**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НТУУ «КПІ Імені Ігоря Сікорського»**

**ІПСА - САПР**

**Лабораторна робота №2**

з курсу **«Проектування інформаційних систем»**

на тему:

“ Скласти опис передпроектної документації (Software Requirement Specifications).”

Виконав:

студент IV курсу

групи ДА-62

Сиротюк О.В.

**Київ 2019**

**Мета роботи:** вивчити основні етапи створення передпроектної документації (Software Requirement Specifications).

**Задача**: використати приклад **Software Requirement Specifications** для створення передпроектної документації згідно індивідуальної темі для виконання лабораторних робіт.

**Завдання:**

1. Вивчити вимоги до передпроектної документації.

2. Скласти опис передпроектної документації для об'єкта проектування.

3. Скласти 5-7 приймальних тестів для ПО об'єкта проектування.

4. Оформити технічне завдання згідно опис передпроектної документації

(використовувати рекомендації IEEE 830).

**Зміст звіту:**

1. Мета роботи.

2. Завдання роботи.

3. Оформлення результатів роботи.

**Вступ**

**Ціль**

Дана специфікація вимог до програмного забезпечення описує функціональні та нефункціональні вимоги до інформаційної системи.

**Масштаб проекту**

**mStorage** – це мобільний додаток, який призначений допомогти користувачу зберігати текстові матеріали будь-якого розміру, видаляти, редагувати та залишати нотатки.

**Цільова аудиторія**

**mStorage** не має вікового обмеження. Проте цільовою аудиторією є люди віком від 16-ти років.

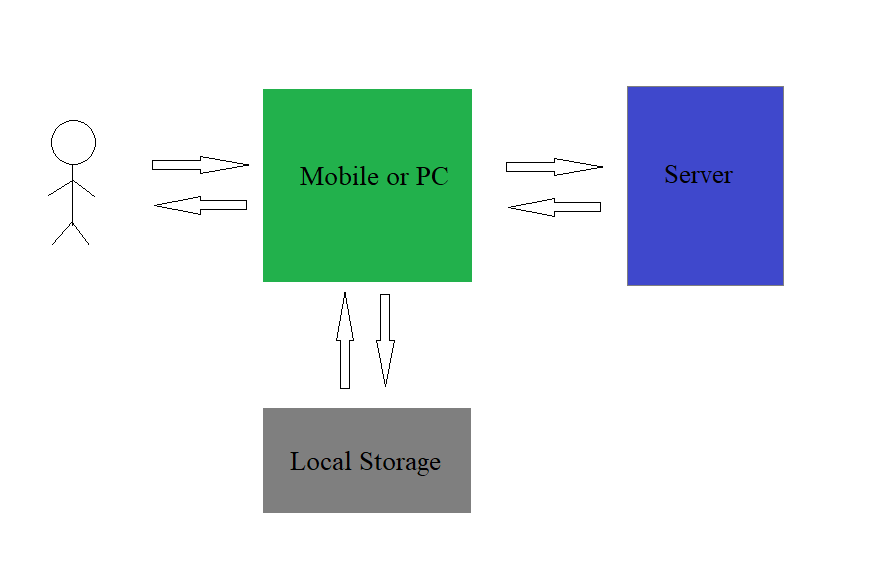
**Ревю**

Специфікація написана на базі стандарту IEEE – 830 – 1998. Опис системи в загальному випадку, її головних функцій та інтерфейсів знаходиться в розділі повного опису. Повний опис функціональних та нефункціональних вимог там обмежень знаходиться у розділі специфічних вимог.

**Загальний опис**

**mStorage –** мобільний додаток. Основний функціонал додатку доступний з телефону або ПК. Сервер грає роль системи автентифікації та віддаленого сховища клієнтських записів. Як показано на малюнку 1, **mStorage** взаємодіє з віддаленим сервером. Користувач отримує доступ до системи через додаток.

Для отримання інформації з локальної бази доступ до інтернету не потрібен, проте, з’єднання потрібне для отримання доступу до записів на віддаленому сервері. Ролі відсутні, планується підтримка декількох мов. Об’єм пам’яті, потрібний для локальної бази – необмежений.



**Малюнок 1**

**Системні інтерфейси**

Кроссплатформний додаток має змогу доступатися до даних, що зберігаються на віддаленому сервері за допомогою вбудованих інтерфейсів.

**Інтерфейси користувача**

Інтерфейс користувачаповинен бути інтуїтивно зрозумілим і не бути перенасиченим інформацією, як в текстовому вигляді, так і в кольоровій гаммі. Користувач повинен бути максимально відгороджений від системних ресурсів та функцій за допомогою абстракцій у вигляді графічного інтерфейсу. Гамма повинна бути підібрана з урахуванням UX різних типів людей. (Наприклад – з кольоровою сліпотою). Бажано – Material дизайн.

**Програмні інтерфейси**

Розрахований на використання на мобільних та десктопних платформах з підтримкою виконання коду на мові С++ та можливістю до передачі даних через вбудовані сокети за допомогою протоку **TCP** та **UDP**.

**Характеристика користувачів**

Користувач – людина, що користується фунукціоналом ПО у власних цілях. Користувачі ПО не поділені на класи в рамках системи та мають однакові права на читання та запис матеріалів. Система не вимагає спеціальних знань та навичок.

**Обмеження та стандарти**

Початкова версія продукту обмежена мінімальним функціоналом та підтримкою однієї мови інтерфейсу та файлів – англійська. Сервер та клієнт реалізуються на мові С++ з використанням бази даних CouchDB, протоколів передачі даних TCP та самописним інтерфейсом мережевої взаємодії.

**Припущення та залежності**

Пристрій підключений до інтернету. Сервер постійно доступний.

**Функціональність системи**

**Сценарій: авторизація в системі**

**Опис**: користувач хоче зайти в систем за допомогою логіна та паролю.

Якщо користувач не зареєстрований, переходимо до наступного пункту, при неправильному введені логіну чи паролю – користувач отримає помилку та попередження. В іншому випадку – авторизація пройшла успішно.

**Сценарій: реєстрація в системі**

**Опис**: користувач хоче зареєструватися в системі, для цього йому потрібно створити логін та пароль. Якщо логін зайнятий, система виведе попередження. Якщо пароль чи логін не задовольняє умови – система видає попередження.

**Сценарій: зміна паролю**

**Опис**: користувач хоче змінити пароль. Для цього йому потрібно ввести старий пароль та новий пароль. Якщо старий пароль введено неправильно – система видає попередження, якщо новий пароль не задовольняє умови – система видає попередження. Якщо новий пароль співпадає зі старим – система видає попередження.

**Сценарій: додання запису**

**Опис:** користувач хоче додати новий запис. Для цього йому потрібно ввести назву запису. Назва запису повинна містити від 1 до 256 символів та не повторювати назву іншого запису. Якщо буде введено коротший запис – він не буде створений, довший – він буде обрізаний до максимального розміру. Якщо. У разі створення запису - запис стає доступним до редагування, якщо створення запису не було відмінено користувачем.

**Сценарій: редагування запису**

**Опис**: користувач хоче редагувати запис. Користувач повинен відкрити вже створений запис, якщо файли на локальному сховищу не пошкоджені, то можливе редагування файлу. Якщо місця в пам’яті замало – неможливо зберегти зміни, внесені в файл.

**Сценарій: видалення запису**

**Опис:** користувач хочу видалити запис. Для цього йому потрібно вибрати запис, а потім підтвердити видалення запису. Система запросить ввести назву запису, для підтвердження його видалення. Якщо введений текс не співпадає з назвою запису – система робить повторний запит користувачу, допоки користувач не введе назву корректно, або не відмінить видалення.

**Сценарій: пошук запису**

**Опис**: користувач хоче знайти запис за іменем. Користувач вводить назву запису, назва повинна містити від 1 до 256 символів. Після підтвердження пошуку, виводиться список релевантних записів, відсортованих за релевантнітью. У разі відсутності записів – система виведе попередження.

**Сценарій: перегляд статистики**

**Опис:** користувач хоче переглянути статистику запису. Для цього користувач вибирає запис та вводить команду, чи вибирає в ГУІ команду для перегляду статистики. В статистику входить розмір файлу, дата створення, дата останнього редагування, назва файлу.

**Сценарій: перегляд запису**

**Опис:** користувач хоче переглянути запису. Для цього користувач вибирає запис та вводить команду, чи вибирає в ГУІ команду для перегляд. На екран друкуються всі символи, що є можливими для друку.

**Сценарій: видалення профілю**

**Опис:** користувач хоче видалити профіль. Для цього користувач вводить потрібну команду, чи вибирає пункт ГУІ. Користувач повинен ввести логін та пароль, у разі помилки – система виведе попередження. Якщо операція успішна, всі записи на віддаленому сервері та локальному сховищі будуть стерті, а логін буде звільнений.

**Сценарій: синхронізація з сервером**

**Опис:** користувач синхронізує записи з сервером. Передає власні зміни на віддалений сервер, або підтягує на пристрій наявний стан віддаленого серверу.

Якщо інтернет зв’язок відсутній – синхронізація неможлива.

**Нефункціональні вимоги**

**Вимоги до продуктивності:**

Програма повинна запускатись на мобільних платформах з обмеженою кількістю оперативної пам’яті. Підтримувати працездатність ПО навіть під час відключення від інтернет-мережі, а також надавати відносно швидкий відклик на дії користувача. Швидкість запису на віддалений сервер не є критичною характеристикою, проте нехтувати нею не потрібно. Головною характеристикою є об’єм інформації, що може бути збережена, та максимальна ефективність використання пам’яті на пристрої.

**Вимоги до бази даних:**

* Таблиця авторизації – глобальна серверна таблиця для зберігання пари логіну та паролю у захешованову вигляді.
* Таблиця заголовків – перелік заголовків записів (унікальна для кожного користувача). Зовнішній ключ для з’єднання з текстом та користувачем.
* Таблиця записів – таблиця з текстом запису. З’єднується з користувачем та записом через зовнішній ключ – ім’я запису.

**Вимоги до якості ПО:**

* Так як ПО є клієнт-серверним, основною вимогою до нього є безпека персональної інформації: логін, пароль, записи.
* Інтерфейс повинен бути мінімальним та простим, інтуїтивно зрозумілим. Бажане використання стилю Material, або Flat. Використання нейтральних кольорів, котрі не можуть відволікати користувача від текстів, або роботи з ПО.
* Програма повинна толерантно відноситись до пам’яті пристрою та швидкості з’єднання, аби мати змогу працювати в фоновому режимі (синхронізація з віддаленим сервером)

**Вимоги до безпеки системи**

Системі потрібно забезпечити декілька рівнів шифрування. **RES** – для передачі ключів шифрування, **AES** – для кодування логінів, паролів та записів. А також – хешування самих логінів та паролі (можливо – записів), аби не представляти даних у чистому вигляді навіть у пам’яті. Важливою частиною є санітазування змінних середовища, аби заблокувати вплив операційної системи на програму, а також – створити власний менеджер пам’яті для ПО, щоб уникнути ін’єкцій в пам’ять. Окремо потрібно захистити БД від SQL ін’єкцій.

**Приймальні тести**

**Тест 1: «Створення запису»**

Ціль:

* Перевірити коректність роботи функції створення записів.
* Перевірити коректність розпізнавання програмою базової таблиці кодування **ASCI**.
* Перевірити коректність збереженої інформації.

Результат:

* Запис створено.
* Запис доступний для перегляду.
* Данні в запису відповідають даним, що були внесені користувачем.
* Запис не займає зайвого вільного місця.

**Тест 2: «Редагування запису»**

Ціль:

* Перевірити коректність роботи функції редагування.

Результат:

* Запис відредаговано.
* Запис доступний для перегляду.
* Данні в запису відповідають даним, що були внесені користувачем.
* Запис не займає зайвого вільного місця.

**Тест 3: «Видалення запису»**

Ціль:

* Перевірити коректність роботи функції видалення.
* Перевірити коректність системи підтвердження видалення.

Результат:

* Запис видалено.
* Можливо створити новий запис з таким же іменем.
* Місце на диску звільнено.

**Тест 3: «Перегляд запису»**

Ціль:

* Перевірити коректність роботи функції перегляду.

Результат:

* На екрані надруковано текст запису.

**Тест 3: «Перегляд статистики запису»**

Ціль:

* Перевірити коректність роботи функції статистики.

Результат:

* На екрані надруковано статистику запису.
* Статистика коректна.

**Тест 4: «Передача даних на сервер»**

Ціль:

* Перевірити коректність роботи функції синхронізації.

Результат:

* Локальні зміни передані та збережені на віддаленому сервері та готові для опрацювання.
* Записи закодовані.
* Канал передачі безпечний.
* Записи займають місця не більше ніж в локальному сховищі.

**Тест 5: «Забір даних з серверу»**

Ціль:

* Перевірити коректність роботи функції синхронізації.

Результат:

* Зміни з серверу записані в локальне сховище.
* Записи не займають зайвого місця.
* Канал передачі безпечний.

**Тест 6: «Пошук запису»**

Ціль:

* Перевірити коректність роботи функції пошуку.

Результат:

* Результат пошуку коректний. Файл знайдено, або ні. В залежності від очікуваного результату пошуку.