**Дніпропетровський ліцей інформаційних технологій**

**при Дніпропетровському національному університеті**

**імені Олеся Гончара**

**Завдання заочного туру**

**Турніру Юних Інформатиків 2015**

**Команда OMEGA в складі:**

**Софійченко Ігор – капітан**

**Байнак Дмитро – член команди**

**Шаталович Ігнатій – член команди**

**Керівники роботи:**

**Малоіван І.Є.**

**Шастітко Н.О.**

**Дніпропетровськ**

**2015**

**ЗМІСТ**

[ВСТУП 5](#_Toc434189338)

[ОСНОВНА ЧАСТИНА 6](#_Toc434189339)

[РОЗДІЛ 1. ОПИС ВИКОРИСТАНОГО ПЗ 6](#_Toc434189340)

[ПІДРОЗДІЛ 1.1. IntelliJ IDEA 6](#_Toc434189341)

[ПІДРОЗДІЛ 1.2. GlassFish 6](#_Toc434189342)

[ПІДРОЗДІЛ 1.3. PostgreSQL 6](#_Toc434189343)

[РОЗДІЛ 2. ОПИС СТВОРЕНИХ БАЗ ДАНИХ 8](#_Toc434189344)

[ПІДРОЗДІЛ 2.1. VOLUNTEERS 8](#_Toc434189345)

[ПІДРОЗДІЛ 2.2. HRESOURSES 8](#_Toc434189346)

[ПІДРОЗДІЛ 2.3. REQUESTS 9](#_Toc434189347)

[ПІДРОЗДІЛ 2.4. STOCKS 9](#_Toc434189348)

[ПІДРОЗДІЛ 2.5. NRESOURCES 9](#_Toc434189349)

[ПІДРОЗДІЛ 2.6. RESOURCES 10](#_Toc434189350)

[ПІДРОЗДІЛ 2.7. TRANSPORT 10](#_Toc434189351)

[ПІДРОЗДІЛ 2.8. WAY 10](#_Toc434189352)

[РОЗДІЛ 3. ПОСІБНИК КОРИСТУВАЧА 11](#_Toc434189353)

[ПІДРОЗДІЛ 3.1. НЕОБХІДНЕ ПЗ 11](#_Toc434189354)

[ПІДРОЗДІЛ 3.2. НАЛАШТУВАННЯ СЕРЕДОВИЩА І ЗАПУСК ВЕБ–ІНТЕРФЕЙСУ ПРОГРАМИ 12](#_Toc434189355)

[ПІДРОЗДІЛ 3.3. ОПИС ВЕБ-ІНТЕРФЕЙСУ 15](#_Toc434189356)

[ВИСНОВОК 16](#_Toc434189357)

[ЗАВДАННЯ 1 16](#_Toc434189358)

[ПУНКТ 1.1 16](#_Toc434189359)

[ПУНКТ 1.2 16](#_Toc434189360)

[ПУНКТ 1.3 16](#_Toc434189361)

[ПУНКТ 1.4 16](#_Toc434189362)

[ПУНКТ 1.5 16](#_Toc434189363)

[ЗАВДАННЯ 2 17](#_Toc434189364)

[ПУНКТ 2.1 17](#_Toc434189365)

[ПУНКТ 2.2 17](#_Toc434189366)

[ПУНКТ 2.3 17](#_Toc434189367)

[ПУНКТ 2.4 17](#_Toc434189368)

[ПУНКТ 2.5 17](#_Toc434189369)

[ПУНКТ 2.6 18](#_Toc434189370)

[ЗАВДАННЯ 3 18](#_Toc434189371)

[ПУНКТ 3.1 18](#_Toc434189372)

[ПУНКТ 3.2 18](#_Toc434189373)

[ПУНКТ 3.3 18](#_Toc434189374)

[ПУНКТ 3.4 18](#_Toc434189375)

[ПУНКТ 3.5 19](#_Toc434189376)

[ЗАВДАННЯ 4 19](#_Toc434189377)

[ПУНКТ 4.1 19](#_Toc434189378)

[ПУНКТ 4.2 19](#_Toc434189379)

[ПУНКТ 4.3 19](#_Toc434189380)

[ПУНКТ 4.4 19](#_Toc434189381)

[ПУНКТ 4.5 20](#_Toc434189382)

[ПУНКТ 4.6 20](#_Toc434189383)

[ПУНКТ 4.7 20](#_Toc434189384)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ІНФОРМАЦІЇ 21](#_Toc434189385)

[ДОДАТКИ 22](#_Toc434189386)

[ДОДАТОК 1 22](#_Toc434189387)

[ДОДАТОК 2 22](#_Toc434189388)

[ДОДАТОК 3 23](#_Toc434189389)

# ВСТУП

В останній час ми почали все частіше чути про волонтерський рух, волонтерів. І якщо десь на теренах нашої держави з’являється нагальна життєва потреба у ліках, одягу, спеціальних засобах тощо, обов’язково з’являються небайдужі люди, що бажають допомогти. Військові дії, природні катаклізми, техногенні аварії та катастрофи – все це може стати об’єктами зосередження уваги волонтерів.

Але буває і так, що допомога приходить не завжди вчасно, буває так, що в одних місцях густо, а в інших пусто. Тож, виникає потреба у координації волонтерських зусиль. Сьогодні цим займаються спеціальні волонтерські організації. Вони збирають речі, медикаменти, обладнання на місцях, а потім пересилають у пункти, де вони потрібні. Транспортуванням допомоги займаються інші волонтери, які або надають транспортні засоби, або самі везуть вантаж у місця призначення. Дорога може бути небезпечною, ситуація в регіонах швидко змінюватися, і перевізники мають знати про ці зміни для того, щоб обрати найбезпечніший та найшвидший маршрут.

Волонтерським організаціям нагально потрібен інструмент, що допоможе вирішувати цілу низку виникаючих проблем. Цей інструмент – комплекс програмного забезпечення, що дозволить вести бази даних волонтерів, наявних ресурсів та потреб в цих ресурсах. Також він включатиме систему ефективного керування запасами, розподілу ресурсів, автоматичного визначення оптимальних маршрутів в динаміці.

# ОСНОВНА ЧАСТИНА

## РОЗДІЛ 1. ОПИС ВИКОРИСТАНОГО ПЗ

### ПІДРОЗДІЛ 1.1. IntelliJ IDEA

IntelliJ IDEA (див. ДОДАТОК 1) — комерційне інтегроване середовище розробки для різних мов програмування (Java, Python, Scala, PHP та ін.) від компанії JetBrains.

Особливостями програми є:

* Розумне автодоповнення, інструменти для аналізу якості коду, зручна навігація, розширені рефакторинги й формутування для багатьох мов програмування.
* Підримка усіх популярних фреймворків і платформ.
* Інтеграція з серверами додатків.
* Інструменти для роботи з базами даних і SQL–файлами, включаючи зручний клієнт і редактор для схеми баз даних.
* Інструменти для запуску тестів і аналізу покриття коду, включаючи підтримку усіх популярних фреймворків для тестування.

У рамках поставленої задачі, це програмне забезпечення використовується для реалізації веб–інтерфейсу за допомогою JavaServer Faces.

### ПІДРОЗДІЛ 1.2. GlassFish

GlassFish (див. ДОДАТОК 2) — сервер додатків з відкритим сирцевим кодом, який реалізує специфікації Java EE. Спочатку розроблений Sun Microsystems, після поглинання цієї компанії спонсорується корпорацією Oracle. Актуальна версія платформи називається Oracle GlassFish Server.

### ПІДРОЗДІЛ 1.3. PostgreSQL

PostgreSQL — об'єктно–реляційна система керування базами даних (СКБД). Розробка цього ПЗ можлива завдяки співпраці багатьох людей та компаній, які хочуть використовувати цю СКБД та впроваджувати у неї найновіші досягнення. Сервер PostgreSQL написаний на мові C.

Особливостями цієї СКБД є:

* Вона є вкрай надійною: на 775000 рядків коду було знайдено лише 20 проблемних місць.
* Можливість виконувати деякі фунції самим сервером бази даних, які можуть бути написані як на SQL, так і на "класичних" мовах програмування (C, C++, Java).
* PostgreSQL підтримує одночасну модифікацію БД декількома користувачами за допомогою механізму Multiversion Concurrency Control (MVCC).
* Підтримка великого типу вбудованих типів даних.
* Таблиці можуть успадковувати характеристики та набори полів від інших таблиць (батьківських).

## РОЗДІЛ 2. ОПИС СТВОРЕНИХ БАЗ ДАНИХ

### ПІДРОЗДІЛ 2.1. VOLUNTEERS

Ця база даних є переліком волонтерів, які зареєстровані у системі.

* u\_id – унікальний ідентифікатор, який видається при реєстрації і є особливим для кожного.
* login – це ім'я під яким цей волонтер заходить в базу даних.
* pass – це пароль за допомогою якого система удостовірюється що це саме цей волонтер.
* sname – це реальне прізвище волонтера.
* fname – це реальне ім'я волонтера.
* address – це адреса електронної пошти волонтера.
* phone – це номер телефону волонтера.
* access – це рівень прав доступу. Так, волонтер може бути як звичайним користувачем, так і адміністратором системи.
* hash – це параметр, який використовується для перевірки достовірності аккаунта волонтера.

### ПІДРОЗДІЛ 2.2. HRESOURSES

Ця база даних є переліком усіх ресурсів, які зберігаються на складах.

* hres\_id – це унікальний ідентифікатор даного ресурсу.
* u\_id – характеризує якому волонтеру даний ресурс належить.
* stock\_id – характеризує на якому складі знаходиться даний ресурс.
* resource – назва ресурсу.
* count – кількість ресурсу, яка доступна.
* type – це параметр, який характеризує даний ресурс з фізичної точки зору, тобто маса, довжина і т.п.
* date – дата, коли ресурс був отриманий.

### ПІДРОЗДІЛ 2.3. REQUESTS

Ця база даних є переліком усіх запитів на отримання певної кількості ресурсів.

* req\_id – унікальний ідентифікатор цього запиту.
* u\_id – унікальний ідентифікатор людини, яка зробила цей запит.
* stock\_id – склад, куди цей ресурс необхідно привезти.
* resource – назва ресурсу
* count – кількість ресурсу.
* type – це параметр, який характеризує даний ресурс з фізичної точки зору, тобто маса, довжина і т.п.
* date – дата видачі ресурси, тобто коли запит був оброблений.

### ПІДРОЗДІЛ 2.4. STOCKS

Ця база даних є переліком усіх складів.

* stock\_id – унікальний ідентифікатор даного складу.
* st\_name – назва складу.
* address – адреса, за якою цей склад знаходиться.
* work\_time – час роботи складу.
* phone – це номер телефону склада.

### ПІДРОЗДІЛ 2.5. NRESOURCES

Ця база даних є переліком усіх необхідних ресурсів.

* nres\_id – унікальний ідентифікатор цього ресурсу.
* u\_id – ідентифікатор людини, якій необхідний цей ресурс.
* address – адреса, де цей ресурс є необхідним.
* resource – назва ресурсу.
* count – кількість одиниць ресурсу, які доступні.
* type – це параметр, який характеризує даний ресурс з фізичної точки зору, тобто маса, довжина і т.п.
* active – параметр, який характеризує чи залишилась необхідність у цьому ресурсі, чи ні.
* date – дата, коли з’явилася необхідність у цьому ресурсі.

### ПІДРОЗДІЛ 2.6. RESOURCES

Ця база даних є переліком ідентифікаторів і назв усіх ресурсів.

* res\_id – унікальний ідентифікатор цього ресурсу.
* resource – назва цього ресурсу.
* type – це параметр, який характеризує даний ресурс з фізичної точки зору, тобто маса, довжина і т.п.

### ПІДРОЗДІЛ 2.7. TRANSPORT

Ця база даних є переліком транспортів, які доступні волонтерам.

* tr\_id – унікальний ідентифікатор цього транспорту.
* number – унікальний номер цього транспорту (який відповідає державному).
* descr – описання цього транспорту.
* type – назва типу транспорту.
* address – адреса, де перебуває транспорт.
* phone – номер телефону володаря цього транспорту.
* work\_time – час, під час якого цей транспорт є доступним.

### ПІДРОЗДІЛ 2.8. WAY

Ця база даних є переліком маршрутів, які поєднують певні дві адреси.

* way\_id – унікальний ідентифікатор цього маршруту.
* address\_1 – перша адреса, яка є початком/кінцем маршруту.
* address\_2 – друга адреса, яка є кінцем/початком маршруту, відповідно.
* length – довжина цього маршруту.
* type – тип дорожнього покриття (або бездоріжжя).

## РОЗДІЛ 3. ПОСІБНИК КОРИСТУВАЧА

### ПІДРОЗДІЛ 3.1. НЕОБХІДНЕ ПЗ

Для того, щоб користуватися даним програмним забезпеченням необхідно встановити наступні програми, відповідно до розрядності вашої операційної системи. Так, для віртуальної машини Java, комлекту розробника Java, PostgreSQL у відповідних папках є дві версії цих програм.

1. Java Runtime Environment (JRE)
   1. \Software\Java Runtime Environment\JRE\_x32.exe – 32–бітна версія
   2. \Software\Java Runtime Environment\JRE\_x64.exe – 64–бітна версія
2. Java Development Kit (JDK)
   1. \Software\ Java Development Kit\JDK\_x32.exe – 32–бітна версія
   2. \Software\ Java Development Kit\JDK\_x64.exe – 64–бітна версія

Задля зручності, рекомендую запам’ятати адресу, якою зберігатиметься це програмне забезпечення.

1. PostgreSQL
   1. \Software\PostgreSQL\PostgreSQL\_x32.exe – 32–бітна версія
   2. \Software\PostgreSQL\PostgreSQL\_x64.exe – 64–бітна версія

Також, цей інсталятор автоматично інсталює Microsoft Visual C++ 2013 (x64). Якщо виникнуть проблеми з цим ПЗ, то в папці \Software\Visual C++ Redist знаходяться всі версії цього продукту (для обох розрядностей, версії 2005, 2008, 2010, 2012, 2013).

Пароль необхідно встановити «admin».

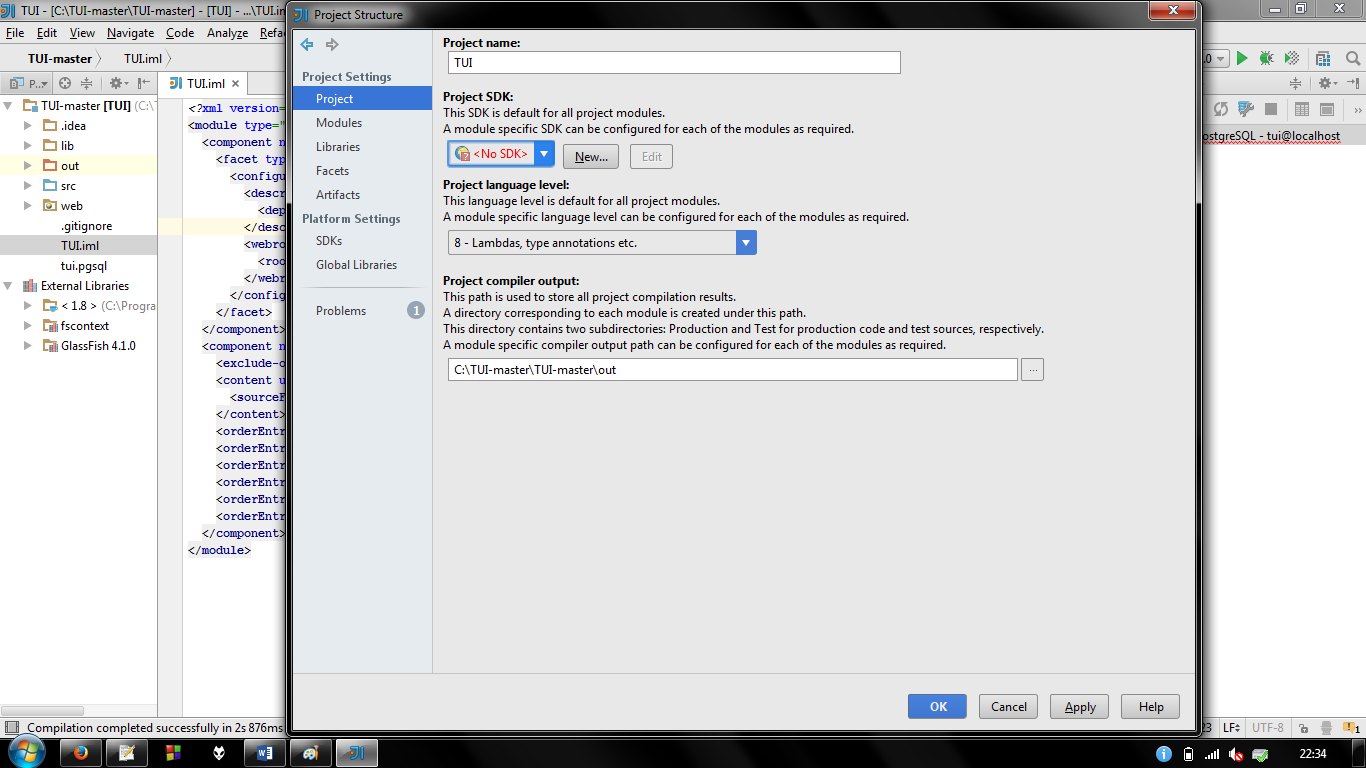
1. IntelliJ IDEA – \Software\IntelliJ IDEA\IntelliJ\_IDEA.exe

Дана програма є комерційним продуктом, але є можливість його отримати для пробного безкоштовного користування на 30 днів. Якщо не вдасться отримати на 30 днів, то у папці з інсталятором знаходиться генератор ключів. Дане ПЗ дозволяється використовувати лише в ознайомлювальних цілях.

1. Glassfish Server – це ПЗ необхідно лише скопіювати у деяке місце на жорсткому диску.

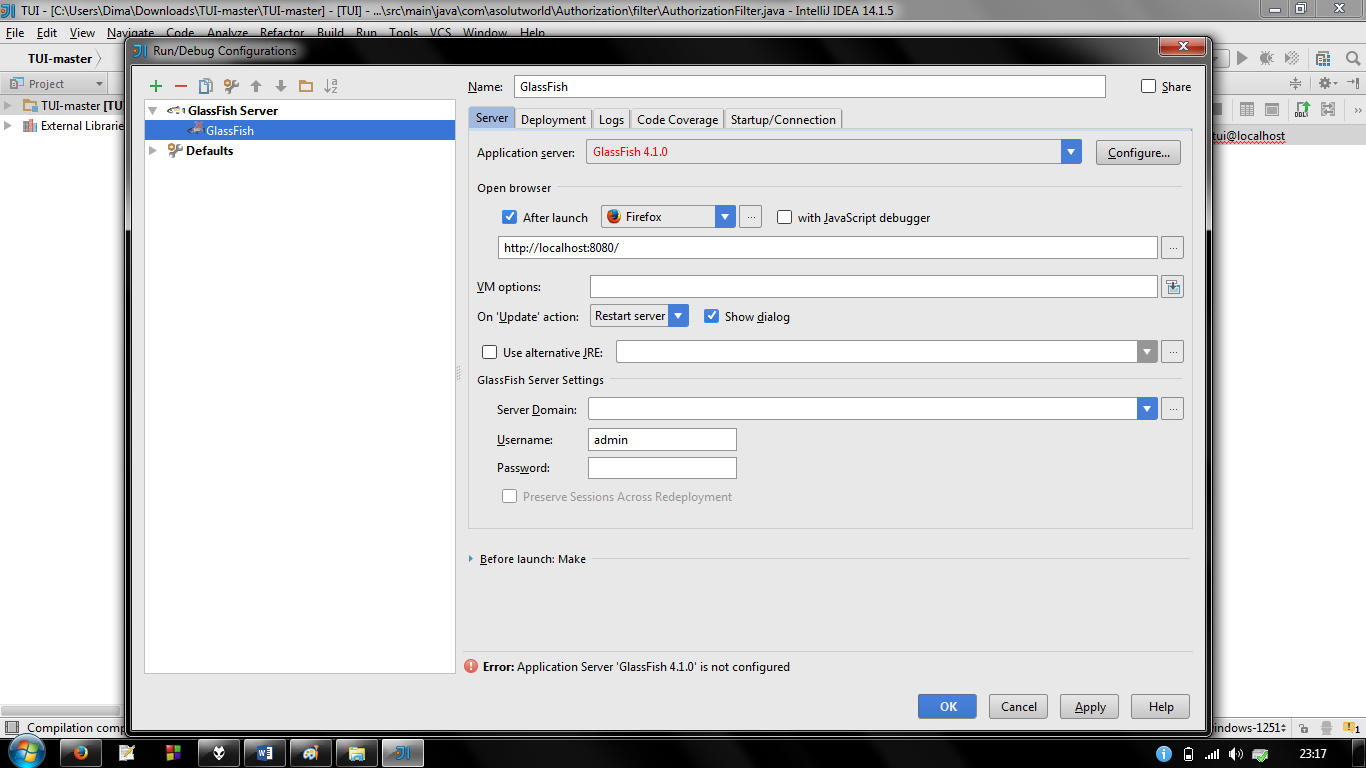
### ПІДРОЗДІЛ 3.2. НАЛАШТУВАННЯ СЕРЕДОВИЩА І ЗАПУСК ВЕБ–ІНТЕРФЕЙСУ ПРОГРАМИ

Задля запуску веб–інтерфейсу необхідно перейти шляхом \Project\TUI.iml и запустити його за допомогою IntelliJ IDEA. Необхідно відчиняти, натиснувши праву кнопку миші на цьому файлі, вибрати «Відчинити за допомогою» та вибрати IntelliJ IDEA. Після цього, відчиниться вікно середовища.

В першу чергу, необхідно додати Java SDK. Це робиться наступним чином:

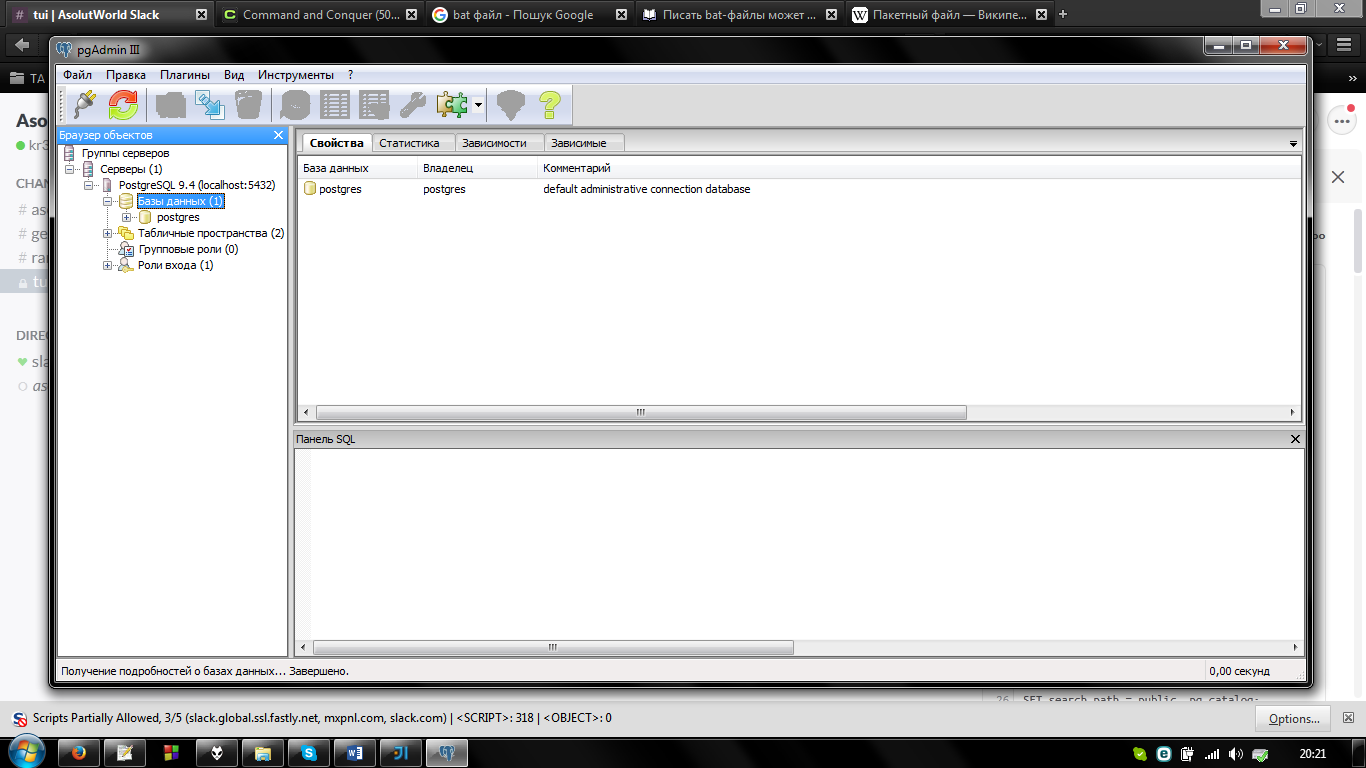
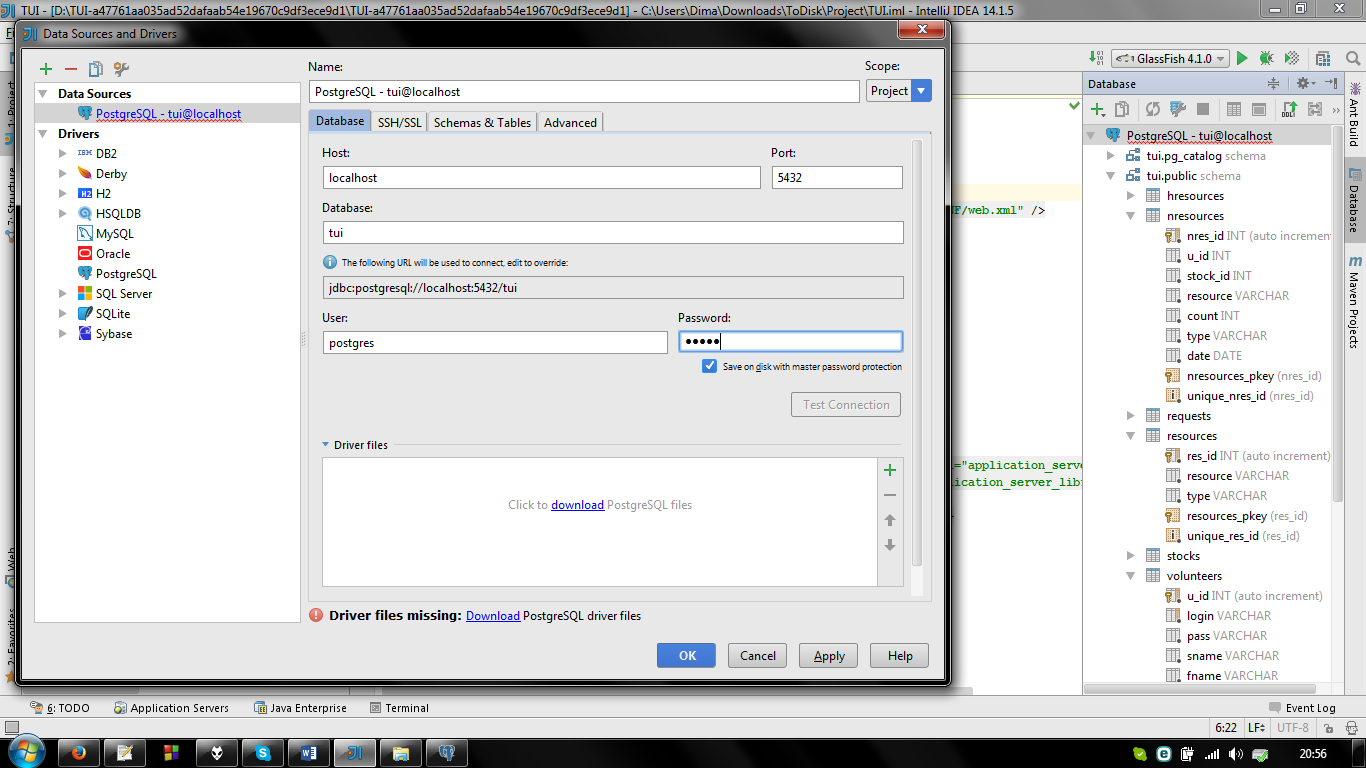
Ви переходите в вкладку File – Project Structure. Має з’явитися вікно, яке зображено справа.

Необхідно натиснути на кнопку New і у випадаючому списку вибрати JDK. Тепер нам стане в нагоді адреса JDK, яку підчас інсталяції рекомендувалося запам’ятати. Тут необхідно її і ввести. Якщо ви не корегували шлях, яким комплект розробника Java інсталювався, то він буде C:\Program Files\Java\jdk1.8.0\_66 – таким. Про всяк випадок, варто перевірити, аби Project language level був 8 – Lambdas, type annotations etc.

Тепер необхідно налаштувати сервер. Можна зачинити вікно Project Structure. Відкрийте вкладку Run – Edit Configurations. З’явиться наступне вікно:

Натисніть кнопку Configure і у полі Glassfish Home вкажіть місце, куди ви скопіювали папку, яка знаходиться на диску за шляхом \Software\Glassfish. Якщо ви скопіювали її, наприклад, в корень диску D:, то шляхом буде D:\Glassfish.

Наступне, що необхідно зробити – це розібратися з базою даних. В першу чергу, вам необхідно відкрити папку, куди ви встановили PostgreSQL. Далі, необхідно перейти в папку bin і запустити там pgAdmin3.exe. З’явиться наступне вікно:

 Необхідно тепер натиснути правою кнопкою миші на назву «База данных» і вибрати «Новая база данных». Її необхідно назвати “tui”. Далі, необхідно запустити командну строку (Win + R і ввести cmd). В першу чергу, змінимо директорію, у якій працюємо, на C:\Program Files\PostgreSQL\9.4\bin (тобто, звідки ми запускали pgAdmin3.exe). Це робиться за допомогою команди “chdir C:\Program Files\PostgreSQL\9.4\bin” (без лапок). Тепер необхідно скопіювати файл tui.pgsql, який знаходиться в папці \Project\, у папку C:\Program Files\PostgreSQL\9.4\bin і ввести команду “psql -U postgres tui < tui.pgsql” (без лапок).

Перейдемо до налаштування БД у IntelliJ. Справа необхідно вибрати PostgreSQL – tui@localhost, натиснути на нього правою кнопкою миші й вибрати Properties. Відкриється відповідне вікно, яке зображено справа. У полі Database має бути назва бази даних – tui. У полі Password – admin. Зліва напису Driver files є синій трикутник, на який треба натиснути. Тепер додаємо драйвер для PostgreSQL наступним чином: натискаємо на зелений плюс справа і задаємо шлях «\Project\web\WEB-INF\lib\postgresql-9.4-1201.jdbc4.jar» (без лапок). Натискаємо OK.

Для запуску веб–інтерфейсу у правому верхньому куточку необхідно натиснути на зелений трикутник.

### ПІДРОЗДІЛ 3.3. ОПИС ВЕБ-ІНТЕРФЕЙСУ

Тільки-но ви потрапляєте на веб-інтерфейс, вам одразу пропонується зареєструватися. Для цього необхідно вигадати логін, пароль, ввести ім’я та прізвище, поштову адресу та мобільний телефон, причому на поштову адресу приходить лист з необхідністю підтвердити реєстрацію.

Але навіть якщо не реєструватися, то завжди є адміністратор бази даних, який має логін і пароль admin.

Перша кнопка «Волонтери» переходить на сторінку, де знаходиться перелік усіх волонтерів з їх іменем, прізвищем, мобільним телефоном. Можна зайти до них у профіль і подивитися на наявність певних ресурсів.

На вкладці «Склади» є кнопки «Новий ресурс» і «Новий склад», які дають можливість створити новий тип ресурсу (вказавши його назву та розмірність) чи новий склад (вказавши його адресу, телефон, час роботи і, звичайно, назву).

У вкладці «Запити» є перелік усіх існуючих нині запитів, хто їх створив. що, де, в якій кількості необхідно певних ресурсів. Є можливість створити новий запит.

У вкладці профіль ви бачите свої ресурси, які, відповідно, ви маєте. Ви можете додати собі ресурсів або забрати деяку кількість з певного складу.

Відповідно, кнопка «Вийти» розлогінює вас і повертає на початкову сторінку. Кнопка «Скачати» - створити табличний звіт даної БД.

# ВИСНОВОК

## ЗАВДАННЯ 1

### ПУНКТ 1.1

Умова: Розробити базу даних волонтерів та ресурсів (БДВР), що вони спроможні надати.

Цей пункт реалізований повністю. База волонтерів описана в підрозділі 2.1., а база ресурсів, що вони спроможні надати, в підрозділі 2.2.

### ПУНКТ 1.2

Умова: Обґрунтувати структуру даних БДВР використовуючи нормальні форми або ER-моделі.

Цей пункт реалізовано повністю. Дивитися ДОДАТОК 3.

### ПУНКТ 1.3

Умова: Створити веб-інтерфейс БДВР для існуючих та потенціальних волонтерів.

Цей пункт реалізовано повністю. Більш детально про запуск і перевірки необхідно звернутися до посібника користувача (розділ 3).

### ПУНКТ 1.4

Умова: Створити інтерфейс керування БДВР для волонтерської організації з розмежуванням прав доступу.

Цей пункт реалізовано повністю. Кожен волонтер має різні права доступу(більш детально – підрозділ 2.1, параметр access).

### ПУНКТ 1.5

Умова: Забезпечити створення звітів БДВР у табличному та графічному вигляді для подальшого аналізу.

Цей пункт реалізовано частково, без графічного вигляду. Табличний вигляд реалізований.

## ЗАВДАННЯ 2

### ПУНКТ 2.1

Умова: Розробити базу даних потреб у ресурсах (БДПР).

Цей пункт реалізований повністю. БДПР описана в підрозділі 2.5.

### ПУНКТ 2.2

Умова: Обґрунтувати структуру даних БДПР використовуючи нормальні форми або ER-моделі.

Цей пункт реалізовано повністю. Дивитися ДОДАТОК 3.

### ПУНКТ 2.3

Умова: Створити веб-інтерфейс БДПР для отримувачів ресурсів.

Цей пункт реалізовано повністю. Більш детально про запуск і перевірки необхідно звернутися до посібника користувача, який знаходиться в розділі 3 основної частини.

### ПУНКТ 2.4

Умова: Створити інтерфейс керування БДПР для волонтерської організації з розмежуванням прав доступу.

Цей пункт реалізовано повністю. Кожен волонтер має різні права доступу(більш детально – підрозділ 2.1, параметр access).

### ПУНКТ 2.5

Умова: Забезпечити створення звітів БДПР у табличному та графічному вигляді для подальшого аналізу.

Цей пункт реалізовано частково, без графічного вигляду. Табличний вигляд реалізований.

### ПУНКТ 2.6

Умова: Створити систему захисту даних, включаючи шифрування та передачу даних через захищені канали зв’язку.

Технологія JSF включає в себе і шифрування, і передачу даних через захищені канали зв’язку.

## ЗАВДАННЯ 3

### ПУНКТ 3.1

Умова: Розробити систему керування запасами ресурсів волонтерської організації (база даних складів, отримання та відвантаження ресурсів).

Цей пункт реалізовано повністю. База даних складів описана в підрозділі 2.4, а відвантаження ресурсів – підрозділ 2.3.

### ПУНКТ 3.2

Умова: Обґрунтувати структуру системи керування запасами.

Цей пункт реалізовано повністю. Дивитися ДОДАТОК 3.

### ПУНКТ 3.3

Забезпечити зв’язок системи керування запасами ресурсів з базами даних волонтерів та потреб у запасах (див. завдання 1. і 2.)

Цей пункт реалізовано повністю. Дивитися ДОДАТОК 3, у якому описаний взаємозв’язок

### ПУНКТ 3.4

Створити підсистему автоматизованого керування запасами, яка буде рекомендувати найефективніші об’єми та моменти часу отримання та відвантаження ресурсів з метою мінімізації витрат на зберігання і максимального задоволення потреб у ресурсах.

Не реалізовано.

### ПУНКТ 3.5

Умова: Створити інтерфейс системи керування запасами для волонтерської організації.

Цей пункт реалізовано повністю. Більш детально про запуск і перевірки необхідно звернутися до посібника користувача (розділ 3).

## ЗАВДАННЯ 4

### ПУНКТ 4.1

Умова: Розробити систему обліку транспортних засобів та маршрутів доставки.

Цей пункт реалізовано повністю, транспортні засоби – це підрозділ 2.7., а маршрути – це підрозділ 2.8.

### ПУНКТ 4.2

Умова: Обґрунтувати структуру даних системи обліку засобів та маршрутів доставки.

Повне описання структури знаходиться у підрозділах 2.7-2.8.

### ПУНКТ 4.3

Умова: Забезпечити використання в системі реальних географічних карт.

Не реалізовано.

### ПУНКТ 4.4

Умова: Забезпечити можливість створення власних маршрутів та підтримку нестандартних транспортних засобів

Не реалізовано.

### ПУНКТ 4.5

Умова: Створити інтерфейс користування системою обліку транспортних засобів та маршрутів доставки.

Цей пункт виконано повністю.

### ПУНКТ 4.6

Умова: Створити підсистему автоматизованого розрахунку оптимального використання наявних транспортних засобів та оптимальних маршрутів транспортування з урахуванням чинників завантаженості доріг, небезпеки, проходимості транспортних засобів, погодних умов тощо.

Не реалізовано.

### ПУНКТ 4.7

Умова: Забезпечити надання звітів з кількох альтернативних варіантів доставки вантажу, з включенням транспортних витрат, прогнозованого терміну доставки, ризику невиконання доставки.

Не реалізовано.

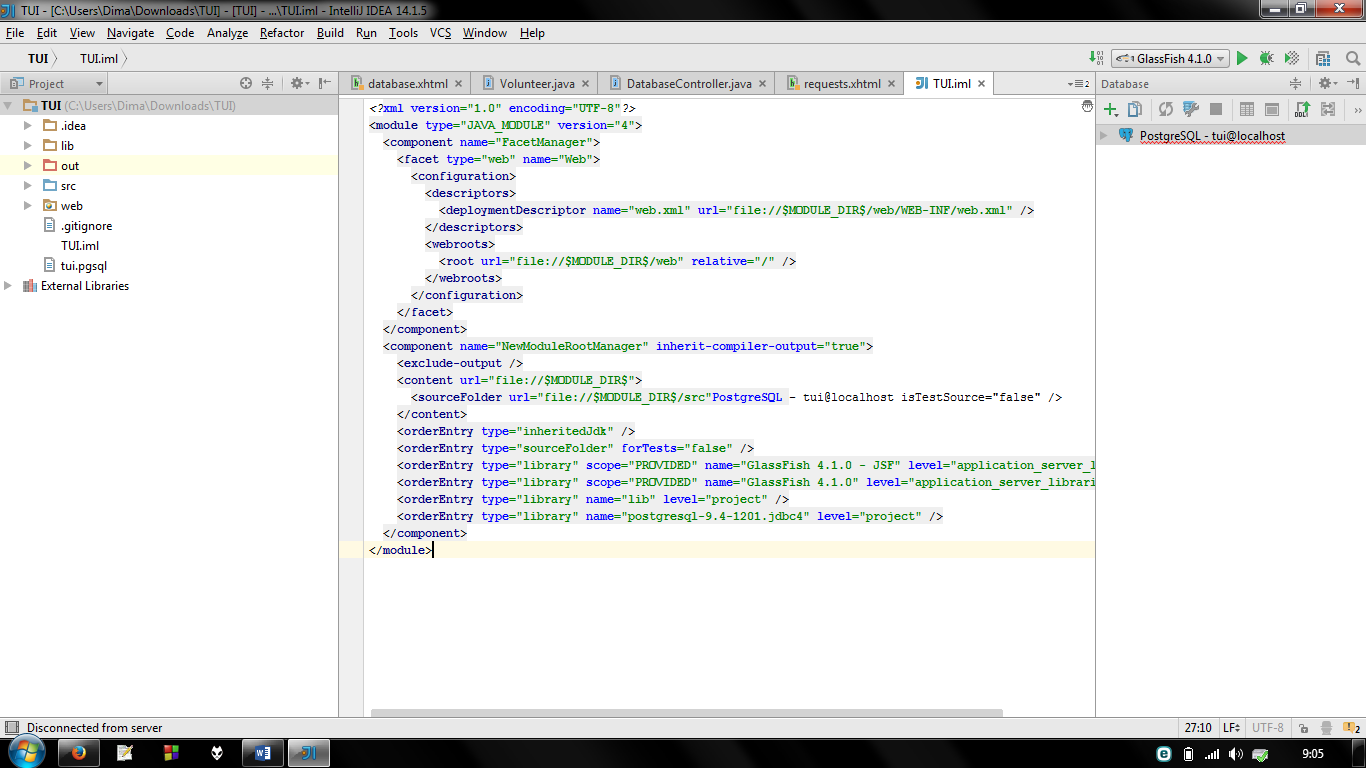
# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ІНФОРМАЦІЇ

1. <http://www.tutorialspoint.com/jsf/>
2. <http://www.vogella.com/tutorials/JavaServerFaces/article.html>
3. <http://stackoverflow.com/>

# ДОДАТКИ

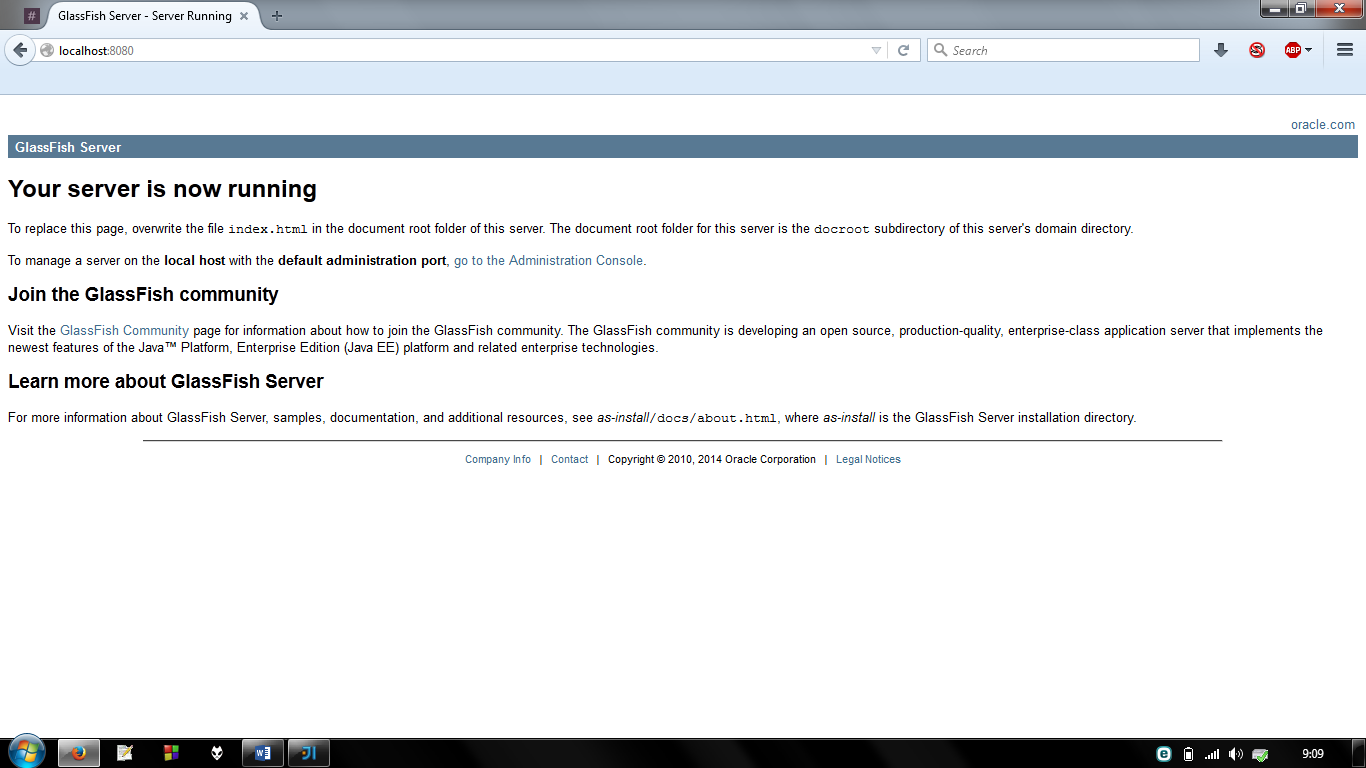
## ДОДАТОК 1

IntelliJ IDEA:



## ДОДАТОК 2

Glassfish Server:



## ДОДАТОК 3

ER-модель:

date

e

**nresources**

nres\_id

type

resource

count

address

active

e

u\_id

**hresources**

count

type

resource

nres\_id

date

stock\_id

e

u\_id

**requests**

req\_id

date

stock\_id

count

resource

type

e

u\_id

**resources**

type

res\_id

e

resource

**stocks**

st\_name

phone

e

address

work\_timeme

e

stock\_id

**volunteers**

u\_id

login

sname

address

access

pass

fname

phone

hash