Звіт з предмету: «Аналіз вимог до програмного забезпечення»

Виконав:

Группа ПЗ19 - 1ду

Циганков Артур

Лабораторна робота **№1**

**Виявлення високорівневих вимог і розробка документу "Бачення"**

**Мета роботи:** набуття навичок виявлення високорівневих вимог та розробки документу “Бачення” на їх основі.

**Теоретичні основи*.***

Одним з перших документів, який розробляється в ході будь-якого проекту з розробки інформаційної системи, є документ “Бачення” (технічне завдання, бачення), що описує на високому рівні призначення системи, контекст її використання, підходи до вирішення поставлених завдань, обмеження і стратегічні рішення щодо реалізації проекту. Цей документ задає межі проекту і забезпечує загальне бачення системи з боку всіх зацікавлених сторін.

З метою однозначного розуміння вимог до програмного продукту створюється специфікація вимог (SRS), що містить високодеталізовані функціональні та нефункціональні вимоги, поведінкові моделі, ескізи користувача інтерфейсу і т.д. За домовленістю із Замовником можуть бути використані різні види специфікацій: Use Case Specification , User Story , Specification by Example та інші.

Що дасть виявлення вимог:

Узгоджений перелік бізнес-вимог , вимог зацікавлених осіб і вимог до рішення;

1. Можливість визначити пріоритети , за якими система буде розроблятися і впроваджуватися;
2. Базу для оцінки трудомісткості і загальних витрат по проекту.

Що дасть розробка документу **“Бачення”:**

1. Високорівневий опис рішення відповідно до загальновизнаних стандартів розробки бізнес-аналітичних документів;
2. Розуміння зацікавленими особами кордонів рішення (що рішення має забезпечувати і що знаходиться поза рамками рішення);
3. Можливість планування робіт і виконання подальших дій з розробки рішення.

Кроки, які необхідно пройти для формування документу "Бачення":

* Формулювання проблем
* Ідентифікація співвласників
* Визначення меж системи
* Ідентифікація обмежень
* Формулювання постановки завдань
* Визначення можливостей системи
* Оцінка результатів

Визначення меж системи є нетривіальним процесом. Для цього використовують контекстні діаграми.

Джерела вимог:

* Законодавство;
* Вимоги стандартів;
* Бізнес-процеси;
* Очікування та бачення користувачів системи.

Шаблон документу "Бачення" містить наступні основні розділи:

I. ВСТУП

* Огляд продукту
* Мета
* Межі
* Посилання
* Означення та абревіатури

II. ЗАГАЛЬНИЙ ОПИС

* Перспективи продукту
* Функції продукту
* Характеристики користувачів
* Загальні обмеження
* Припущення й залежності

III. КОНКРЕТНІ ВИМОГИ

* Вимоги до зовнішніх інтерфейсів
  + [Інтерфейс користувача](https://uk.wikipedia.org/wiki/Інтерфейс_користувача#_blank)
  + [Апаратний інтерфейс](https://uk.wikipedia.org/w/index.php?title=Апаратний_інтерфейс&action=edit&redlink=1#_blank)
  + [Програмний інтерфейс](https://uk.wikipedia.org/wiki/Програмний_інтерфейс#_blank)
  + [Комунікаційний протокол](https://uk.wikipedia.org/wiki/Комунікаційний_протокол#_blank)
  + Обмеження пам'яті
  + Операції
  + Функції продукту
  + Припущення й залежності
  + Властивостi програмного продукту
* Атрибути програмного продукту
* [Надійність](https://uk.wikipedia.org/wiki/Надійність#_blank)
* [Доступність](https://uk.wikipedia.org/wiki/Доступність#_blank)
* [Безпека](https://uk.wikipedia.org/wiki/Інформаційна_безпека#_blank)
* [Супроводжуваність](https://uk.wikipedia.org/wiki/Супроводжуваність#_blank)
* [Переносимість](https://uk.wikipedia.org/w/index.php?title=Переносимість_програмного_забезпечення&action=edit&redlink=1#_blank)
* [Продуктивність](https://uk.wikipedia.org/wiki/Продуктивність#_blank)

IV. ДОДАТКОВІ МАТЕРІАЛИ

У додаток виносяться атрибути можливостей. Рекомендується наступний набір атрибутів: статус, вигода, об'єм робіт, ризик, стабільність, цільовий випуск, призначення, причина.

**Бачення**

***Мета:*** Мета створення цього документа полягає в тому, щоб зібрати, проаналізувати та визначити високорівневі потреби і можливості системи та редагування розкладу y музичній школі з різними рівнями доступу. Документ акцентує увагу на можливостях, необхідних адміністрації, вчителям та учням, і на те, чому ці потреби існують. Подробиці того, як система складання розкладу виконує ці потреби, будуть деталізовані в реализації повного розкладу.

Контекст

Справжній документ розробляється в рамках проекту розробки системи прийому у лікарні.

Визначення, акроніми та скорочення

Основні визначення наведені в документі «Словник проекту».

**Короткий зміст**

Документ описує високорівневі вимоги до системи складання розкладу у музичній школі. Вказані основні ділові переваги розглянутого в Баченні рішення, сформульовані ключові проблеми та способи їх вирішення, наведено характеристики користувачів системи, можливості системи, обмеження, показники якості та інші вимоги до продукту.

Позиціонування. Ділові переваги

Ми позиціонуємо цю систему як найкращій засіб для прийому в лікарні.

**Визначення проблеми**

|  |  |
| --- | --- |
| Проблема | Автоматизування прийому у лікарні. |
| Зачіпає | Затримки виконання прийому у лікарні |
| Успішне вирішення | Зменшення часу, що витрачаеться на процесс прийому у лікарні |

**Визначення позиції виробу**

Створюеться для розповсюження у Лікарнях. Потрібно розробити систему створення автоматизованого процесу прийому у лікарнях. Данна система має назву АІС «АвтоЛіки», котра заснована на СУБД та автоматизованій системі, на відміну від застарілого підходу до створення процесу прийому у лікарнях.

Описання користувачів. Відомості про користувачів

У системи існують три основних актора: Черговий лікар, Лікуючий лікар та медсестра.

Черговий лікар — Оформлює пацієнтів, оформлює історію хвороб, повідомляє рідних.

Лікуючий лікар — Проводить лікування, керує медсестрами, може проводити перш-черговий огляд.

Медсестра — Виконує вказівки Лікуючого Лікаря, може проводити перш-черговий огляд та дивитися історію хвороб.

Призначена для користувача середа

Розумний час додавання та редагування даних - не більше 3 хвилин. Операційна система: IOS, Android, Microsoft Windows, Linux, MacOS.

**Профілі користувачів**

|  |  |
| --- | --- |
| Типовий представник | **Черговий ліккар** |
| Опис | Користувач системи, наділений правами на прийом пациєнтів та оформлення історій хвороб, та повідомлення рідних. |
| Тип | Користувач |
| Відповідальності | Оформлення пацієнтів, оформлення історії хвороб, повідомлення рідних. |
| Критерій успіху | Швидке та своечасне надання послуг лікаря |

|  |  |
| --- | --- |
| Типовий представник | **Лікуючий лікар** |
| Опис | Користувач системи, наділений правами на керування процесом лікування |
| Тип | Користувач |
| Відповідальності | Керування процесом лікування, керування медсестрами |
| Критерій успіху | Швидке та своечасне надання послуг лікаря |

|  |  |
| --- | --- |
| Типовий представник | **Медсестра** |
| Опис | Користувач системи, наділений правами перегляду історії хвороб, надання перш-чергового перегляду. |
| Тип | Користувач |
| Відповідальності | Виконання вказівок лікуючого лікаря |
| Критерій успіху | Швидке та своечасне надання послуг медсестри |

**Ключові потреби користувачів**

Пацієнт повинен отримати швидку та своечасну допомогу послуг персоналу лікарні.

**Контекст використання системи**

Система є закінченою незалежною розробкою. У перспективі можливе використання системи в комплексі з бізнес системами.

Припущення і залежності

Немає

Можливості продукту. Структурований опис замовлення

Можливість вносити до бази данних новий персонал.

Обмеження

Впровадження системи не повинно займати більше 1 місяця.

У ядрі системи повинна бути представлена СУБД реляційного доступу.

Показники якості. Застосовність

Індивідуально.

Надійність

* Доступність - час, що витрачається на обслуговування системи не повинно перевищувати 1% від загального часу роботи.
* Середній час безвідмовної роботи – 98% годин у місяць.
* Максимальна норма помилок або дефектів - 1 помилка на десять тисяч рядків коду.

Інші вимоги до виробу. Застосовувані стандарти

Система повинна відповідати всім стандартам інтерфейсу користувача Операційна система: IOS, Android, Microsoft Windows, Linux, MacOS.

Експлуатаційні вимоги

Система повинна бути здатна підтримувати мінімум 60 одночасно працюючих користувачів, пов'язаних із загальною базою даних і мати можливість збільшити їх кількість на випадок збільшення штату медичного персоналу.

Вимоги до документації. Керівництво користувача

В системі повинні бути представлені Керівництва користувачів (за типами користувачів). Вони повинні містити розшифровку всіх використовуваних термінів, описи основних варіантів використання, включаючи альтернативні сценарії, а також детальний огляд інтерфейсу програми.

Інтерактивна довідка

Інтерактивна довідка необхідна для вирішення виниклих під час роботи питань. У довідці має бути реалізована можливість пошуку інформації за ключовими словами, а також варіант подання інформації по окремих позиціях меню програми. Довідка повинна містити максимально повну і детальну інформацію по роботі системи.

Маркування та пакетування

Система буде поширюватися через лікарні, у яких буде знаходитися сама система, а також інтерактивна довідка, керівництво по установці і керівництво користувача до неї.

Інсталяційна програма повинна включати загальну ліцензійну угоду, і, інформацію про авторські права.

Лабораторна робота **№2**

**Виявлення вимог користувачів, акторів і варіантів використання**

**Мета роботи:** набуття навичок виявлення і описання вимог користувачів до розробленого ПЗ; визначення основних акторів і формування варіантів використання.

**Теоретичні основи*.***

Хороші вимоги починаються з хороших першоджерел. Пошук таких якісних джерел первинної інформації - найважливіше завдання, що має, по щастю, безліч ресурсів для свого виконання. Приклади джерел вимог включають:

1. Замовників або клієнтів;
2. Користувачів;
3. Адміністраторів і технічний персонал;
4. Партнерів;
5. Експертів предметної області;
6. Промислових аналітиків;
7. Інформацію про конкурентів;

Після того, як ви встановите ці джерела вимог, ви можете використовувати ряд стратегій (методів) для збору вимог:

1. Провести сеанс “мозкового штурму”;
2. Інтерв'ювати користувачів;
3. Розповсюдити анкети;
4. Вникнути в оточення цільової організації;
5. Вивчити аналогічні системи;
6. Дослідити пропозиції та звіти про проблеми;
7. Спілкуватися з командами супроводу;
8. Вивчити поліпшення, зроблені користувачами;
9. Виявити непередбачене використання;
10. Проводити семінари;

Яку методику слід застосовувати залежить від ряду чинників:

1. Доступність і розташування зацікавлених осіб;
2. Знання командою розробників проблемної області;
3. Знання клієнтів і користувачів про проблемну область;
4. Знання клієнтів і користувачів про процес і методи розробки;

Результатом виявлення вимог є реєстр вимог. Вимоги співвласників зазвичай оформляються в простій письмовій формі, без якої-небудь особливої регламентації. Найпопулярнішим і дуже ефективним способом підвищення інформативності вимог є оформлення їх у вигляді варіантів використання (use case), який запропоновано Якобсоном.

Перш, ніж приступити до специфікації вимог у формі варіантів використання рекомендується виявити реєстр акторів (actors) і варіантів використання.

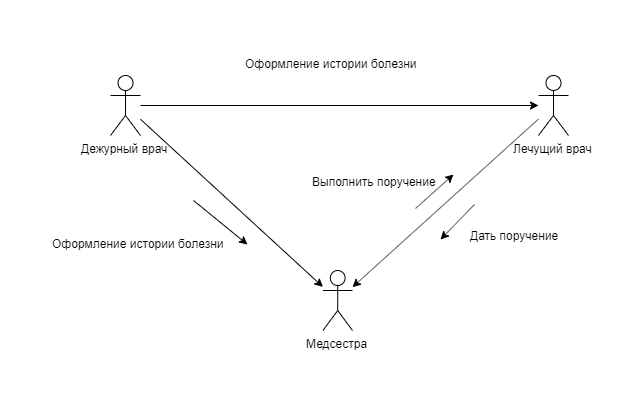
Актор - це хтось або щось, що має активність по відношенню до програмної системи. Окрім користувача в якості актора може розглядатися інша програмна система, апаратний пристрій, у ряді випадків - активна компонента самої системи.

Варіант використання в першому наближенні можна розглядати просто - як функцію, що реалізовується системою. Проте, сучасний погляд на організацію бізнесу говорить про те, що всяка функція повинна мати цінність для кінцевого споживача продукту або послуги.

Існують різні шаблони опису варіантів використання:

1. Вільний формат
2. Повний формат (запропонований А. Коберном)
3. Таблиця в дві колонки
4. Таблиця в три колонки
5. Псевдокод
6. Діаграма активності UML
7. Інші графічні моделі

Пошук акторів і варіантів використання

**Виявлення акторів*****Мал. 1. Аналіз акторів системи***

Інтерв'ю, проведене з зазначеними вище кандидатами показало, що вони припускають використовувати розроблювану АІС у пропоновому порядку. Це дозволило узагальнити ці ролі, **див. Мал. 1.**

***Короткий опис акторів представлено в табл. 1.***

Табл. 1. Виявлення акторів

|  |  |
| --- | --- |
| *Актор* | *Короткий опис* |
| Черговий лікар | Оформлює пацієнтів, оформлює історію хвороб, повідомляє рідних. |
| Лікуючий лікар | Проводить лікування, керує медсестрами, може проводити перш-черговий огляд. |
| Медсестра | Виконує вказівки Лікуючого Лікаря, може проводити перш-черговий огляд та дивитися історію хвороб. |

**Виявлення варіантів використання**

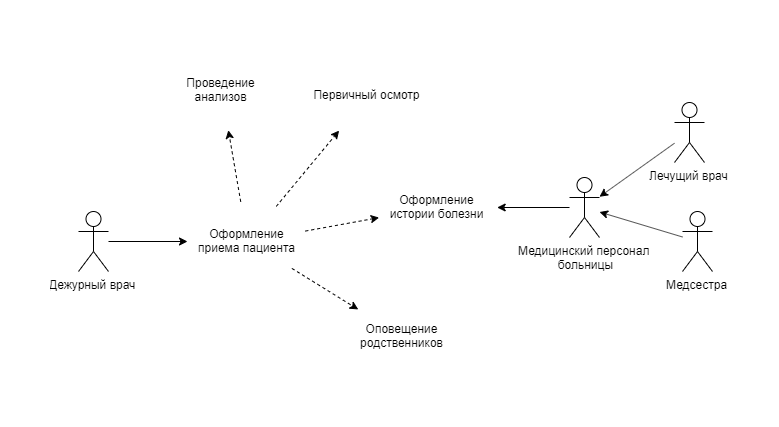
Виявлені варіанти використання зведені в таблицю 2.

Табл. 2. Виявлення варіантів використання

|  |  |  |
| --- | --- | --- |
| *Основний актор* | *Найменування* | *Формулювання* |
| Черговий лікар | Оформлення історії хвороб | Цей варіант використання дозволяє черговому лікарю оформлювати історію хвороб для лікуючого лікаря. |
| Черговий лікар | Оформлення прийому пацієнта | Цей варіант використання дозволяє черговому лікарю оформлювати прийом пацієнта |
| Черговий лікар | Повідомлення рідних | Цей варіант використання дозволяє черговому лікарю повідомляти рідних |
| Лікуючий лікар | Перегляд історії хвороб | Цей варіант використання дозволяє лікуючому лікарю переглядати історію хвороб |
| Лікуючий лікар | Проводження процесу лікування | Цей варіант використання дозволяє лікуючому лікарю проводити процес лікування |
| Лікуючий лікар | Проводження перш-черговий огляд | Цей варіант використання дозволяє лікуючому лікарю проводити перш-черговий огляд |
| Лікуючий лікар | Керування медсестрами | Цей варіант використання дозволяє лікуючому лікарю керувати медсестрами |
| Медсестра | Проводження перш-черговий огляд | Цей варіант використання дозволяє медсестрі проводити перш-черговий огляд |
| Медсестра | Виконання вказівок лікуючого лікаря | Цей варіант використання дозволяє медсестрі виконувати вказівки лікуючого лікаря |
| Медсестра | Перегляд історії хвороб | Цей варіант використання дозволяє медсестрі переглядати історію хвороб |

**Розробка діаграм варіантів використання**

Всі варіанти використання показані на Мал. 2.



*Мал. 2. Діаграма прецедентів системи*

Лабораторна робота **№3**

**Короткий опис варіантів використання**

**Мета роботи:** набуття навичок уточнення варіантів використання, конкретизація їх описів; складання реєстру функціональних вимог.

**Теоретичні основи.**

Діаграма прецедентів — в [UML](https://uk.wikipedia.org/wiki/UML), [діаграма](https://uk.wikipedia.org/wiki/Діаграма), на якій зображено відношення між *акторами* та *прецедентами* в системі. Також, перекладається як **діаграма варіантів використання**. Діаграма задумана так, щоб дати найбільш загальне уявлення про функціональність системи (її компоненти), не вдаючись до деталей взаємозв'язків функцій. Тому основний вид відношення, використовуваний в діаграмі, - асоціація між актором і варіантом використання.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

1. асоціації ([англ.](https://uk.wikipedia.org/wiki/Англійська_мова) *association relationship*)
2. включення ([англ.](https://uk.wikipedia.org/wiki/Англійська_мова) *include relationship*)
3. розширення ([англ.](https://uk.wikipedia.org/wiki/Англійська_мова) *extend relationship*)
4. узагальнення ([англ.](https://uk.wikipedia.org/wiki/Англійська_мова) *generalization relationship*)

При цьому загальні властивості варіантів використання можуть бути представлені трьома різними способами, а саме — за допомогою відношень включення, розширення і узагальнення.

**Асоціація (association)** – одне з фундаментальних понять в мові UML і в тій або іншій мірі використовується при побудові всіх графічних моделей систем у формі канонічних діаграм. Стосовно діаграм варіантів використання асоціація специфікує особливості взаємодії зовнішніх агентів і варіантів використання. На діаграмі варіантів використання, так само як і на інших діаграмах, відношення асоціації позначається суцільною лінією між зовнішнім агентом і варіантом використання. Ця лінія може мати деякі додаткові позначення, наприклад, ім'я і кратністьВключення ([англ.](https://uk.wikipedia.org/wiki/Англійська_мова) *include*) у мові UML — це різновид відношення залежності між базовим варіантом використання і його спеціальним випадком. При цьому відношенням залежності ([англ.](https://uk.wikipedia.org/wiki/Англійська_мова) *dependency*) є таке відношення між двома елементами моделі, при якому зміна одного елемента (незалежного) приводить до зміни іншого елемента (залежного).

Включення (include) — це різновид відношення залежності між базовим варіантом використання і його спеціальним випадком. Відношення включення встановлюється лише між двома варіантами використання і вказує на те, що задана поведінка для одного варіанту використання включається як складовий фрагмент в послідовність поведінки іншого варіанту використання.

Так, наприклад, відношення включення, направлене від варіанту використання "Надання кредиту в банку" до варіанту використання "Перевірка платоспроможності клієнта", вказує на те, що кожна реалізація першого варіанту використання завжди включає виконання другого варіанту використання. Графічно дане відношення позначається як відношення залежності у формі пунктирної лінії із стрілкою, направленою від базового варіанту використання до варіанту використання, що включається. При цьому дана лінія позначається стереотипом «**<<include>>**»

**Розширення (extend)**визначає взаємозв'язок базового варіанту використання з іншим варіантом використання, функціональна поведінка якого задіюється базовою не завжди, а лише при виконанні додаткових умов.

У мові UML відношення розширення між варіантами використання позначається як відношення залежності у формі пунктирної лінії із стрілкою, направленою від того варіанту використання, який є розширенням для базового варіанту використання. Дана лінія із стрілкою має бути помічена стереотипом «**<<extend>>**»

**Узагальнення(generalization)** позначається суцільною лінією із стрілкою у формі незафарбованого трикутника, яка вказує на батьківський варіант використання. Ця лінія із стрілкою має спеціальну назву — стрілка-узагальнення.

Шаблон повного опису варіанту використання по А. Коберну:

*1)Назва* <коротка фраза у вигляді дієслова в невизначеній формі доконаного виду, відбиваюча мета>

*2) Контекст використання* <уточнення мети, при необхідності - умови її нормального завершення>.

*3) Зона дії* <посилання на рамки проекту>. Наприклад - підсистема бухгалтерського обліку.

*4) Рівень* <один з трьох: узагальнений, цілі користувача, підфункції>. Автор задає зумовлену трирівневу класифікацію вимог, що в цілому відповідає класифікації вимог на бізнес-вимоги, вимоги користувачів і функціональні вимоги.

*5) Основна дійова особа* <ім'я ролі основного актора або його опис>.

*6) Учасники і інтереси* <список інших акторів-учасників прецеденту з вказівкою їх інтересів>.

*7)Передумова* <те, що очікується, вже має місце>.

*8) Мінімальні гарантії* <що гарантується акторам-учасникам>. Наприклад - у разі невдалої транзакції усі дані, що були в системі до її початку, зберігаються незмінними.

*9) Гарантії успіху* <що отримають актори-учасники у разі успішного досягнення мети>.

*10) Тригер* <те, що "запускає" варіант використання, зазвичай - подія в

часі>.

*11) Основний сценарій* <тут перераховуються кроки основного сценарію, починаючи від тригера і аж до досягнення гарантії успіху>.

Формат опису : <Номер кроку> <Опис дії>

*12) Розширення* <тут послідовно описуються усі альтернативні сценарії>.

Кожна з альтернатив прив'язана до кроку основного сценарію.

Формат опису: <Номер кроку. Номер розширення> <Умова>:<Дія або посилання на підпорядкований варіант використання>.

Будь-який з кроків основного сценарію може мати один або більше розгалужень. Кожне розгалуження оформляється у вигляді розширення. У блоці "Розширення" усі розширення описуються послідовно.

У разі, якщо альтернативний сценарій не вдається описати одним рядком - застосовується наступний формат:

Починаючи з рядка, наступного після опису розширення, йде опис його дій у форматі основного сценарію:

<Номер кроку. Номер розширення. Номер кроку розширення> <Дія>

Опис розширення закінчується описом виходу з розширення. Основні варіанти виходу з розширення: повернення до чергового по номеру кроку основного сценарію, закінчення прецеденту, перехід до іншого кроку основного сценарію.

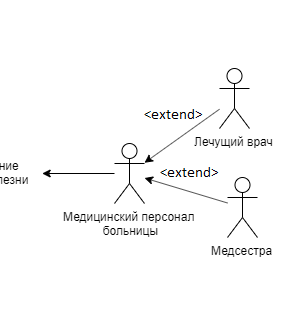
*Список змін в технології і даних* <що гарантується акторам- учасникам>. Наприклад - у разі невдалої транзакції усі дані, що були в системі до її початку, зберігаються незмінними.

*Допоміжна інформація* <додаткова інформація, корисна при описі варіанту використання>.

**Короткий опис варіантів використання**

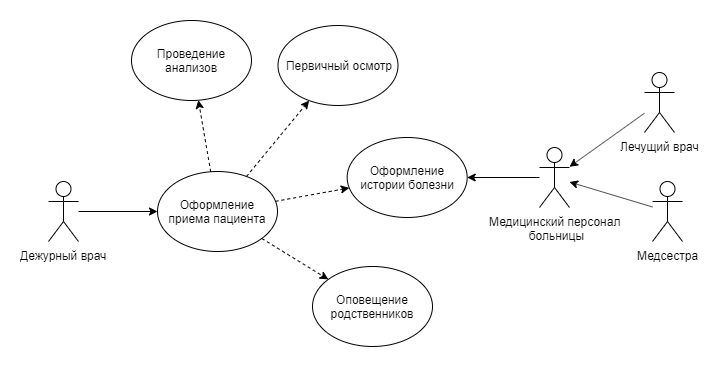
**Структуризація варіантів використання**

1. Варіант використання «Редагування» був пов“язанний у зв“язку з тим, що кожний підрозділ розкладу це різні таблиці у базі данних, які пов“язані між собою (див. Мал. 1.1).



*(Мал.1.1)Узагальнення варіантів використання редагування розкладу.*

Модифікована діаграма прецидентів



**Реєстр варіантів використання**

Табл. 1. Реєстр варіантів використання

|  |  |  |  |
| --- | --- | --- | --- |
| *Код* | *Основний актор* | *Найменування* | *Формулювання* |
| О1 | Черговий лікар | Оформлення історії хвороб | Цей варіант використання дозволяє черговому лікарю оформлювати історію хвороб для лікуючого лікаря. |
| Р1 | Лікуючий лікар | Проводження процесу лікування | Цей варіант використання дозволяє лікуючому лікарю проводити процес лікування |
| C1 | Медсестра | Виконання вказівок лікуючого лікаря | Цей варіант використання дозволяє медсестрі виконувати вказівки лікуючого лікаря |

**Конкретизація варіантів використання**

**О1. Оформлення історії хвороб**

|  |  |  |  |
| --- | --- | --- | --- |
| О1 | Черговий лікар | Оформлення історії хвороб | Цей варіант використання дозволяє черговому лікарю оформлювати історію хвороб для лікуючого лікаря. |

*Основні діючі особи:* Черговий лікар

*Короткий опис*.

Даний варіант використання дозволяє черговому лікарю оформлювати історію хвороб для лікуючого лікаря.

**Р1. Проводження процесу лікування**

|  |  |  |  |
| --- | --- | --- | --- |
| Р1 | Лікуючий лікар | Проводження процесу лікування | Цей варіант використання дозволяє лікуючому лікарю проводити процес лікування |

*Основна діюча особа:* Лікуючий лікар

*Інші учасники прецеденту:* Відсутні.

*Зв'язки з іншими варіантами використання:* Відсутні

*Короткий опис*.

Даний варіант використання дозволяє лікуючому лікарю проводити процес лікування.

**С1. Виконання вказівок лікуючого лікаря**

|  |  |  |  |
| --- | --- | --- | --- |
| C1 | Медсестра | Виконання вказівок лікуючого лікаря | Цей варіант використання дозволяє медсестрі виконувати вказівки лікуючого лікаря |

*Основна діюча особа:* Медсестра

*Інші учасники прецеденту:* Відсутні.

*Зв'язки з іншими варіантами використання:* Відсутні

*Короткий опис*.

Даний варіант використання дозволяє медсестрі виконувати вказівки лікуючого лікаря

Лабораторна робота **№4**

**Детальний опис ключових варіантів використання**

**Мета роботи:** спецефікування ключової функціональності систем

**Теоретичні основи.**

Для специфікації функціональності системи будемо використовувати шаблон варіанту використання RUP.

Нижче приведений короткий огляд його розділів.

* 1. *Найменування і короткий опис.* У цьому розділі вказується: найменування варіанту використання, актори варіанту використання, короткий (у один абзац) опис варіанту використання.
  2. *Потік подій*
     + Основний потік подій

Так само, як в "Основний сценарій" в шаблоні по А.Коберну.

* + - *Альтернативні потоки подій*

Кожен з альтернативних сценаріїв описується в окремому параграфі, в тому ж стилі, що і основний потік подій. Альтернативні сценарії описують поведінку системи при будь-яких відхиленнях від основного сценарію, а також поведінку у виняткових ситуаціях.

* 1. *Спеціальні вимоги*

Тут перераховуються нефункціональні вимоги, що мають безпосереднє відношення саме до цього варіанту використання.

* 1. *Передумови*

Події, що описуються передумовами або постумовами, мають бути станами, які користувач може спостерігати. Передумова описує стан, в якому система повинна знаходитися до початку виконання прецеденту.

* 1. *Постумови*

Постумова RUP по суті описує те ж, що і мінімальна гарантія у Коберна. Коректно сформульована постумова має бути істинною при будь-якому можливому сценарії прецеденту, а не описаному в основному потоці.

* 1. *Точки розширення*

Цей параграф визначає положення точок, що розширюють потік подій.

Опис ключових прецедентів

Пошук ключових варіантів використання

Аналіз сформульованих варіантів використання показав, що з точки зору потенційних ризиків і архітектурної значимості найбільш істотними є прецеденти, пов'язані з роботою адміністрації та вчителя.

Для подальшої деталізації обрані чотири прецедента:

* + О1. Оформлення історії хвороб
  + Р1. Проводження процесу лікування
  + С1. Виконання вказівок лікуючого лікаря

**О1. Оформлення історії хвороб**

Короткий опис

Цей варіант використання дозволяє черговому лікарю оформлювати історію хвороб для лікуючого лікаря.

*Дійові особи цього прецеденту* – Черговий лікар

Потік подій

Прецедент починається, коли Черговий лікар оформлює нового пацієнта

Базовий потік

* Черговий лікар оформлює історію хвороб

Спеціальні вимоги

* Час оформлення не повинен перевищувати 1-2 хвилин.

Передумови.

* Перед тим як починається цей прецедент, черговий лікар, вже зареєстрований в системі.

Постумови

* При успішному закінченні прецеденту програма повертає історію хвороб пациєнта

Точки розширення

* Відсутні.

**Р1. Проводження процесу лікування**

**Короткий опис**

Даний варіант використання дозволяє лікуючому лікарю проводити процес лікування.

*Дійові особи цього прецеденту* – лікуючий лікарь

**Потік подій**

**Базовий потік**

* Лікуючий лікарь проводить лікування.

**Альтернативні потоки**

* Відсутні

**Спеціальні вимоги**

* Виконання операції повинно бути своєчасним та відповідати усім стандартам якості

**Передумови**

* Перед тим як починається цей прецедент, Лікарь повинен отримати історію хвороб.

**Постумови**

* При успішному закінченні прецеденту програма повертає інформацію про процес лікування.

**Точки розширення**

* Відсутні

**С1. Виконання вказівок лікуючого лікаря**

**Короткий опис**

Даний варіант використання дозволяє медсестрі виконувати вказівки лікуючого лікаря

*Дійові особи цього прецеденту* – медсестра

**Потік подій**

**Базовий потік**

* Медсестра виконує вказівки лікуючого лікаря

**Альтернативні потоки**

* Відсутні

**Спеціальні вимоги**

* Виконання операції повинно бути своєчасним та відповідати усім стандартам якості

**Передумови**

* Перед тим як починається цей прецедент, медсестра повинна отримати вказівку від Лікуючого лікаря

**Постумови**

* При успішному закінченні прецеденту програма повертає інформацію про виконану роботу.

**Точки розширення**

* Відсутні

Лабораторна робота **№5**

**Аналіз і специфікація спеціальних вимог**

**Мета роботи:** специфікування нефункціональних вимог до системи, а також функціональних вимог, опис яких у формі прецедентів виконати складно, або недоцільно.

**Теоретичні основи.**

Функціональні вимоги регламентують функціонування або поведінку системи (behavioral requirements). Функціональні вимоги відповідають на питання "що повинна робити система" в тих або інших ситуаціях. Функціональні вимоги визначають основний "фронт робіт" Розробника, і встановлюють цілі, завдання і сервіси, що надаються системою Замовникові.

Нефункціональні вимоги, відповідно, регламентують внутрішні і зовнішні умови або атрибути функціонування системи. К.Вігерс виділяє наступні основні групи нефункціональних вимог:

1. Зовнішні інтерфейси (External Interfaces),
2. Атрибути якості (Quality Attributes)
3. Обмеження (Constraints).

Серед зовнішніх інтерфейсів в більшості сучасних АІС найбільш важливим є інтерфейс користувача (User Interface, UI). Крім того, виділяються інтерфейси із зовнішніми пристроями (апаратні інтерфейси), програмні інтерфейси та інтерфейси передачі інформації (комунікаційні інтерфейси).

Основні атрибути якості :

1. Застосовність,
2. Надійність,
3. Продуктивність,
4. Експлуатаційна придатність.

Обмеження - формулювання умов, що модифікують вимоги або набори вимог, звужуючи вибір можливих рішень по їх реалізації. Вибір платформи реалізації і розгортання (протоколи, сервери додатків, баз даних), які, у свою чергу, можуть відноситися, наприклад, до зовнішніх інтерфейсів.

**Опис ключових прецедентів**

**Мета:** мета цього документа - в тому, щоб визначити додаткові вимоги до розроблюваної АІС. Розглядаються функціональні вимоги, опис яких в формі прецедентів важкий, або недоцільний. Описуються нефункціональні вимоги, що ставляться в цілому до системи.

Посилання

Супутня інформація представлена в наступних документах:

1. Вимогах співвласників;
2. Баченні;
3. Описі акторів і варіантів використання.
4. Описі ключових варіантів використання.

Функціональність

АІС «АвтоЛіки» повинен мати такі функції:

* Перегляд усіх типів товару
* Редагування каталогу товарів (Адміністрація)

**Авторизація та аутентифікація користувачів в системі**

В АІС «**АвтоЛіки**» повинні бути представлені довідник ролей користувачів і довідник користувачів. Повинна бути можливість реєстрації пацієнта.

Застосовність. Зручність використання

Інтерфейс програмного додатку «АвтоЛіки» повинен мати властивості зручності та інтуїтивної ясності і не вимагати додаткової підготовки користувачів.

Надійність. Доступність

АРМ Медичний персонал повинен бути доступний в робочі дні в робочий час.

Час, що витрачається на обслуговування системи не повинен перевищувати 2% від загального часу роботи.

**Напрацювання на відмову**

Середній час безвідмовної роботи - 30 робочих днів.

**Норма дефектів**

Максимальна норма помилок або дефектів - 1 помилка на десять тисяч рядків коду.

Продуктивність

АІС «АвтоЛіки» повинна:

* Успішно проводити процес лікування та реабілітації
* Не допускати збоїв при реєстрації та авторизації користувачів

**Одночасно працюючі користувачі**

Система повинна бути здатна підтримувати мінімум 60 одночасно працюючих користувачів, пов'язаних із загальною базою даних.

Час відгуку

Час відгуку для типових задач - не більше 1-2 секунд, для складних завдань - не більше 5 секунд.

Придатність до експлуатації. Масштабованість

Система повинна бути здатна підтримувати мінімум 60 одночасно працюючих користувачів, пов'язаних із загальною базою даних і мати можливість збільшити їх кількість на випадок збільшення штату робітників.

Оновлення версій

Оновлення версій має здійснюватися в автоматизованому режимі на основі системи контролю версій і системи (сервера) оновлення версій на робочих місцях користувачів.

Обмеження проектування. Застосовувані стандарти

Система повинна відповідати всім стандартам інтерфейсу користувача Операційна система: IOS, Android, Microsoft Windows, Linux, MacOS.

Вимоги до СУБД і доступу до даних.

*У ядрі системи повинна бути представлена СУБД реляційного доступу.*

Лабораторна робота **№6**

**Документування вимог**

**Мета роботи:** необхідно описати зовнішню поведінку системи, а також вимоги і обмеження в документі "Специфікація вимог програмного забезпечення".

**Теоретичні основи.**

Щоб вимоги, виявлені і описані, прийняли силу угоди між Замовником і Розробником, їх необхідно оформити у вигляді документу.

Шаблон "Специфікація вимог програмного забезпечення" (SRS), запропонований в RUP, по суті є контейнером, в який необхідно "упакувати" артефакти, отримані в процесі специфікації вимог. Крім того, SRS частково перекликається з документом "Бачення". Шаблон зручний своєю компактністю і лаконізмом.

Шаблон SRS по RUP:

1. Вступ.
   1. Мета. Документ повинен вичерпним чином описувати зовнішню поведінку системи, а також нефункціональні вимоги і обмеження.
   2. Коротке зведення можливостей.
   3. Визначення, акроніми і скорочення.
   4. Посилання.
   5. Короткий зміст.
2. Огляд системи
   1. Огляд прецедентів. Містить список імен і коротких описів варіантів використання і акторів з ілюстраціями у вигляді діаграм прецедентів.
   2. Припущення і залежності. Ця секція описує ключові технічні можливості, компоненти, підсистеми, пов'язані проекти, які можуть впливати на життєздатність системи, що розробляється.

Припущенням (assumption) називається положення, яке вважається істинним за відсутності доказу або визначальної інформації.

При визначенні залежностей (dependencies) проекту від зовнішніх чинників, необхідно проаналізувати, які нові операційні системи, регламенти бізнес-процесів, стандарти якості, інформаційні системи можуть з'явитися на підприємстві впровадження і як це може вплинути на функціонування автоматизованої інформаційної системи, що виготовляється.

1. Опис вимог
   1. Опис варіантів використання. Параграф містить опис варіантів використання і пов'язаних з ними нефункціональних вимог, або посилання на відповідні артефакти.
   2. Спеціальні вимоги. Параграф містить опис функціональних вимог (не описаних в якості варіантів використання), а також опис нефункціональних вимог загального характеру (не зіставлених жодному прецеденту в попередньому розділі), або посилання на відповідні артефакти.
2. Допоміжна інформація.

Сюди включається інформація, що полегшує розуміння документу. Це може бути зміст і додатки, наприклад, прототипи призначеного для користувача інтерфейсу, що описують.

**Специфікація умов до АІС**

***Мета:*** Мета цього документа - в тому, щоб сформулювати вимоги до розроблюваної АІС обліку товарів на складі. Дані вимоги описані в формі прецедентів, коротких описів функціональних вимог і описів не функціональних вимог.

Посилання

Супутня інформація представлена в наступних документах:

1. баченні (Лабораторна робота №1) і
2. глосарії (Лабороторна робота №4).

Короткий зміст. Огляд системи. Огляд прецедентів

Короткий опис акторів представлено в табл. 1.

Табл. 1. Актори системи

|  |  |
| --- | --- |
| *Актор* | *Короткий опис* |
| Черговий лікар | Оформлює пацієнтів, оформлює історію хвороб, повідомляє рідних. |
| Лікуючий лікар | Проводить лікування, керує медсестрами, може проводити перш-черговий огляд. |
| Медсестра | Виконує вказівки Лікуючого Лікаря, може проводити перш-черговий огляд та дивитися історію хвороб. |

Список варіантів використання у таблиці 2.

Табл. 2. Реестр варіантів використання

|  |  |  |  |
| --- | --- | --- | --- |
| *Код* | *Основний актор* | *Найменування* | *Формулювання* |
| О1 | Черговий лікар | Оформлення історії хвороб | Цей варіант використання дозволяє черговому лікарю оформлювати історію хвороб для лікуючого лікаря. |
| Р1 | Лікуючий лікар | Проводження процесу лікування | Цей варіант використання дозволяє лікуючому лікарю проводити процес лікування |
| C1 | Медсестра | Виконання вказівок лікуючого лікаря | Цей варіант використання дозволяє медсестрі виконувати вказівки лікуючого лікаря |

Припущення і залежності

Немає

Опис вимог

**Короткі описання варіантів використання**

**О1. Оформлення історії хвороб**

|  |  |  |  |
| --- | --- | --- | --- |
| О1 | Черговий лікар | Оформлення історії хвороб | Цей варіант використання дозволяє черговому лікарю оформлювати історію хвороб для лікуючого лікаря. |

*Основні діючі особи:* Черговий лікар

*Короткий опис*.

Даний варіант використання дозволяє черговому лікарю оформлювати історію хвороб для лікуючого лікаря.

**Р1. Проводження процесу лікування**

|  |  |  |  |
| --- | --- | --- | --- |
| Р1 | Лікуючий лікар | Проводження процесу лікування | Цей варіант використання дозволяє лікуючому лікарю проводити процес лікування |

*Основна діюча особа:* Лікуючий лікар

*Інші учасники прецеденту:* Відсутні.

*Зв'язки з іншими варіантами використання:* Відсутні

*Короткий опис*.

Даний варіант використання дозволяє лікуючому лікарю проводити процес лікування.

**С1. Виконання вказівок лікуючого лікаря**

|  |  |  |  |
| --- | --- | --- | --- |
| C1 | Медсестра | Виконання вказівок лікуючого лікаря | Цей варіант використання дозволяє медсестрі виконувати вказівки лікуючого лікаря |

*Основна діюча особа:* Медсестра

*Інші учасники прецеденту:* Відсутні.

*Зв'язки з іншими варіантами використання:* Відсутні

*Короткий опис*.

Даний варіант використання дозволяє медсестрі виконувати вказівки лікуючого лікаря

Повні описи варіантів використання

Аналіз сформульованих варіантів використання показав, що з точки зору потенційних ризиків і архітектурної значимості найбільш істотними є прецеденти, пов'язані з роботою адміністрації та вчителя.

Для подальшої деталізації обрані такі прецинденти:

* + О1. Оформлення історії хвороб
  + Р1. Проводження процесу лікування
  + С1. Виконання вказівок лікуючого лікаря

Функціональність

АІС «АвтоЛіки» повинен мати такі функції:

* Перегляд усіх типів товару
* Редагування каталогу товарів (Адміністрація)

**Авторизація та аутентифікація користувачів в системі**

В АІС «**АвтоЛіки**» повинні бути представлені довідник ролей користувачів і довідник користувачів. Повинна бути можливість реєстрації пацієнта.

Застосовність. Зручність використання

Інтерфейс програмного додатку «АвтоЛіки» повинен мати властивості зручності та інтуїтивної ясності і не вимагати додаткової підготовки користувачів.

Надійність. Доступність

АРМ Медичний персонал повинен бути доступний в робочі дні в робочий час.

Час, що витрачається на обслуговування системи не повинен перевищувати 2% від загального часу роботи.

**Напрацювання на відмову**

Середній час безвідмовної роботи - 30 робочих днів.

**Норма дефектів**

Максимальна норма помилок або дефектів - 1 помилка на десять тисяч рядків коду.

Продуктивність

АІС «АвтоЛіки» повинна:

* Успішно проводити процес лікування та реабілітації
* Не допускати збоїв при реєстрації та авторизації користувачів

**Одночасно працюючі користувачі**

Система повинна бути здатна підтримувати мінімум 60 одночасно працюючих користувачів, пов'язаних із загальною базою даних.

Час відгуку

Час відгуку для типових задач - не більше 1-2 секунд, для складних завдань - не більше 5 секунд.

Придатність до експлуатації. Масштабованість

Система повинна бути здатна підтримувати мінімум 60 одночасно працюючих користувачів, пов'язаних із загальною базою даних і мати можливість збільшити їх кількість на випадок збільшення штату робітників.

Оновлення версій

Оновлення версій має здійснюватися в автоматизованому режимі на основі системи контролю версій і системи (сервера) оновлення версій на робочих місцях користувачів.

Обмеження проектування. Застосовувані стандарти

Система повинна відповідати всім стандартам інтерфейсу користувача Операційна система: IOS, Android, Microsoft Windows, Linux, MacOS.

Вимоги до СУБД і доступу до даних.

*У ядрі системи повинна бути представлена СУБД реляційного доступу.*