

Технічне завдання № 1

Розробити об'єктно-орієнтовану систему керування магазину товарів, яка дозволяє зареєстрованим користувачам робити замовлення товарів різних типів та надсилати SMS повідомлення у випадку рекламних акцій та днів народжень. Паролі користувачів не повинні зберігатись у відкритому вигляді, також персональні данні користувача не повинні бути доступні іншим особам. Для збереження необхідної інформації скористатись текстовими файлами власного формату даних

У базі даних (текстових файлах) повинна зберігатись наступна інформація:

- Інформація про користувача: ПІБ; дата народження; адреса доставки товару; Номер телефону (для відправлення повідомлень); електронна адреса; роль (Administrator|Advanced user| User). Додавати, видаляти та редагувати інформацію про типи товарів;
- Інформація про замовлення користувача: дата замовлення, тип товару (food, drinks, clothes), кількість одиниць

Передбачити розподілений доступ до бази даних на основі визначених ролей:

- Administrator - повинен мати можливість додавати/видаляти користувачів та переглядати дозволу інформацію користувачів (все окрім персональних даних);
- Advanced user - повинен мати можливість переглядати дозволу інформацію користувачів (все окрім персональних даних); відправляти SMS-повідомлення (або електронні листи) користувачам у випадку рекламних акцій (в залежності від типів товарів, які були придбані користувачами). Також повідомляти про бонусні знижки до дня народження користувача.
- User – повинен мати можливість, реєструватись в системі, робити замовлення різних типів товарів, переглядати повну інформацію про себе та свої замовлення, також видаляти всю інформацію про себе згідно GDPR.

Кожна операція в системі має супроводжуватись логуванням, на основі якого можна відновити базу даних на випадок втрати всіх даних.

Запропонуйте варіант захисту (шифрування/дешифрування) бази даних та системи логування.

Під час розробки програмного продукту повинні бути враховані методи безпечного програмування – перевірка вхідних даних та захист персональних даних користувача.

Розроблений функціонал повинен бути протестований за допомогою unit tests framework.

Програмне забезпечення розмістити на GitHub разом з документацією (звіт виробничої практики), яка містить опис архітектури системи (опис всіх розроблених компонент з UML діаграмами класів).

Технічне завдання № 2

Розробити об'єктно-орієнтовану систему бібліотеки, яка дозволяє зареєстрованим користувачам обирати книги на певний період часу та надсилати SMS-нагадування про повернення книг. Паролі користувачів не повинні зберігатись у відкритому вигляді, також персональні данні користувача не повинні бути доступні іншим особам. Для збереження необхідної інформації скористатись текстовими файлами власного формату даних

У базі даних (текстових файлах) повинна зберігатись наступна інформація:

- Інформація про користувача: ПІБ; дата народження; адреса проживання; номер телефону; електронна адреса; роль (Administrator| Advanced user| User). Додавати, видаляти та редагувати інформацію про типи книг;
- Інформація про книги користувача: дата взяття, термін (кількість днів), тип книги (fantasy, criminal, drama), кількість одиниць

Передбачити розподілений доступ до бази даних на основі визначених ролей:

- Administrator - повинен мати можливість додавати/видаляти користувачів та переглядати дозволена інформація користувачів (все окрім персональних даних);
- Advanced user - повинен мати можливість переглядати дозволена інформація користувачів (все окрім персональних даних); відправляти SMS-повідомлення (або електронні листи) користувачам у випадку, якщо книги не повернули вчасно.
- User – повинен мати можливість реєструватись в системі, обирати книги для себе, переглядати повну інформацію про себе та список книг, також видаляти всю інформацію про себе згідно GDPR.

Кожна операція в системі має супроводжуватись логуванням, на основі якого можна відновити базу даних на випадок втрати всіх даних.

Запропонуйте варіант захисту (шифрування/дешифрування) бази даних та системи логування.

Під час розробки програмного продукту повинні бути враховані методи безпечного програмування – перевірка вхідних даних та захист персональних даних користувача.

Розроблений функціонал повинен бути протестований за допомогою unit tests framework.

Програмне забезпечення розмістити на GitHub разом з документацією (звіт виробничої практики), яка містить опис архітектури системи (опис всіх розроблених компонент з UML діаграмами класів).

Технічне завдання № 3

Розробити об'єктно-орієнтовану систему станції технічного обслуговування автомобілів, яка дозволяє зареєстрованим користувачам користуватись послугами станції та надсилати SMS-повідомлення по мірі готовності авто. Паролі користувачів не повинні зберігатись у відкритому вигляді, також персональні данні користувача не повинні бути доступні іншим особам. Для збереження необхідної інформації скористатись текстовими файлами власного формату даних

У базі даних (текстових файлах) повинна зберігатись наступна інформація:

- Інформація про користувача: ПІБ; дата народження; адреса проживання; номер телефону; електронна адреса; роль (Administrator| Advanced user| User).
- Інформація про авто користувача: дата взяття на ремонт, тривалість ремонту (кількість годин/днів), тип авто (sedan, cabriolet, offroad car), ціна технічного обслуговування

Передбачити розподілений доступ до бази даних на основі визначених ролей:

- Administrator - повинен мати можливість додавати/видаляти користувачів та переглядати дозволenu інформацію користувачів (все окрім персональних даних). Додавати, видаляти та редагувати інформацію про типи авто;
- Advanced user - повинен мати можливість переглядати дозволenu інформацію користувачів (все окрім персональних даних); відправляти SMS-повідомлення користувачам по мірі готовності авто.
- User – повинен мати можливість реєструватись в системі, записуватись на ремонт, переглядати повну інформацію про себе та поточний стан ремонту, також видаляти всю інформацію про себе згідно GDPR.

Кожна операція в системі має супроводжуватись логуванням, на основі якого можна відновити базу даних на випадок втрати всіх даних.

Запропонуйте варіант захисту (шифрування/дешифрування) бази даних та системи логування.

Під час розробки програмного продукту повинні бути враховані методи безпечного програмування – перевірка вхідних даних та захист персональних даних користувача.

Розроблений функціонал повинен бути протестований за допомогою unit tests framework.

Програмне забезпечення розмістити на GitHub разом з документацією (звіт виробничої практики), яка містить опис архітектури системи (опис всіх розроблених компонент з UML діаграмами класів).

Технічне завдання № 4

Розробити об'єктно-орієнтовану систему оренди тенісних кортів, яка дозволяє зареєстрованим користувачам бронювати kort на певний період та надсилати SMS-нагадування за пару годин до бронювання. Паролі користувачів не повинні зберігатись у відкритому вигляді, також персональні данні користувача не повинні бути доступні іншим особам. Для збереження необхідної інформації скористатись текстовими файлами власного формату даних

У базі даних (текстових файлах) повинна зберігатись наступна інформація:

- Інформація про користувача: ПІБ; дата народження; адреса проживання; номер телефону; електронна адреса; роль (Administrator| Advanced user| User).
- Інформація про тенісні корти – можливі години бронювання по кожному корту.
- Інформація про бронь користувача: дата початку, тривалість оренди (кількість годин/днів), номер корту, ціна бронювання

Передбачити розподілений доступ до бази даних на основі визначених ролей:

- Administrator - повинен мати можливість додавати/видаляти користувачів та переглядати дозволена інформація користувачів (все окрім персональних даних). Додавати, видаляти та модифікувати інформацію про тенісні корти;
- Advanced user - повинен мати можливість переглядати дозволена інформація користувачів (все окрім персональних даних); відправляти SMS-нагадування користувачам за пару годин до початку дії бронювання корту.
- User – повинен мати можливість реєструватись в системі, бронювати kort, переглядати повну інформацію про себе та свої бронювання, також видаляти всю інформацію про себе згідно GDPR.

Кожна операція в системі має супроводжуватись логуванням, на основі якого можна відновити базу даних на випадок втрати всіх даних.

Запропонуйте варіант захисту (шифрування/дешифрування) бази даних та системи логування.

Під час розробки програмного продукту повинні бути враховані методи безпечного програмування – перевірка вхідних даних та захист персональних даних користувача.

Розроблений функціонал повинен бути протестований за допомогою unit tests framework.

Програмне забезпечення розмістити на GitHub разом з документацією (звіт виробничої практики), яка містить опис архітектури системи (опис всіх розроблених компонент з UML діаграмами класів).