Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Проект

на тему : «система організації навчального процесу в університеті»

*Викова:*

Студент групи IO-з91

Загородоній Богдан

Залікова книжка №9103

*Перевірив: Болдак* А. А.

г. Киев

2021 р.

Вступ

1. У даному документі описуються запити зацікавлених осіб котрими виступає замовник – Будь-який Національний Технічний університет України , для якого розроблюється система організації навчального процесу в університеті.

1.1 Мета

Метою документу є визначити головні вимоги до функціональності, продуктивності, надійності, зручності, доступності, а також визначити бізнес-правила та технологічні обмеження, що накладаються на предмет розробки.

1.2 Контекст

Перелік вимог, зазначених у даному документі, є основою технічного завдання для розробки системи організації навчального процесу в НТУ

Короткий огляд продукту

2. Система організації навчального процесу в НТУУ *-* це база даних всіх студентів та працівників університету разом із програмними продуктами, надають 3 різні рівні доступу. Студенти (перший рівень) можуть надіслати запит на огляд проекту,надіслати запит для приєднання до проекту і редагувати його . Також можна залишити повідомлення чи отримати консультацію,. Викладачі(другий рівень) зможуть швидко створювати проекти, додавати та видаляти студентів до них, робити оголошення. Адміністрація університету(третій рівень) зможе ефективно організувати та контролювати навчальний процес, вносити зміни, швидко отримувати інформацію та вести статистику.

1. Ділові правила
   1. Призначення системи

Система призначена для збору інформації про студентів та викладачів із подальшим її використанням на різних рівнях доступу для полегшення організації навчального процесу та економії часу та зусиль студентів та викладачів університету.

3.2 Політика взаємовідносин із клієнтом

Клієнтами системи можуть бути студенти, аспіранти, спеціалісти, професори, працівники адміністрації, що навчаються та/або працюють у даному університеті.

Політика взаємовідносин із клієнтом системи полягає в наданні йому різного роду інформації, в основному із допомогою веб-сервісів, із можливістю чи неможливістю внесення змін до неї.

3.3 Характеристика ділового процесу

Керівництво системою здійснюється адміністраторами, що взаємодіють із керівництвом університету.

Адміністратори системи несуть відповідальність за створення, збереження та видалення інформації будь-якого роду.

Керівництво університету має доступ на рівні адміністраторів й також несе відповідальність за створення, збереження, видалення чи внесення змін до інформації будь-якого роду.

Викладачі мають доступ до інформації про проекти та групи, у яких викладають, й також несуть відповідальність за створення, збереження, видалення чи внесення змін до даної інформації.

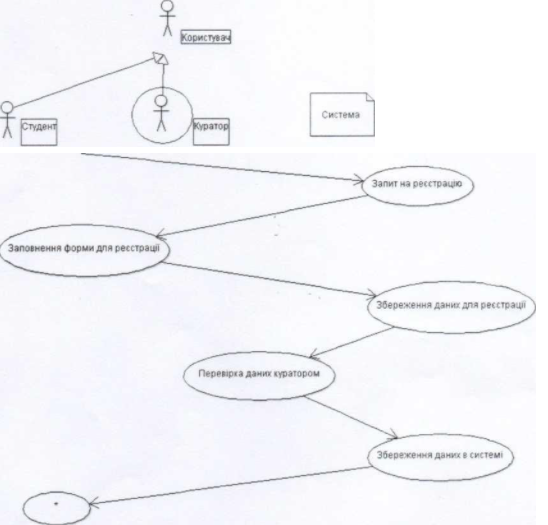
Студенти системи мають доступ до ознайомлювальної частини системи (проекти,інші користувачі та викладачі) й несуть відповідальність за редаугвання проектів та використання особистих даних інших користувачів системи.

3.4 Сценарій реєстрації нового користувача

Студент реєструється самостійно, заповнюючи реєстраційну форму на веб-сайті, вводить усі необхідні дані (логін, пароль, підтвердження паролю, ім'я, прізвище, по-батькові, e-mail, дійсний контактний телефон, контакти батьків, адресу проживання, ін.). Зі створеного акаунту студент відправляє запит куратору своєї групи, куратор перевіряє дані та включає студента до списків відповідної групи.

Викладач або працівник адміністрації реєструється також самостійно, згодом відсилаючи запит адміністраторам системи, які повинні перевірити введену інформацію та надати відповідні права доступу.

3.5 Реєстрація нового користувача



4. Функціональність

Головні вимоги щодо функціональності, пред'явлені зацікавленими особами до предмету розробки, відносяться до чотирьох категорій :

* Студент
* Викладач, викладач-куратор
* Адміністратор системи
* гість

4.1 Послуги для студента

Для студентів система надаватиме такі послуги :

* Залишити запит на додання до проекту
* Знайти викладача
* Залишити повідомлення для викладача
* Залишити повідомлення для всієї групи
* Знайти проект
* Переглянути останні новини, повідомлення
* Редагувати акаунт
* Редагування проекту

4.2 Можливості для викладача, куратора

Для викладачів система надаватиме наступні можливості :

* Можливість створення, редагування та видалення проектів
* Залишити повідомлення групі/окремому студенту
* Отримати інформацію про студента
* Переглянути/залишати новини, повідомлення
* Редагувати акаунт
* Перевірити правильність реєстраційних даних студента

4.3 Можливості для адміністраторів системи

Для адміністраторів система повинна надавати наступні можливості

* Створити/редагувати/видалити акаунт за згодою керівництва університету
* Розширити/обмежити права користувачів
* Додати/видалити проект
* Залишити новини
* Обмежувати можливості користувача за порушення правил (нецензурну лексику, неправдиві повідомлення, неправильні дані)

1. 4.4 Можливості гостя

* Переглядати оголошення
* Переглядати контактну інформацію
* Реєструватися

5.Мобільність

Веб-сайт повинен бути оптимізованим для роботи не тільки із комп'ютера, а також із мобільних пристроїв.

6. Надійність

6.1 Резервне копіювання та відновлення даних

Повинно проводитись резервне копіювання баз даних

6.2 Захист від зловмисних атак

Система повинна бути добре захищена від різного роду зловмисних атак із метою заволодіння інформації чи атак типу DDoS.

6.3 Великі навантаження

Система повинна витримувати великі навантаження, обслуговуючи значну кількість користувачів.



7. Сценарії

7.1 **Назва:** сценарій реєстрації на сайті.

**Учасники:** гість, система.

**Передумови:** гість не зареєстрований у системі.

**Результат:** зареєстровано нового студента.

**Сценарій:**

1.Гість реєструеться на сайті

2.Система надсилає запит підтвердження на почту Гостю

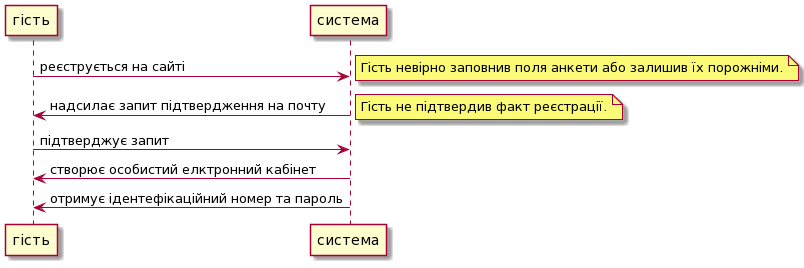
3.Гість підтверджує запит

4.Система створює особистий елктронний кабінет для гостя

5. Гість отримує ідентефікаційний номер та пароль

**Виключні ситуації:**

1. Гість невірно заповнив поля анкети або залишив їх порожніми.
2. Гість не підтвердив факт реєстрації.
3. Порушено зв’язок між сервером із базою даних та сайтом.

****

* 1. **Назва:** сценарій додавання до проекту.

**Учасники:** Студент, Викладач.

**Передумови:** Студент зареєстрований на сайті.

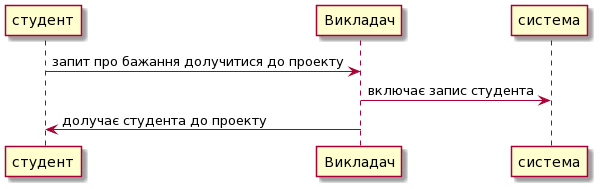
**Результат:** Студент залучений до проекту.

**Сценарій:**

1. Студент заходить до особистого кабінету.
2. Студент надсилає Викладачу повідомлення про бажання долучитися до проекту.
3. Студент виходить із особистого кабінету.
4. Викладач отримує повідомлення.
5. Викладач включає запис Студента до бази даних.
6. Викладач додає особистий кабінет Студента.
7. Викладач перевіряє наявність змін на сайті.

**Виключні ситуації:**

* 1. Порушено зв’язок між сервером із базою даних та сайтом.



* 1. **Назва:** сценарій видалення з проекту.

**Учасники:** Студент, Викладач.

**Передумови:** Студент на сайті.

**Результат:** Студент усунений від проекту.

**Сценарій:**

1. Викладач заходить до особистого кабінету.
2. Викладач надсилає студенту повідомлення про вилучення з проекту.
3. Викладач виходить із особистого кабінету.
4. Студент отримує повідомлення.
5. Викладач виключає запис Студента з бази даних.
6. Викладач видаляє особистий кабінет Студента.
7. Викладач перевіряє наявність змін на сайті.

**Виключні ситуації:**

* 1. Порушено зв’язок між сервером із базою даних та сайтом.
  2. **Назва:** сценарій з деактивуванням акаунту.

**Учасники:** Студент, система.

**Передумови:** Студент на сайті.

**Результат:** Студент деактивував акаунт.

**Сценарій**

1. Студент робить запит на видалення акаунту
2. Система робить запит на підтвердження дії
3. Студент погоджується
4. Система видаляє акаунт

**7.5 Назва:** сценарій з створенням проекту.

**Учасники:** Викладач, система.

**Передумови:** Викладач на сайті.

**Результат:** Викладач створив проект.

**Сценарій**

1.викладач робить запит на створення проекту

2.система Створює проект

* 1. **Назва:** сценарій редагування проекту.

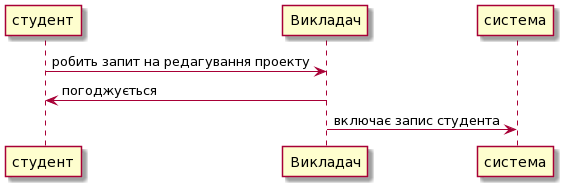
**Учасники:** Студент, Викладач.

**Передумови:** Студент на сайті.

**Результат:** проект відредагованний.

**Сценарій:**

1. Студент робить запит на редагування проекту
2. Викладач погоджується та включає запис студента



**7.7 Назва:** сценарій з видаленням акаунту користувача.

**Учасники:** Система, Адміністратор.

**Передумови:** Адміністратор на сайті.

**Результат:** Адміністратор видалив користувача.

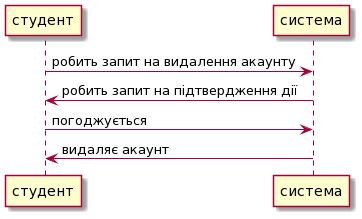
**Сценарій**

1.адміністратор робить запит на видалення акаунту

2. система робить запит на підтврдження.

3. адміністратор погоджується.

4. система видаляє акаунт.



**7.8 Назва:** сценарій з відновленням паролю.

**Учасники:** Система, Адміністратор.

**Передумови:** Студент на сайті.

**Результат:** Студент змінив пароль.

**Сценарій**

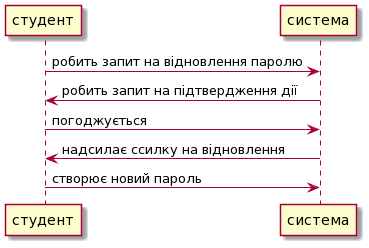
**1.**Студент робить запит на відновлення паролю

2.система робитьб запит на підтвердеження паролю

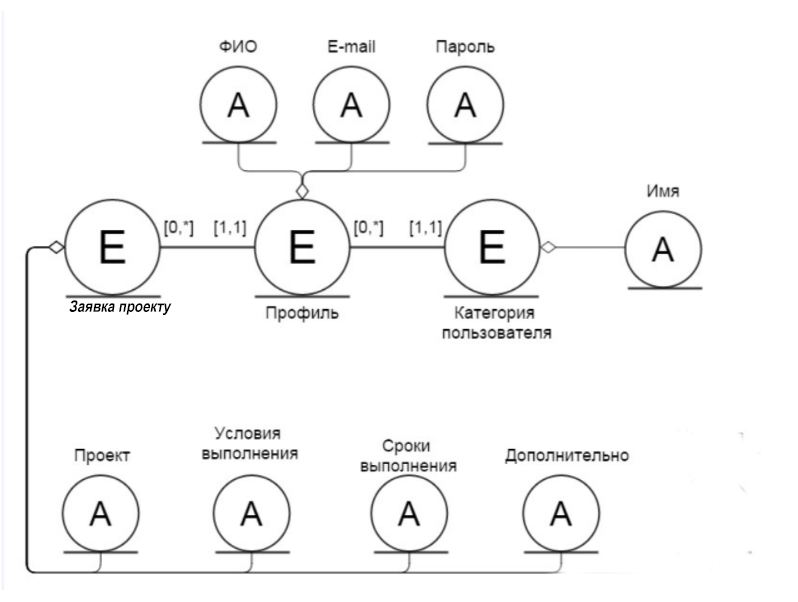
3. студент погоджується

4. система надсилає ссилку на відновлення

5. створює новий пароль



**Діаграма сутності:**



**Подання про користувача**

Профіль - внутрішнє уявлення користувача, яке задається атрибутами:

-ПІБ;

-E-mail;

-Пароль.

Категорія користувача - сукупність людей, які мають спільні правами, роллю, можливостями, доступом. Задається атрибутом:

-Тип користувача.

Заявка проекту- подається Викладачем, задається атрибутами:

-проект;

-Терміни виконання;

-додатково;

-Умови виконання.

**атрибути**

ПІБ - атрибут профілю;

Пароль - атрибут профілю;

E-mail - атрибут профілю;

Тип користувача - атрибут категорії користувача;

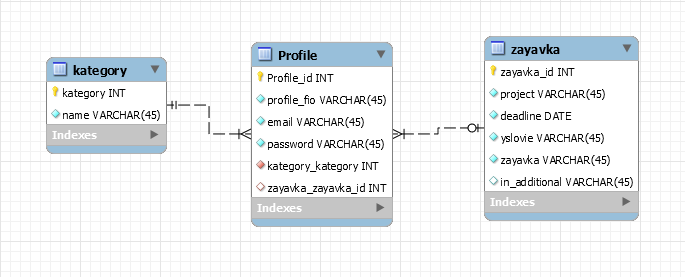
Проект - атрибут заявки проекту;

Терміни виконання - атрибут заявки проекту;

Додатково - атрибут заявки проекту;

Умови виконання - атрибут заявки проекту.

Mysql workbench диаграмма:



Лістинг програми:

-- MySQL Script generated by MySQL Workbench

-- Mon Jan 17 00:21:33 2021

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

DROP SCHEMA IF EXISTS `mydb` ;

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;

USE `mydb` ;

-- -----------------------------------------------------

-- Table `mydb`.`kategory`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`kategory` ;

CREATE TABLE IF NOT EXISTS `mydb`.`kategory` (

`kategory` INT NOT NULL,

`name` VARCHAR(45) NOT NULL,

PRIMARY KEY (`kategory`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`zayavka`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`zayavka` ;

CREATE TABLE IF NOT EXISTS `mydb`.`zayavka` (

`zayavka\_id` INT NOT NULL,

`project` VARCHAR(45) NOT NULL,

`deadline` DATE NOT NULL,

`yslovie` VARCHAR(45) NOT NULL,

`zayavka` VARCHAR(45) NOT NULL,

`in\_additional` VARCHAR(45) NULL,

PRIMARY KEY (`zayavka\_id`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`Profile`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`Profile` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Profile` (

`Profile\_id` INT NOT NULL,

`profile\_fio` VARCHAR(45) NOT NULL,

`email` VARCHAR(45) NOT NULL,

`password` VARCHAR(45) NOT NULL,

`kategory\_kategory` INT NOT NULL,

`zayavka\_zayavka\_id` INT NULL,

PRIMARY KEY (`Profile\_id`),

INDEX `fk\_Profile\_kategory\_idx` (`kategory\_kategory` ASC) VISIBLE,

INDEX `fk\_Profile\_zayavka1\_idx` (`zayavka\_zayavka\_id` ASC) VISIBLE,

CONSTRAINT `fk\_Profile\_kategory`

FOREIGN KEY (`kategory\_kategory`)

REFERENCES `mydb`.`kategory` (`kategory`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Profile\_zayavka1`

FOREIGN KEY (`zayavka\_zayavka\_id`)

REFERENCES `mydb`.`zayavka` (`zayavka\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;