміни контактної інформації, додаткових послуг, персоналу тощо.

Інтеграція з НСЗУ: Модуль повинен мати можливість передавати дані про реєстрацію та зміни до НСЗУ, щоб організація могла проводити аналіз та контроль за діяльністю медичних закладів на рівні країни.

Синхронізація з Телереабілітаційною платформою: База даних Модуля Реєстрації повинна бути постійно узгодженою з БД Телереабілітаційної платформи, щоб забезпечити доступ до актуальної інформації про заклади для цієї

платформи.

Забезпечення безпеки даних: Модуль має дотримуватися високих стандартів безпеки, щоб захистити конфіденційну інформацію про медичні заклади від несанкціонованого доступу, злому або втрати.

Інтерфейс для адміністраторів: Модуль повинен мати зручний і легкий у використанні інтерфейс для адміністраторів закладів, що дозволяє зручно та ефективно здійснювати внесення та редагування інформації.

Підсумовуючи, Модуль Реєстрації закладу відповідає за впорядковане зберігання та оновлення інформації про медичні заклади в Національній службі здоров'я України та забезпечення синхронізації з Базою Даних Телереабілітаційної платформи, забезпечуючи тим самим актуальність та достовірність даних про заклади, що допомагає поліпшити організацію медичної допомоги та контроль за її наданням.

**Модуль Авторизації** є ключовою частиною Телереабілітаційної платформи, і його основна мета - забезпечити безпеку та контроль доступу до системи. Вимоги до цього модуля включають наступні функціональні можливості:

Створення профіля: Модуль повинен дозволяти користувачам створювати особисті профілі в системі. При створенні профілю, користувач повинен вказати необхідні особисті дані, такі як ім'я, прізвище, адреса, контактні дані та інші релевантні інформаційні поля.

Оновлення профілю: Користувачі повинні мати можливість оновлювати свої профілі, наприклад, змінювати контактні дані, додавати або змінювати інформацію про медичний стан, а також змінювати свої налаштування в системі.

Автентифікація та авторизація: Модуль має забезпечувати процес автентифікації користувачів з використанням надійних методів, наприклад, через пароль або двофакторну автентифікацію. Після автентифікації модуль здійснює авторизацію, тобто контролює доступ до різних функцій та даних системи, забезпечуючи, що користувачі мають доступ лише до тих даних, які їм дозволено переглядати та редагувати.

Інтеграція з Національною службою здоров'я України (НСЗУ): Модуль

повинен передбачити можливість зв'язку з системою НСЗУ, щоб надавати доступ до відповідних медичних даних користувачам та забезпечувати обмін інформацією з цим органом.

Робота з базою даних: Модуль повинен зберігати та управляти інформацією про користувачів, їх профілі та інші деталі, необхідні для функціонування платформи. Це означає, що модуль має підтримувати базу даних, забезпечувати безпеку зберігання і передачі даних та забезпечувати швидкий та ефективний доступ до інформації.

Забезпечення конфіденційності даних: Оскільки платформа обробляє медичні дані, модуль повинен дотримуватись найвищих стандартів забезпечення конфіденційності та захисту персональних даних користувачів.

Запобігання несанкціонованому доступу: Модуль має впроваджувати заходи безпеки для запобігання несанкціонованому доступу до системи, а також має забезпечувати захист від злону або зловживання.

Загалом, Модуль Авторизації є критично важливим для забезпечення безпеки та функціональності Телереабілітаційної платформи, дозволяючи користувачам впевнено та зручно користуватись усіма її можливостями, забезпечуючи одночасно захист цінних медичних даних.

**Модуль Багатомовність**— це компонент або програмне забезпечення, яке ставить перед собою завдання забезпечити можливість використання додатка або середовища із підтримкою кількох мов. Основними вимогами до такого модуля є забезпечення зручності та легкості використання для користувача, незалежно від його мовних навичок.

Динамічна зміна мови: Модуль має забезпечувати зміну мови в режимі реального часу без необхідності перезавантаження додатка. Користувач повинен мати змогу змінити мову, і всі елементи інтерфейсу та тексти автоматично адаптуватимуться до обраної мови.

Підтримка локалізації: Модуль повинен дозволяти використовувати різні мовні ресурси для локалізації додатка. Це означає, що переклади, формати дати, часу, валюти та інші мовні аспекти повинні бути легко змінювати без необхід-

ності втручання в код додатка.

Враховуючи ці вимоги, Модуль Багатомовність забезпечить користувачів з усього світу можливістю насолоджуватися продуктивним та зручним досвідом використання програмного забезпечення або додатка, незалежно від їхньої мовної приналежності.

**СПЕЦИФІКАЦІЯ ВИМОГ ДО БЕЗПЕКИ ТА НАДІЙНОСТІ**  
**ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ ПІДСИСТЕМИ**

## Технічні вимоги до системи MedRehabBot

### Специфікація функціональних вимог

RQкод	RQ опис
<b>F1</b>	<b>Архітектура системи</b>
F1.1	<p>Система повинна бути гнучкою, адаптованою до швидко змінюваних умов і вимог користувача. Тому її ядро, або основна частина, має бути розроблене на основі модульної архітектури. Це означає, що кожна окрема компонента системи, або модуль, має відповідати за конкретний аспект її функціональності. Основні функції:</p> <ul style="list-style-type: none"><li>– обробка і аналіз текстових повідомлень;</li><li>– формування запитів до бази даних/знань;</li><li>– взаємодія з базою даних/знань;</li><li>– формування структури даних для відповіді;</li><li>– взаємодія і координація роботи наявних модулів.</li></ul>
F1.2	<p>Можливість інтеграції системи з іншими системами і сервісами. Взаємодія із зовнішнім графічним інтерфейсом користувача та забезпечення інтерфейсу прикладного програмування (API) має здійснюватися окремими модулями.</p>
F1.3	<p>Обмін даними між різними компонентами системи має відбуватися за допомогою формату JSON, який є стандартним форматом для передачі структурованих даних через мережу. Використання JSON як єдиного формату забезпечує консистентність у взаємодії компонентів системи, що спрощує процес розробки та інтеграції.</p>
F1.4	<p>Графічним інтерфейс користувача має бути представлено у вигляді Телеграм-боту. Має бути забезпечена надійна інтеграція з API Telegram для прийому та відправки повідомлень в режимі реального часу. Наявність функціональної можливості розпізнавати команди від користувачів, які надіслані через Телеграм-бота, і передавати їх до відповідних компонентів системи для подальшої обробки. Всі обмінні дані між системою і Телеграм-ботом мають бути захищені, зокрема через викорис-</p>

	тання автентифікаційних токенів і шифрування з'єднань. Усі запити та відповіді між системою і Телеграм-ботом мають реєструватися, що дозволить відстежувати та аналізувати історію спілкування для покращення сервісу.
F1.5	Модулі, що працюють з графічним інтерфейсом користувача (Телеграм-бот) і API повинні бути такими, що легко замінюють один одного без потреби внесення змін до компонентів ядра.
F1.6	Налаштування та параметри системи вносяться та зберігаються у окремі конфігураційні файли, які підтримують формати JSON або XML. Конфігураційні файли мають мати чітко визначену структуру, яка забезпечує легкість читання та редагування параметрів. Важливі параметри, такі як ключі доступу або паролі, мають бути захищені або зовнішнім шифруванням, або специфічними механізмами зберігання. Система має забезпечити легкий доступ до конфігураційних файлів для адміністраторів, але обмежити доступ для інших користувачів. Система має забезпечити легкий доступ до конфігураційних файлів для адміністраторів, але обмежити доступ для інших користувачів. Система повинна підтримувати автоматичне створення резервних копій конфігураційних файлів. Зміни у конфігураційних файлах повинні бути враховані системою без необхідності перезавантаження або перезапуску сервісів.
F1.7	Система повинна працювати як серверний додаток, запущений у фоновому режимі. Система має забезпечувати стабільність та неперервність роботи 24/7, автоматично відновлюючись після збоїв або перебоїв живлення. Система має вести журнал подій, який допоможе відстежувати та аналізувати роботу системи.
F1.8	Система повинна підтримувати можливість безперебійних оновлень, дозволяючи вносити зміни без зупинки сервісу.
F1.9	Система повинна оптимізувати використання процесорного часу, пам'яті та інших ресурсів, щоб не заважати іншим додаткам.
F1.10	Допускається застосування і приєднання до системи зовнішніх програмних пакетів та бібліотек, створених сторонніми розробниками, якщо їх ліцензія дозволяє їх вільне використання.
<b>F2</b>	<b>Бази даних / знань</b>



F2.1	Система має використовувати, відповідно потреб певного компоненту, графові, реляційні, документно-орієнтовані та інші бази даних і системи управління базами даних (залежно від специфіки задачі, система може використовувати також ключ-значення, колоночні, об'єктно-орієнтовані бази даних та ін.). Незалежно від типу бази даних, всі дані повинні бути захищені від несанкціонованого доступу, збоїв і втрат.
F2.2	Залежно від вибраного типу бази даних, система має забезпечити оптимальний запит до бази, ефективне зберігання та відновлення даних.
F2.3	Система повинна мати засоби для інтеграції різних баз даних, дозволяючи їм взаємодіяти між собою та обмінюватися даними.
F2.4	Дані онтологічного графу, що є основною базою знань системи, повинні бути представлені та зберігатися у форматі OWL, що є стандартом для представлення онтологій у Вебі. Всі сутності і предикати онтології не тільки повинні відповідати формату OWL, але й дотримуватися всіх його специфікацій і обмежень. Це забезпечує стабільність, надійність та сумісність бази знань з іншими системами.
F2.5	Обов'язкове використання інструментів для валідації онтологічного графу, щоб перевіряти, чи відповідають сутності та предикати стандарту OWL та чи немає в них порушень або протиріч.
F2.6	<p>База знань системи може бути розподіленою та розділеною в межах графової бази даних:</p> <ul style="list-style-type: none"> <li>– розподілена структура бази знань може забезпечити швидкий доступ до даних, оскільки запити можуть оброблятися паралельно в різних частинах бази.</li> <li>– система повинна дозволяти організацію знань не як одну масивну OWL-онтологію, а як сукупність менших пов'язаних онтологій. Такий підхід сприяє більш швидкому доступу до конкретних блоків інформації та гнучкому управлінню змінами в структурі знань.</li> <li>– при розподіленні або розділенні онтологій, важливо забезпечити, щоб взаємозв'язки між різними частинами бази знань зберігалися належним чином, і інформація могла бути відновлена в повному обсязі.</li> <li>– OWL-онтології онтологій, вони повинні бути легко інтегровані.</li> </ul>
F2.7	Отримані результати запитів до онтологічної бази знань мають

	зберігатися (кешуватися) у реляційній СУБД під управлінням PostgreSQL. Це дозволяє швидко отримувати дані без необхідності звернення до первинної бази при повторних запитах. Система повинна мати механізми автоматичного оновлення кешованих даних, щоб вони відповідали актуальному стану онтологічної бази. Для оптимізації продуктивності може бути застосована стратегія кешування з використанням "Термін придатності", де старі записи автоматично видаляються через певний час, або при досягненні максимального обсягу кешу.
F2.8	<p>У якості системи для зберігання OWL-онтологій та виконання запитів до неї має використовуватися Apache Jena Fuseki. У випадку, якщо Apache Jena Fuseki не задовольнить потреби проєкту, може бути використана будь-яка інша СУБД з відкритим вихідним кодом, яка підтримує завантаження даних у форматі OWL (RDF/XML).</p> <p>Обрана СУБД повинна підтримувати виконання запитів на мові SPARQL, що дозволяє здійснювати гнучке та ефективне пошукове запитання до онтологічних даних.</p>
F2.9	Мовою запитів до бази знань є SPARQL.
F2.10	Окрім OWL формату, для зберігання графу онтології допускається використовувати графову СУБД Neo4j. В цьому разі формальні запити будуються на мові Cypher.
F2.11	Онтології, що містяться у системі, мають бути розділені за вузькою тематикою певної предметної галузі.
F2.12	Має бути забезпечена автоматизована взаємодія системи із онтологічною базою знань та допоміжною СУБД PostgreSQL.
F2.13	OWL-онтологія повинна мати чітку наперед визначену структуру для забезпечення уніфікації набору шаблонів SPARQL запитів, що не повинна значно та без зворотної сумісності змінюватися при подальшому розвитку системи.
F2.14	Системи управління базою знань і реляційною базою даних мають працювати як окремі незалежні сервіси.
<b>F3</b>	<b>Інтерфейс користувача</b>
F3.1	Система має забезпечувати графічний інтерфейс користувача, що дає можливість спілкуватися із системою природною мовою у текстово-

	му режимі. У якості такого інтерфейсу може виступати зовнішній сервіс обміну повідомленнями, наприклад, Телеграм-бот.
F3.2	Графічний інтерфейс користувача може бути окремим застосунком, що взаємодіє із системою через API.
F3.3	Обробка повідомлень користувача має протидіяти потенційно шкідливим для системи повідомленням, як-то занадто довгі тексти, невідомі символи, часта відправка вхідних або вихідних повідомлень, тощо.
F3.4	Для створення графічного інтерфейсу користувача і для інтерфейсу роботи з базами даних і базами знань системи можуть залучатися готові рішення, в тому числі такі, що йдуть у комплектації відповідних СУБД та фреймворків.
<b>F4</b>	<b>Мова і термінологія</b>
F4.1	Назви хвороб і ступенів функціональності, що містяться в системі та її базах, даних повинні відповідати Міжнародній статистичній класифікації хвороб і проблем, пов'язаних зі здоров'ям (МКХ-10, англ. International Statistical Classification of Diseases and Related Health Problems, ICD-10) та Міжнародній класифікації функціонування, обмежень життєдіяльності та здоров'я (МКФ, англ. International Classification of Functioning, Disability and Health, ICF), відповідно.
F4.2	У компонентах інтерфейсу системи допускається вживання спеціалізованих символів, а також посилання на веб-ресурси, e-mail адреси, посилання на документи, зображення та інші медіа матеріали.
F4.3	Назви на екзотичних мовах, що використовують не латинські і не кириличні символи (китайська, японська, арабська, іврит, тощо), при необхідності їх наведення (пункт F4.2 цієї специфікації), транслітеруються символами латинського (англійського) алфавіту.
<b>F5</b>	<b>Інтерфейс прикладного програмування (API)</b>
F5.1	Система повинна мати інтерфейс прикладного програмування (API).
F5.2	В основу взаємодії через API повинні бути покладені POST-запити.
F5.3	Форматом передачі даних до API, та отримання відповіді JSON.

F5.4	API має забезпечити роботу у режимах передачі «сирого» тексту вхідного повідомлення у вигляді значення певного поля у JSON структурі, що передається та виконання певного запиту до бази знань із переданим набором параметрів.
<b>F6</b>	<b>Призначення системи</b>
F6.1	Система призначена для супроводження комплексних телереабілітаційних заходів шляхом інтерактивної інформаційно-довідкової підтримки на всіх етапах реабілітаційного циклу, починаючи від формулювання реабілітаційної цілі і прогнозу, закінчуючи методологічними рекомендаціями і супроводженням процесів реабілітації.
F6.2	Функції, які виходять за рамки предметної області, означеної у пункті F6.1 даної специфікації, не передбачається даною системою.
F6.3	Потенційними користувачами системи є лікарі (насамперед у галузі фізичної та реабілітаційної медицини), студенти медичних і суміжних з медичними спеціальностей та пацієнти (в режимі телереабілітації).

*Специфікація вимог до безпеки та надійності*

<b>RQко д</b>	<b>RQ опис</b>
<b>F1</b>	<b>Безпека інформації</b>
F1.1	Доступ до онтологічної бази знань та допоміжної реляційної СУБД має бути реалізовано через систему аутентифікації на основі логіна та пароля. Доступ до даних через графічний інтерфейс повинен вимагати введення логіна та пароля.
F1.2	Після декількох спроб введення неправильного пароля, система повинна блокувати користувача на визначений час або повідомляти адміністратора про спробу несанкціонованого доступу.
F1.3	Кожна база даних і база знань, що функціонує в системі потребує окремого паролю.
F1.4	При інтеграції з іншими системами або при доступі до бази знань за допомогою API, також повинна використовуватися система аутентифікації на основі логіна та пароля. Всі повідомлення, в яких не містяться або не є вірним пароль ігноруються.

F1.5	<p>Пароль доступу до системи через API має зберігатися у зашифрованому вигляді (як hash-код, з якими зіставляється введений пароль після його хешування за тим самим алгоритмом). Всі додаткові конфіденційні дані, мають зберігатися у зашифрованому вигляді, використовуючи сучасні методи шифрування.</p>
F1.6	<p>Програмна взаємодія із інтерфейсом користувача (зокрема, Телеграм-ботом) здійснюється з використанням відповідного секретного токена, що не розповсюджується.</p>
F1.7	<p>Паролі можуть зберігатися у базі даних, у коді програми, або в окремих файлах на сервері, але лише у вигляді hash-кодів.</p>
F1.8	<p>Запити до API, що не відповідають встановленому формату передачі даних ігноруються.</p>
F1.9	<p>Доступ по SSH до сервера, на якому зберігаються і працюють компоненти системи, потребує авторизації з окремим логіном і паролем або SSH-ключем.</p>
F1.10	<p>Програмні файли системи зберігаються на сервері в окремому каталозі, доступ до редагування вмісту якого має лише користувач з правами адміністратора.</p>
F1.11	<p>Логін і пароль адміністратора у всіх компонентах системи, які того потребують, не повинен бути очевидним і тривіальним, а саме:</p> <ul style="list-style-type: none"> <li>– не дозволяється використовувати у якості логіну слова “admin”, “root”, “postgres”, “main”, тощо;</li> <li>– не допускаються паролі типу: “1111”, “123456”, “qwerty”, “abcd” і тому подібні тривіальні поширені комбінації символів;</li> <li>– не допускається, щоб логін і пароль співпадали;</li> <li>– рекомендовано для більшої надійності використовувати у якості паролю комбінації літер різного регістру, цифр і символів у кількості не менше 8-ми. При цьому не слід допускати комбінацію латинських і кирилических символів через схожість багатьох літер. Використовувати лише стандартні латинські літери і цифрові символи.</li> </ul>
F1.12	<p>Заборонено використовувати проміжні інтерфейси авторизації користувачів із правами адміністрування і редагування баз даних і без знань, крім веб-сторінки авторизації, яка надходить з даної системи.</p>

F1.13	Автоматичне спрямування (redirect) може здійснюватися тільки на адреси, які належать до домену даної системи, за виключенням спеціально оговорених, перевірених адрес.
F1.14	Система повинна відповідати вимогам законодавства України "Про захист інформації в інформаційно-телекомунікаційних системах", "Про захист персональних даних" та інших нормативно-правових актів, що регулюють питання захисту інформації, у тому числі в інформаційно-телекомунікаційних системах.
<b>F2</b>	<b>Надійність системи</b>
F2.1	Система має гарантувати стабільну роботу системи при одночасному доступі багатьох користувачів. Користувачі повинні мати можливість працювати паралельно без взаємних перешкод.
F2.2	Система повинна бути стабільною, щоб уникнути непередбачуваних збоїв, незалежно від кількості одночасних запитів
F2.3	Виникнення помилки при роботі з певним користувачем не повинно призводити до зупинення, зависання чи руйнування всієї системи і якимось чином негативно впливати на роботу системи з іншими користувачами.
F2.4	Час відклику системи не повинен суттєво збільшуватися незалежно від обсягу завантаження, що гарантує відсутність лагів та затримок для кінцевого користувача.
F2.5	Збої у роботі програмної системи не повинні призводити до безповоротної втрати інформації з баз даних та пошкодженню програмних та конфігураційних файлів.
F2.6	Використання системи балансування навантаження для розподілення запитів між серверами.