Функціональна специфікація

*(Огляд вимог, де потрібно описати те, що система повинна робити*)

# Короткий огляд системи

## загальні положення

Основною причиною ініціювання даної роботи є потреба в розробці сервісу для зберігання та реалізації алгоритмів, що виражається в:

* Необхідності збереження алгоритмів (мат.формул).
* В Їх виконанні.
* Можливості розповсюдження алгоритмами між користувачами

## призначення системи

Система призначена для збереження, виконання та розповсюдження алгоритмів.

Основними учасниками та користувачами системи будуть:

1. Розробники алгоритмів
2. Професори.

Додатковими учасниками системи можуть бути:

1. Студенти.

# Короткий опис головних модулів системи

В результаті своєї роботи, система повинна зберігати , виконувати та розповсюджувати алгоритми .

Загальна схема роботи системи може мати наступний вигляд:

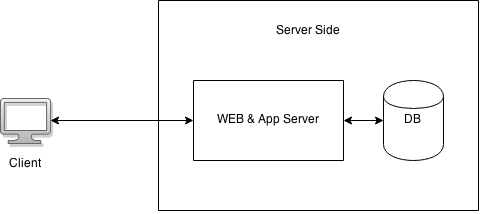


Рис. 1.1. Приблизна схема системи в загальному вигляді.

## Пояснення до схеми.

Клієнт відправляє запит на сервер щодо редагування, зберігання, виконання або видалення алгоритму. Сервер у свою чергу згідно до запиту клієнта виконує операції з інформацією, що зберігається в базі даних.

# Функції системи

## Для Користувачів система дозволяє:

1. Зберігати алгоритми.
2. Виконувати алгоритми.
3. Надавати доступ до алгоритмів.

## Загальні функції системи:

1. Зберігання алгоритмів.
2. Виконання алгоритмів.
3. Розповсюдження алгоритмів.

## Загальні вимоги до функцій та учасників :

1. Клієнт
   1. Наявність елементарних знань в області алгоритмізації в учасників.
   2. Наявність елементарних навичок роботи з браузером.
2. Функції
   1. Розділення прав здійснюється за допомогою таблиці доступу в базі даних.
   2. Редагувати алгоритм може тільки користувач, що виклав алгоритм
   3. Виконати алгоритм можуть тільки ті користувачі, яким доступна тека з алгоритмом.
   4. Виконання проводиться через спеціальний інтерфейс.

# Алгоритм функціонування системи

Більш детальна схема функціонування системи та дії по її ініціалізації мають наступний вигляд :

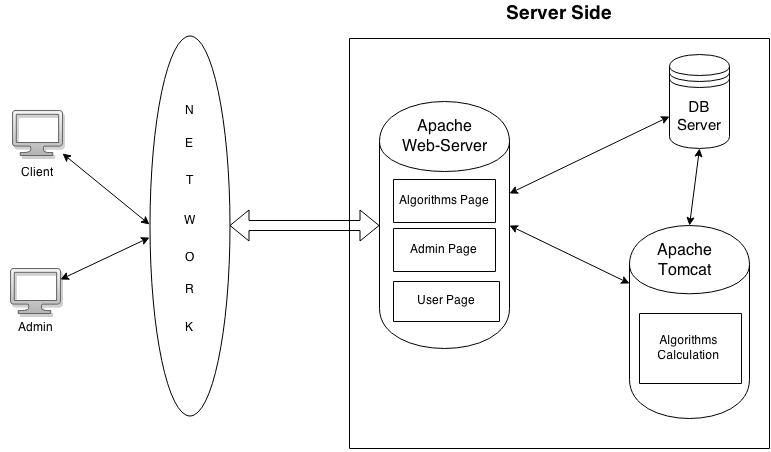


Рис. 1.2. Схема функціонування системи.

## Пояснення до схеми.

Клієнт або адміністратор авторизуються в системі. Після чого переходять в залежності від прав доступу або на сторінку адміністратора, або на сторінку користувача. Адміністратор має можливість редагувати список користувачів та алгоритмів. Користувач в свою чергу може продивлятися алгоритми, редагувати їх, та надавати для використання іншим користувачам.

## Алгоритм підготовки системи до функціонування:

### Для клієнтів:

1. Клієнт, який хоче користуватися сервісом, заходить через web-browser на “login Page” web-сервера та авторизується, або реєструється.
2. Сервлет обробляє анкету (перевіряє на наявність, тощо), створює в БД нового юзера, генерує для нього Login & Password;
3. Клієнт приступає до роботи з алгоритмами.

## Алгоритм виконання алгоритму:

1. Він отримує можливість знайти необхідний для роботи алгоритм , або створює новий.
2. Web-server передає запит на Tomcat. Tomcat завантажує дані про алгоритм з бази даних.
3. Tomcat генерує сторінку роботи з алгоритмом та передає її на Web-server. Web-server передає її користувачу.
4. Користувач працює з алгоритмом.
5. Користувач запускає процес виконання алгоритму.Web-server передає Tomcat-у вхідні дані до алгоритму, задані користувачем.
6. Сервлет на Tomcat виконує обчислення алгоритму.
7. Tomcat передає вихідні дані на Web-server. Web-server передає їх користувачу.

## Перелік даних, які зберігаються в системі

### Дані користувачів:

1. Персональні дані користувачів
2. Алгоритми користувачів.

# Екранні форми системи

Серверна частина системи буде мати приблизно такі сторінки з екранними формами:

## Для користувачів

1. Login;
2. “User page”.
3. Page “My Personal Info”;
4. Page “My Algorithms”;
5. Page “Algorithm editing” ;

***Для адміністратора системи –***

1. Login;
2. “Administrator page”;
3. “All algorithms”;
4. “Users”.