

Лабораторна робота №1
Аналіз і формалізація вимог до програмного продукту.
Розробка технічного завдання на створення програмного продукту

Мета роботи: навчитися аналізувати й формалізувати вимоги замовника з використанням UML, розраховувати витрати на створення програмного продукту, виконувати планування робіт, розробляти та оформлювати технічне завдання на створення програмного продукту відповідно до ДСТУ.

Хід виконання роботи:

1. Вивчити теоретичні відомості.
2. Виконати аналіз і формалізацію вимог замовника на розробку програмного продукту відповідно до індивідуального завдання.
3. Розробити діаграму прецедентів використання й виконати опис прецедентів.
4. Виконати розрахунок витрат на створення програмного продукту.
5. Виконати планування робіт зі створення програмного продукту.
6. Розробити технічне завдання на створення програмного продукту.
7. Зробити висновки про вибір моделі створення програмного продукту.

Вимоги до змісту роботи:

1. Назва роботи.
2. Мета роботи.
3. Формулювання індивідуального завдання.
4. Діаграма прецедентів використання з їхнім описом.
5. Розрахунок витрат на створення програмного продукту.
6. Технічне завдання на створення програмного продукту.
7. Висновки про вибір моделі створення програмного продукту.

Індивідуальне завдання

Розробка системи з базою даних. Створення системи для запису студентів на лабораторні стенди з метою виконання лабораторних робіт за розкладом по визначених дисциплінах.

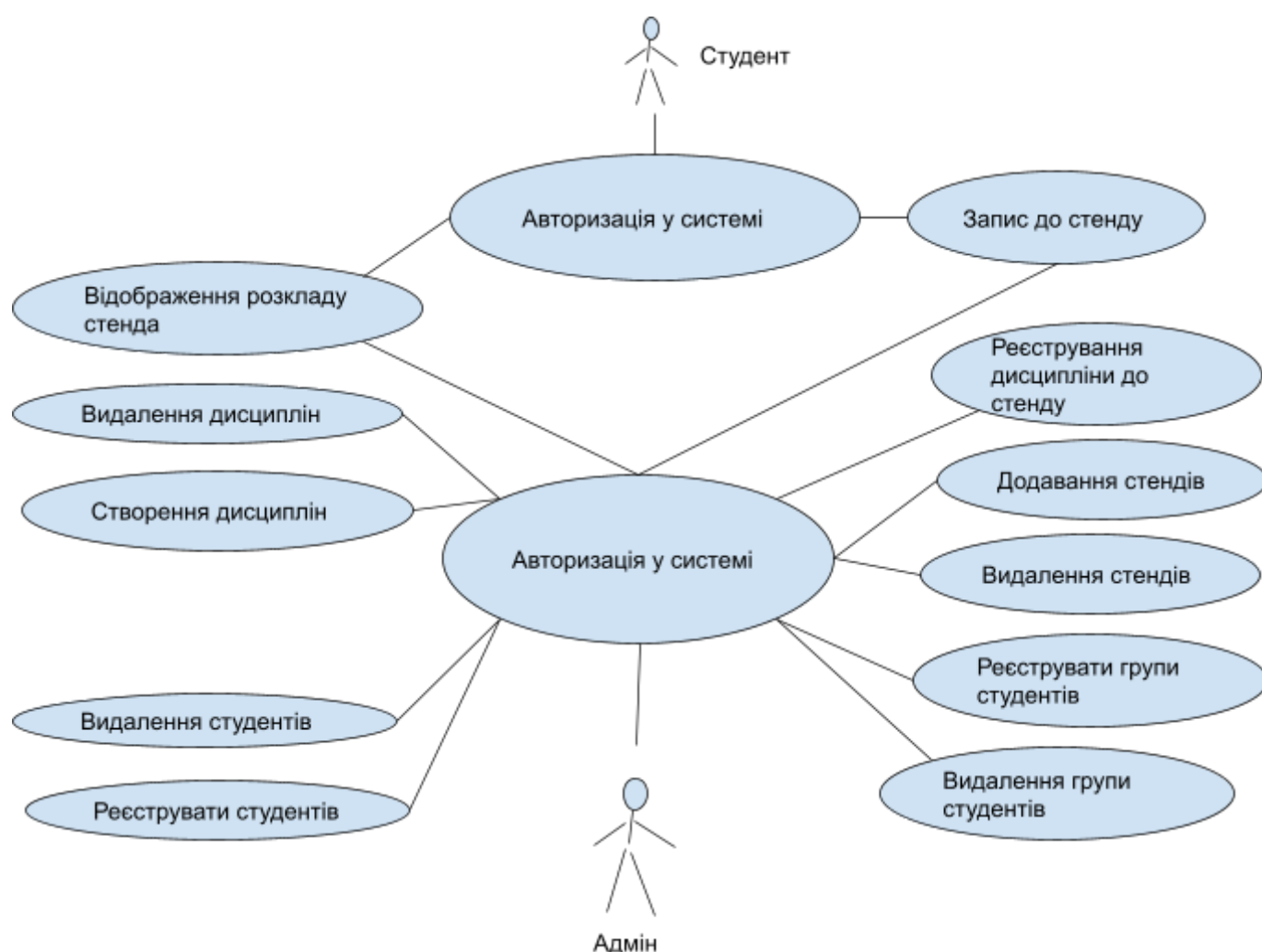
Технічне завдання

Програмна частина:

1. MySQL
2. PHP
3. HTML, CSS

Зміст та функціонал додатку:

Розділ	Опис
1.Розділ з авторизацією Адміна	Адмін повинен зареєструвати кожного студента до бази даних. Адмін також додає студента до його групи. Реєструє стенд з дисципліною, записує студента до нього
2. Розділ з авторизацією студента	Студент може: записатися до стенда, та може переглянути розклад занять та стенду.
3. Сторінка профілю студента	На сторінці профілю студента можна буде подивитися більше інформації про студента та його дисципліни.



Опис діаграми прецедентів “Реєстрування студента”

Основний виконавець – програмний продукт
Зацікавлені особи – користувач з правами адміністратора
Передумови: Користувач має зареєстрований аккаунт.
Вхідні дані: Додавання профілю нового студента.

Основний успішний сценарій: <ul style="list-style-type: none"> - Адміністратор авторизується у системі; - Адміністратор переходить до списку студентів; - Адміністратор вводить у форму дані нового студента; - Натискає кнопку “Додати”.
Частота виконання: Для додавання нового студента.
Постумови (результати): Адміністратор успішно додав нового студента
Вихідні дані: Доданий новий студент

Опис діаграми прецедентів “Видалення студента”

Основний виконавець – програмний продукт Зацікавлені особи – користувач з правами адміністратора
Передумови: Користувач має зареєстрований аккаунт.
Вхідні дані: Видалення профілю студента.
Основний успішний сценарій: <ul style="list-style-type: none"> - Адміністратор авторизується у системі; - Адміністратор переходить до списку студентів; - Адміністратор видаляє профіль студента;
Частота виконання: Для видалення профілю студента.
Постумови (результати): Адміністратор успішно видалив профіль студента
Вихідні дані: Видалений профіль студента

Опис діаграми прецедентів “Створення групи”

Основний виконавець – програмний продукт Зацікавлені особи – користувач з правами адміністратора
Передумови: Користувач має зареєстрований аккаунт.
Вхідні дані: Додавання нової групи.
Основний успішний сценарій: <ul style="list-style-type: none"> - Адміністратор авторизується у системі; - Адміністратор переходить до списку груп; - Адміністратор вводить у форму назву групи; - Натискає кнопку “Додати”.
Частота виконання: Для додавання нової групи.
Постумови (результати): Адміністратор успішно додав нову дисципліну

Вихідні дані: Додана нова група
--

Опис діаграми прецедентів “Видалення групи”

Основний виконавець – програмний продукт Зацікавлені особи – користувач з правами адміністратора

Передумови: Користувач має зареєстрований аккаунт.

Вхідні дані: Видалення групи

Основний успішний сценарій:

- | |
|---|
| <ul style="list-style-type: none">- Адміністратор авторизується у системі;- Адміністратор переходить до списку груп;- Адміністратор видаляє обрану групу; |
|---|

Частота виконання: Для видалення групи.
--

Постумови (результати): Адміністратор успішно видалив групи
--

Вихідні дані: Видалена група

Опис діаграми прецедентів “Створення дисципліни”

Основний виконавець – програмний продукт Зацікавлені особи – користувач з правами адміністратора

Передумови: Користувач має зареєстрований аккаунт.

Вхідні дані: Додавання нової дисципліни.

Основний успішний сценарій:

- | |
|--|
| <ul style="list-style-type: none">- Адміністратор авторизується у системі;- Адміністратор переходить до списку дисциплін;- Адміністратор вводить у форму назву дисципліни;- Натискає кнопку “Додати”. |
|--|

Частота виконання: Для додавання нової дисципліни.

Постумови (результати): Адміністратор успішно додав нову дисципліну
--

Вихідні дані: Додана нова дисципліна

Опис діаграми прецедентів “Видалення дисципліни”

Основний виконавець – програмний продукт Зацікавлені особи – користувач з правами адміністратора

Передумови: Користувач має зареєстрований аккаунт.
Вхідні дані: Видалення дисципліни
Основний успішний сценарій: <ul style="list-style-type: none"> - Адміністратор авторизується у системі; - Адміністратор переходить до списку дисциплін; - Адміністратор видаляє обрану дисципліну;
Частота виконання: Для видалення дисципліни.
Постумови (результати): Адміністратор успішно видалив дисципліну
Вихідні дані: Видалена дисципліна

Опис діаграми прецедентів “Реєстрування стенда”

Основний виконавець – програмний продукт Зацікавлені особи – користувач з правами адміністратора
Передумови: Користувач має зареєстрований аккаунт.
Вхідні дані: Додавання профілю нового стенда.
Основний успішний сценарій: <ul style="list-style-type: none"> - Адміністратор авторизується у системі; - Адміністратор переходить до списку стендів; - Адміністратор вводить у форму дані нового стенда; - Натискає кнопку “Додати”.
Частота виконання: Для додавання нового стенда.
Постумови (результати): Адміністратор успішно додав новий стенд
Вихідні дані: Доданий новий стенд

Опис діаграми прецедентів “Видалення стенда”

Основний виконавець – програмний продукт Зацікавлені особи – користувач з правами адміністратора
Передумови: Користувач має зареєстрований аккаунт.
Вхідні дані: Видалення стенда.
Основний успішний сценарій: <ul style="list-style-type: none"> - Адміністратор авторизується у системі; - Адміністратор переходить до списку стендів; - Адміністратор видаляє стенд;
Частота виконання: Для видалення стенда.

Постумови (результати): Адміністратор успішно видалив стенд
Вихідні дані: Видалений стенд

Опис діаграми прецедентів “Відображення розкладу стенда”

Основний виконавець – програмний продукт Зацікавлені особи – користувач з правами адміністратора
Передумови: Користувач має зареєстрований аккаунт.
Вхідні дані: Відображення розкладу стенда
Основний успішний сценарій: <ul style="list-style-type: none"> - Адміністратор авторизується у системі; - Адміністратор переходить до списку розкладу стенда;
Частота виконання: Для відображення розкладу стенда.
Постумови (результати): Адміністратор успішно відобразив розклад стенда
Вихідні дані: Список розкладу стенда

Опис діаграми прецедентів “Запис до стенду”

Основний виконавець – програмний продукт Зацікавлені особи – користувач з правами адміністратора
Передумови: Користувач має зареєстрований аккаунт.
Вхідні дані: Запис до стенду
Основний успішний сценарій: <ul style="list-style-type: none"> - Адміністратор авторизується у системі; - Адміністратор переходить до розкладу стенда; - Адміністратор резервує вільний час у розкладі;
Частота виконання: Для запису до стенда.
Постумови (результати): Адміністратор успішно записав студента до стенду.
Вихідні дані: Студент записаний до стенду.

Опис діаграми прецедентів “Відображення розкладу стенда”

Основний виконавець – програмний продукт Зацікавлені особи – користувач з правами студента
Передумови: Користувач має зареєстрований аккаунт.
Вхідні дані: Відображення розкладу стенда

Основний успішний сценарій: <ul style="list-style-type: none"> - Студент авторизується у системі; - Студент переходить до списку розкладу стенда;
Частота виконання: Для відображення розкладу стенда.
Постумови (результати): Студент успішно відобразив розклад стенда
Вихідні дані: Список розкладу стенда

Опис діаграми прецедентів “Запис до стенду”

Основний виконавець – програмний продукт Зацікавлені особи – користувач з правами студента
Передумови: Користувач має зареєстрований аккаунт.
Вхідні дані: Запис до стенду
Основний успішний сценарій: <ul style="list-style-type: none"> - Студент авторизується у системі; - Студент переходить до розкладу стенда; - Студент резервує вільний час у розкладі;
Частота виконання: Для запису до стенда.
Постумови (результати): Студент успішно записався до стенду.
Вихідні дані: Студент записаний до стенду.

Вимоги до користувача:

Для взаємодії з програмним застосунком обладнання користування має відповідати наступним мінімальним вимогам:

- монітор з мінімальною роздільною здатністю 1024x768 (для перегляду веб-сторінки);
- клавіатура та мишка (для взаємодії з застосунком).

Платформи, на яких можна буде запустити програмний застосунок:

- Windows 7/8/10 x32, x64, Android
- Linux з графічним інтерфейсом та інтернет браузером
- MacOS

Для відображення веб-сторінок, потребується порядку 200 МБ оперативної пам'яті (в залежності від браузера).

Серверні вимоги: веб-сервер Apache + стандартні розширення включно з PHP, СУБД MySQL. Оперативна пам'ять для виконання серверних скриптів: 128 МБ. Розмір бази даних на початку становить ~ 16 МБ вільного місця (стартовий розмір файлу InnoDB).

Загальний мінімальний обсяг пам'яті на сервері 256 МБ.

Витрати на розробку:

На розробку програмного забезпечення потребується порядку 120 годин.

Вартість часу розробки 200 грн/час. Термін розробки складає $120 / 4 = 30$ днів.

Вартість програмного застосунку для розробки – 0 грн (MySQL, PHP, HTML, CSS).

Загальна вартість: $K = (120 * 200) = 24\ 000$ грн.

Висновок: на цій лабораторній роботі я навчився аналізувати й формалізувати вимоги замовника з використанням UML, розраховувати витрати на створення програмного продукту, виконувати планування робіт, розробляти та оформлювати технічне завдання на створення програмного продукту відповідно до ДСТУ.