**Зміст**

Вступ……………………………………………………………………………..…..4

Мета переддиплоиної практики………..……………..………………………..…...7

Перша частина……………………………………………………………….……....7

Опис бази практики………………………………………………………….7

Освоєні інформаційні технології…………………………….……………...8

Друга частина…………………………………………………………………….….9

Аналіз вимог до програмного продукту……………………………………9

Модуль авторизації користувача………………….………………………...14

Тестування..………………………………..…………………………………16

Індивідуальна частина………………………………….……………………17

Висновок…………………………….………………………………………..18

Використана література……………………………………………...……...19

Словник термінів……………………………………………………...……..19

Додаток А…………………………………………………………………...….…...21

**Вступ**

Мета переддипломної практики – поглиблене ознайомлення студентів зі специфікою майбутньої спеціальності, розширення й доповнення знань, здобутих студентами під час вивчення теоретичних курсів та виконання практичних і лабораторних завдань з дисциплін “Алгоритмізація та програмування”, “Проблемно-орієнтоване програмування”, “Основи інформаційних технологій”, “Системне програмування”, “Операційні системи”, “Теорія алгоритмів», “Схемотехніка комп’ютерів”, “Методи та засоби КІТ”, “Архітектура комп’ютерів”, “Комп’ютерні мережі”, “Чисельні методи в інформатиці”, “Моделювання систем”, “Організація баз даних та знань”, “Системи штучного інтелекту” та інших, а також формування професійних умінь і навичок для прийняття самостійних рішень і розв’язування задач пов’язаних із сучасними комп’ютерними інформаційними технологіями в реальних науково-дослідницьких або виробничих умовах, виховання потреби систематично оновлювати свої знання та творчо застосовувати їх у практичній діяльності.

Основними завданнями переддипломної практики є:

* провести дослідження на тему власного диплому «Автоматизована система оброблення резюме кандидатів на посаду в ІТ-компанію»;
* спроектувати реалізацію автоматизованої системи оброблення резюме кандидатів на посаду в ІТ-компанію;
* ознайомлення з нормативно-технологічною документацією підприємства – бази практики, вивчення особливостей адміністративної та інформаційної взаємодії між його окремими підрозділами;
* ознайомлення з конфігурацією та архітектурою технічних засобів підприємства;
* ознайомлення з програмним забезпеченням і програмними продуктами, які використовуються або створюються в процесі діяльності даного підприємства;
* ознайомлення з технологічними процесами керування інформаційними потоками, організацією зберігання даних, створення баз даних і доступу до даних;
* вивчення послідовності, методів і засобів оброблення даних;
* опанування технологіями розроблення, впровадження та супроводу програмного забезпечення;
* вивчення основ конфігурування технічних засобів, інформаційних систем та мереж, систем автоматизованого проектування та керування;
* оволодіння сучасними технологіями оброблення інформації та застосування їх для виконання практичних завдань;
* набуття виробничих навичок, а саме: технічної, проектувальної, виконавської;
* набуття навичок групової роботи та роботи в колективі спеціалістів;
* підготовка до вивчення спеціальних дисциплін на старших курсах.

Під час переддиплоиної практики було розроблено макет програми для менеджменту робочого процесу компанії, організації, підприємства, тощо. За основу бралися певні проблеми в організації цього процесу та втрати данних про членів Молодіжної громадської організації «Рада студентів технічних університетів».

На сьогоднішній день існує багато таких систем, з великим функціоналом, але в основному такі сервіси є платними (по піписці на кожного користувача), або ж вони не наповнені достатнім функціоналом. Тому було обрано створити власний програмний продукт для таких цілей.

Все більше різноманітних компаній та організацій почитають працювати по аналогічних системах. Та й не тільки в ІТ-компаніях. Просто зараз такі технології є дуже популярними саме серед них, оскільки це є їх технологія, але все більше і більше компаній починають впроваджувати певну автоматизації в свою повсякденну діяльність, оскільки це є набагато дешевше та зрічніше.

Звичайно такі системи вже є реалізовані,проте шлях в якому ми їх просто купуємо доволі кошторисний. Тому більшість людей в таких випадках використовують два варіанти:

Перший варіант заснований на тому, щоб знайти сервіси по наданю послуг - JIRA, Harvest і тому подібне. Але такі сервіси є доволі дорогими (10$ на місяць з людини). Тому для багатьох компаній такий варіант відпадає одразу.

Іншим варіантом є створення власного програмного забезпечення. Такий варіант і було обрано компанією, в якій я проходив практику. Крім витрачання меншої кількості грошей, однією з головних переваг є можливість модифікування цього програмного забезпечення власноруч. Якщо необхідно додати якийсь функціонал, характерний лише для вашого підприємства, а платні сервіси такого функціоналу не мають, створення власного програмного забезпечення це єдиний вихід.

Під час виконання переддипломної практики закріплено теоретичні знання та практичні навички розробки програмного забезпечення в невеликій команді розробників.

Для виконання роботи було сформовано команду, проаналізовано вже існуючі програмні продукти в даній області, обрано модель архітектури додатку клієнт-сервер та розподілено роботу між учасниками команди.

**Метапереддипломної практики**

Метою даної переддипломної практики (проекту) являється створення макету програмного продукту, пов’язаного з темою бакалаврської роботи – «Розробка застосунку для менеджменту завдань».

**Перша частина**

**Опис бази практики**

Базою практики було обрано та одобрено Молодіжної громадської організації «Рада студентів технічних університетів»(далі BEST).

[**BEST**](http://best.eu.org/) (Board of European Students of Technology) або Рада студентів технічних університетів – неприбуткова, неполітична та нерелігійна організація, яка представлена у 94 осередках 32 країн світу. З 1989 року BEST забезпечує обмін знаннями та співробітництво між студентами, компаніями та університетами на теренах Європи.

**Візія: Сила в різноманітті**

Люди розуміють та поважають різні культури та суспільства. Середовище, де сила в різноманітності, підтримує людей у реалізації їх потенціалу та відповідальній діяльності.

**Місія: Розвиток студентів**

BEST допомагає студентам досягти міжнародності мислення, кращого розуміння культур та народів, а також розвинути навички роботи в міжнародному середовищі.

**BEST Spirit**

Це те, завдяки чому ми залишаємося змотивованими, те, що підштовхує нас постійно рухатися далі. BEST Spirit – це натхнення у всьому, що ми робимо. Ось основні **5 цінностей**, які визначають BEST Spirit: **Flexibility** (Гнучкість), **Friendship** (Дружба), **Fun** (Фан), **Improvement** (Розвиток), **Learning** (Навчання).

**Локальний осередок BEST Lviv** існує з 2002 року. Діяльність львівського осередку забезпечується її Виконавчою радою та декількома робочими групами. Окрім Ради, у осередку налічується приблизно 70 активних членів організації. Також варто додати, що BEST Lviv співпрацює з більше ніж 50 українськими та міжнародними компаніями та працює для 31 000 студентів Львівської Політехніки.

**Освоєні інформаційні технології**

В проекті я використовував мову програмування Java включаючи такі технології як Java Beans, AWT, Swing, Spring, Aspectj та Hibarnate.

**Друга частина**

***Аналіз вимог до програмного продукту***

Принципи побудови програмного продукту

Необхідно створити додаток, за допомогою якого користувач зможе здійснити автоматизований процес видачі та виконання завдань, та берігання інформації про членів компанії-органіації. Багатосторінкова та багаторівнева структура системи повинна забезпечити зручний інтерфейс взаємодії з користувачем. Активація різного функціоналу здійснюється при активації графічних елементів (кнопок). Принцип роботи програми ви можете побачити на Рис 1.



*Рис. 1. Схема роботи системи в загальному вигляді.*

Протокол верифікації

|  |  |  |
| --- | --- | --- |
| **Вимога** | **Програмний продукт** | **Оцінка** |
| Швидкодія ПП = 8 сек | Час відклику на запит в середньому триває до 5 сек | виконано |
| Кількість одночасних доступів до сервера ≥ … | В залежності від сервера | виконано |
| Вимоги до оперативної пам’яті не більше 40 Мб | Використовує до 20 МБ | виконано |

*Вимоги до функціональних характеристик*

Здатність видавати потрібні результати. Здатність запобігати неавторизованому доступу до даних та забезпечувати цілісність і конфіденційність даних.

Програмний продукт повинен забезпечити можливість перегляду актуальної інформації, можливість надсилання і прийому завдань та оформлення і редагування завдання. Також потрібна можливість створювати нові компанії та заповнювати інформацію про користувачів.

*Функції зареєстрованого користувача:*

- вхід в систему за допомогою електронної адреси;

- заповнення змісту завдання;

- завантаження завдання на сервер та виведення відповідальній особі;

- вікно, яка повідомляє про успішно виконані завдання.

*Вимоги до інтерфейсу програмного продукту*

Програмний продукт має бути зручним у використанні. Повинна бути передбачена зрозуміла навігація по сторінках додатку. Графічні елементи повинні бути відповідним чином оптимізовані з метою мінімізації часу завантаження на комп'ютері користувача. Керування додатком має бути інтуїтивним, щоб користувач легко орієнтувався у структурі додатку і довго не шукав потрібну йому кнопку.

***Вимоги до ергономічних характеристик***

Необхідними є умови максимально швидкого оновлення даних, точності та актуальності інформації. Зміни внесені користувачами повинні одразу ж відображатися на екрані.

***Вимоги до алгоритму роботи і структури програми***

Система працює цілодобово. Користувач відкриває додаток, авторизується або реєструється і отримує доступ до всіх функцій, наданих йому відповідно до його повноважень.

Структура системи має ряд основних функцій:

- авторизація користувача;

- реєстрація користувача;

- внесення персональної інформації про користувача;

- відображення списку інших працівників компанії-організації;

- відображення списку завдань;

- можливість надіслання та отримування листів між користувачами;

- можливість створення оголошень-розсилок в межах компанії-організації;

- створення голосувань для прийняття рішень;

- редагування інформації про людей та компанію;

Програмний продукт повинен бути швидкодіючим, тому потрібно використовувати ефективні та оптимальні алгоритми.

Структура програми не повинна бути складною, щоб легко можна було вносити зміни, додавати новий функціонал та покращувати програму.

***Вимоги до надійності***

У програмі повинен відбуватися контроль та перевірка коректності даних, які вводить користувач. Також повинна здійснюватись перевірка інформації, які вводить користувач, для виявлення невалідних даних та недопущення їх відправки кандидатам.

Забезпечити надійність зберігання та унеможливити витік персональної інформації про користувачів.

***Умови функціонування (експлуатації) програми***

Додаток повинен коректно працювати на всіх браузерах з наявним підключенням до інтернету, а також працювати безперебійно при великій кількості користувачів та зберіганні одночасно багатьох змін.

За більш детальним обслуговуванням, програмними змінами та оновленням системи можна буде звернутись до розробників програмного продукту.

***Вимоги до складу і параметрів технічних засобів***

Користування додатком можливе з усіх операційних систем, що мають доступ до Інтернету. Особливих вимог до технічних характеристик пристроїв не висувається.

**Модуль авторизації користувача**

Авторизація - це процедура підтвердження прав на додаткові можливості в залежності від того, який рівень доступу має користувач, що проходить авторизацію. Для користувачів ці права отримати нескладно - достатньо пройти реєстрацію та мати доступ на дану електронну пошту.

Авторизація дозволяє ідентифікувати користувачів додатку і одночасно обмежити права доступу неавторизованим користувачам. Також після авторизації вказане при реєстрації ім’я відображається разом з коментарем.

Авторизація може частково захистити власника ресурсу та його відвідувачів від спаму. Його розсилають спеціальні програми - спам - боти, які автоматично заповнюють коментарями з рекламою та посиланнями будь-які місця, де можливо додати коментар. Захиститись від них можна різними способами, і авторизація є найбільш ефективним з них.

Інша, не менш важлива, мета авторизації - це отримання необхідної інформації про людину, її ідентифікація. Цю інформацію користувач надає добровільно, і використовується вона виключно в цілях розпізнавання іншими відвідувачами.

Переваги авторизації для користувача:

1) Можливість захистити себе від спаму. На ресурсах, де є авторизація, куди менше спам-роботів і людей, що заробляють на спам та рекламі.

Модуль авторизації має два поля: «Авторизуватись через логін» та «Введіть пароль».

Дані з форми авторизації перевіряються з існуючою базою даних і при відсутності користувача виводиться повідомлення про помилку.

Також користувач може зрозуміти, що на даному додатку він ще не зареєстрований і перейти до сторінки реєстрації.

**Мета тестування**

Метою тестування даного проекту є перевірка усіх функціональних можливостей програмного продукту «Розробка застосунку для менеджменту завдань», а також коректність роботи програмного продукту на різних браузерах та при використанні браузерів з різним розширенням.

**План процесу тестування**

Тестування проводитиметься в три рівні:

1) Модульне тестування полягає в перевірці працездатності та коректності роботи модулів програми окремо один від одного.

2) Інтеграційне тестування проводиться до тих пір, поки тестовані на попередньому кроці окремі модулі програми, не почнуть працювати як система.

3) На останньому рівні системного тестування інтегрована система перевіряється на відповідність всім вимогам. Крім того, системне тестування ПЗ повинно гарантувати, що програма працює так, як очікувалося, а також, що її не можна знищити чи перевести систему в неробочий стан. Системне інтеграційне тестування перевіряє, чи система інтегрується в будь-яку зовнішню систему .

Функції, які повинен виконувати даний програмний продукт:

- вхід в систему за допомогою логіна і пароля;

- реєстрація в систему;

- можливість редагувати;

- можливість надавати умови доступу іншим користувачам;

- можливість менеджити видані завдання;

- можливість налаштовувати конфігурацію;

- перегляд стану виконання;

Всі виявлені невідповідності та дефекти прохання надсилати розробникам проекту на електронну пошту.

**План-графік робіт тестування**

|  |  |  |  |
| --- | --- | --- | --- |
| **Завдання** | **Час виконання** | **Дата початку** | **Дата кінця** |
| Створення тест плану | 8 год. | 25.05.2018 | 25.05.2018 |
| Модульне тестування | 12 год. | 26.07.2018 | 27.05.2018 |
| Інтеграційне тестування | 12 год. | 27.05.2018 | 02.05.2018 |
| Системне тестування | 16 год. | 03.05.2018 | 04.05.2018 |

**Індивідуальна частина**

Я був головним розробником даної системи, звичайно ще був замовник (голова організації) та інші, більш досвідченні члени ІТ департаменту організації, у яких я міг спитатись поради. Все що стосувалось розробки було спроектовано на написано мною, всі дизайнерські рішення та їх реалізація також лежала на мені. Мною була реалізований з нуля застосунок для менеджменту завдань.

Крім написання коду я також приймав участь у розробці архітектури додатка та оцінці розробки проекта, тобто детальному описанні всього функціоналі та опису скільки часу займе реалізація того чи іншого модуля програми. Має вказуватись нижня та верхня границя а також найбільш ймовірну кількість годин, яку треба для реалізації окремою функції.

Я використовував систему контролю версій git. Git є однією з найефективніших, надійних і високопродуктивних [систем керування версіями](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D1%96%D1%8F%D0%BC%D0%B8), що надає гнучкі засоби нелінійної розробки, що базуються на відгалуженні і злитті гілок. Для роботи з нею я використовув git bash. Хоча на даний момент з’явилося достатньо інструментів для git з графічним інтерфейсом і саме середовище розробки містить інструменти для роботи зним, всеодно, на мій погляд, найкращим інструментом залишається консольна утиліта. Спільний репозиторій ствонено на GitHub.

**Висновок**

Виконуючи переддипломну практику нами було здобуто вміння працювати в команді та розподіляти обов’язки. Усі стадії проекту є важливими, тому ми допомагали один одному та вирішували ключові завдання разом.

Результатом нашої роботи є «Автоматизована система оброблення резюме кандидатів на посаду в ІТ-компанію» - додаток, що дозволяє частково автоматизувати роботу підприємства у сфері прийому на роботу.

Під час виконання переддипломної практики було розроблено проектну документацію та план тестування програмного продукту.

Перед тим, як прийнятися за проектування та розробку додатку, було проведено аналіз вже існуючих програмних рішень в цій сфері, з метою визначити доцільність даної системи.

**Використана література**

1. [*Кеті Сіерра*](https://www.ozon.ru/person/2164439/)*,*[*Берт Бейтс*](https://www.ozon.ru/person/2164441/)*.* Head First Java — Indianapolis, IN 46256: Wiley Publishing, Inc., 20012. — P. 12 — 13. — (Programmer's Reference). — [ISBN 9781306813204](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8_%D0%BA%D0%BD%D0%B8%D0%B3/9780470344729) .
2. *Cay S. Horstmann.*  Core Java Volume I-Fundamentals — 11st ed. — 1005 Gravenstein Highway North, Sebastopol, CA 95472: [O’Reilly Media](https://ru.wikipedia.org/wiki/O%E2%80%99Reilly_Media), 2010. — P. xiii. — [ISBN 9780137082247](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8_%D0%BA%D0%BD%D0%B8%D0%B3/9780596802790) .
3. *Herbert Schildt.* The Complete Reference. — Ninth Edition. — O’Reilly Media, 2014. — P. 41. — [ISBN 978-5-8459-1955-7](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8_%D0%BA%D0%BD%D0%B8%D0%B3/9781449365035) .

**Терміни**

*Проект (програмний продукт)* – сайт «Інтернет аукціон і магазин речей», що дозволяє робити користувачам ставки на товари та покупку товарів на сайті.

*Тестування* — процес контролю якості, що перевіряє відповідність між реальною і очікуваною поведінкою програмного забезпечення, тестування функцій програми на відповідність вимогам.

*Модульне тестування* - відноситься до тестів, які перевіряють функціональність певного розділу коду, зазвичай на функціональному рівні. Зазвичай виконується самим програмістом. Програму, що не пройшла такий тест, немає сенсу передавати на глибше тестування.

*Інтеграційне тестування* - виявлення дефектів у інтерфейсах та перевірка взаємодії інтегрованих компонентів (модулів).

### *Системне тестування* - тестує інтегровану систему для перевірки відповідності всім вимогам. Крім того, системне тестування ПЗ повинно гарантувати, що програма працює так, як очікувалося. Системне інтеграційне тестування перевіряє, чи система інтегрується в будь-яку зовнішню систему (або системи) відповідно до системних вимог.

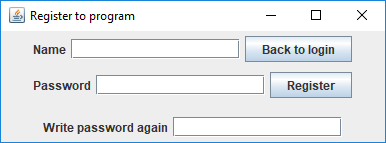
*Технічне завдання* - документ, що описує набір технічних і функціональних вимог до програмного продукту .

**Додаток А**

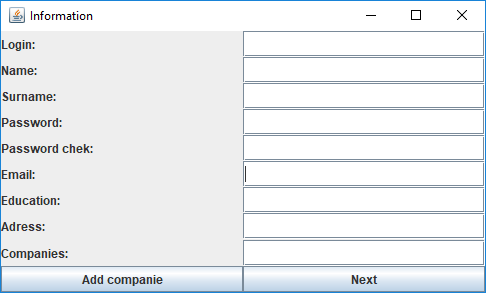
**Зображення додатку**



*Рис.1 Вікно авторизації*



*Рис.2 Вікно реєстрації*



*Рис.3 Вікно заповнення інформації*

**Код програми**

**package** Diplom.Diplom;

**import** java.awt.BorderLayout;

**import** java.awt.GridLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.awt.event.WindowEvent;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JMenu;

**import** javax.swing.JMenuBar;

**import** javax.swing.JMenuItem;

**import** javax.swing.JPanel;

**import** javax.swing.JPasswordField;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTabbedPane;

**import** javax.swing.JTextArea;

**import** javax.swing.JTextField;

**public** **class** Frame {

**void** login()

{

**final** JFrame loginframe = **new** JFrame("Login to program");

loginframe.setSize(400,120);

loginframe.setLocationRelativeTo(**null**);

loginframe.setDefaultCloseOperation(JFrame.***HIDE\_ON\_CLOSE***);

JButton loginbutton = **new** JButton("Login");

JButton regbutton = **new** JButton("Register");

**final** JTextField namefield = **new** JTextField(15);

JPasswordField passfield = **new** JPasswordField(15);

JLabel namelbl = **new** JLabel("Name");

JLabel passlbl = **new** JLabel("Password");

JPanel uptpanel = **new** JPanel();

JPanel downpanel = **new** JPanel();

uptpanel.add(namelbl, BorderLayout.***WEST***);

downpanel.add(passlbl, BorderLayout.***WEST***);

uptpanel.add(namefield, BorderLayout.***CENTER***);

downpanel.add(passfield, BorderLayout.***CENTER***);

uptpanel.add(loginbutton, BorderLayout.***EAST***);

downpanel.add(regbutton, BorderLayout.***EAST***);

loginframe.add(uptpanel, BorderLayout.***NORTH***);

loginframe.add(downpanel, BorderLayout.***CENTER***);

loginframe.setVisible(**true**);

regbutton.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

register();

loginframe.dispatchEvent(**new** WindowEvent(loginframe, WindowEvent.***WINDOW\_CLOSING***));

}

});

loginbutton.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

workspase(namefield.getText());

loginframe.dispatchEvent(**new** WindowEvent(loginframe, WindowEvent.***WINDOW\_CLOSING***));

}

});

}

**void** register()

{

**final** JFrame registerframe = **new** JFrame("Register to program");

registerframe.setSize(400,150);

registerframe.setLocationRelativeTo(**null**);

registerframe.setDefaultCloseOperation(JFrame.***HIDE\_ON\_CLOSE***);

JButton loginbutton = **new** JButton("Back to login");

JButton regbutton = **new** JButton("Register");

**final** JTextField namefield = **new** JTextField(15);

JPasswordField passfield = **new** JPasswordField(15);

JPasswordField repassfield = **new** JPasswordField(15);

JLabel namelbl = **new** JLabel("Name");

JLabel passlbl = **new** JLabel("Password");

JLabel repasslbl = **new** JLabel("Write password again");

JPanel uptpanel = **new** JPanel();

JPanel midpanel = **new** JPanel();

JPanel downpanel = **new** JPanel();

uptpanel.add(namelbl, BorderLayout.***WEST***);

midpanel.add(passlbl, BorderLayout.***WEST***);

downpanel.add(repasslbl, BorderLayout.***WEST***);

uptpanel.add(namefield, BorderLayout.***CENTER***);

midpanel.add(passfield, BorderLayout.***CENTER***);

downpanel.add(repassfield, BorderLayout.***CENTER***);

uptpanel.add(loginbutton, BorderLayout.***EAST***);

midpanel.add(regbutton, BorderLayout.***EAST***);

registerframe.add(uptpanel, BorderLayout.***NORTH***);

registerframe.add(midpanel, BorderLayout.***CENTER***);

registerframe.add(downpanel, BorderLayout.***SOUTH***);

registerframe.setVisible(**true**);

regbutton.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

userinformation();

registerframe.dispatchEvent(**new** WindowEvent(registerframe, WindowEvent.***WINDOW\_CLOSING***));

}

});

loginbutton.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

login();

registerframe.dispatchEvent(**new** WindowEvent(registerframe, WindowEvent.***WINDOW\_CLOSING***));

}

});

}

**void** workspase(String name) {

JFrame frame = **new** JFrame(name);

frame.setSize(800, 500);

frame.setVisible(**true**);

frame.setLocationRelativeTo(**null**);

frame.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

frame.setExtendedState(JFrame.***MAXIMIZED\_BOTH***);

JTabbedPane tabby = **new** JTabbedPane();

JPanel panel1= **new** JPanel();

JPanel panel2 = **new** JPanel();

//Creating Menu

JMenuBar menuBar = **new** JMenuBar();

JMenu fileMenu = **new** JMenu("lalala");

JMenuItem newFile = **new** JMenuItem("lalala");

JMenuItem openFile = **new** JMenuItem("lalala");

JMenuItem saveFile = **new** JMenuItem("lalala");

JMenuItem saveAsFile = **new** JMenuItem("lalala");

JMenuItem exitFile = **new** JMenuItem("lalala");

fileMenu.add(newFile);

fileMenu.add(openFile);

fileMenu.addSeparator();

fileMenu.add(saveFile);

fileMenu.add(saveAsFile);

fileMenu.addSeparator();

fileMenu.add(exitFile);

JMenu editMenu = **new** JMenu("lalala");

JMenuItem canselEdit = **new** JMenuItem("lalala");

JMenuItem cutEdit = **new** JMenuItem("lalala");

JMenuItem copyEdit = **new** JMenuItem("lalala");

JMenuItem pasteEdit = **new** JMenuItem("lalala");

JMenuItem deleteEdit = **new** JMenuItem("lalala");

JMenuItem findEdit = **new** JMenuItem("lalala");

JMenuItem replaceEdit = **new** JMenuItem("lalala");

editMenu.add(canselEdit);

editMenu.addSeparator();

editMenu.add(cutEdit);

editMenu.add(copyEdit);

editMenu.add(pasteEdit);

editMenu.add(deleteEdit);

editMenu.addSeparator();

editMenu.add(findEdit);

editMenu.add(replaceEdit);

JMenu analysMenu = **new** JMenu("lalala");

menuBar.add(fileMenu);

menuBar.add(editMenu);

menuBar.add(analysMenu);

tabby.addTab("First", panel1);

tabby.addTab("Second", panel2);

frame.setJMenuBar(menuBar);

frame.add(tabby);

}

**void** userinformation() {

JFrame frame = **new** JFrame("Information");

frame.setSize(500, 300);

frame.setVisible(**true**);

frame.setLocationRelativeTo(**null**);

frame.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

JLabel loginlbl = **new** JLabel("Login:");

JLabel namelbl = **new** JLabel("Name:");

JLabel surnamelbl = **new** JLabel("Surname:");

JLabel passlbl = **new** JLabel("Password:");

JLabel repasslbl = **new** JLabel("Password chek:");

JLabel maillbl = **new** JLabel("Email:");

JLabel educationlbl = **new** JLabel("Education:");

JLabel adresslbl = **new** JLabel("Adress:");

JLabel copmpanieslbl = **new** JLabel("Companies:");

JTextField loginfld = **new** JTextField(15);

JTextField namefld = **new** JTextField(15);

JTextField surnamefld = **new** JTextField(15);

JTextField passfld = **new** JTextField(15);

JTextField repassfld = **new** JTextField(15);

JTextField mailfld = **new** JTextField(15);

JTextField educationfld = **new** JTextField(15);

JTextField adressfld = **new** JTextField(15);

JTextField companiesfld = **new** JTextField(15);

JButton compbut = **new** JButton("Add companie");

JButton nextbut = **new** JButton("Next");

JPanel pan = **new** JPanel(**new** GridLayout(0, 2) );

JPanel dowpan = **new** JPanel(**new** GridLayout(2, 2) );

frame.add(pan, BorderLayout.***CENTER***);

frame.add(dowpan, BorderLayout.***SOUTH***);

pan.add(loginlbl);

pan.add(loginfld);

pan.add(namelbl);

pan.add(namefld);

pan.add(surnamelbl);

pan.add(surnamefld);

pan.add(passlbl);

pan.add(passfld);

pan.add(repasslbl);

pan.add(repassfld);

pan.add(maillbl);

pan.add(mailfld);

pan.add(educationlbl);

pan.add(educationfld);

pan.add(adresslbl);

pan.add(adressfld);

dowpan.add(copmpanieslbl);

dowpan.add(companiesfld);

dowpan.add(compbut);

dowpan.add(nextbut);

}

}