РОЗРОБКА ЗАСОБІВ ДЛЯ ТЕСТУВАННЯ КЛІЄНТ-СЕРВЕРНИХ СИСТЕМ

Ткач Є. С., гр.3КСУА-14сп

Вінниця ВНТУ 2015

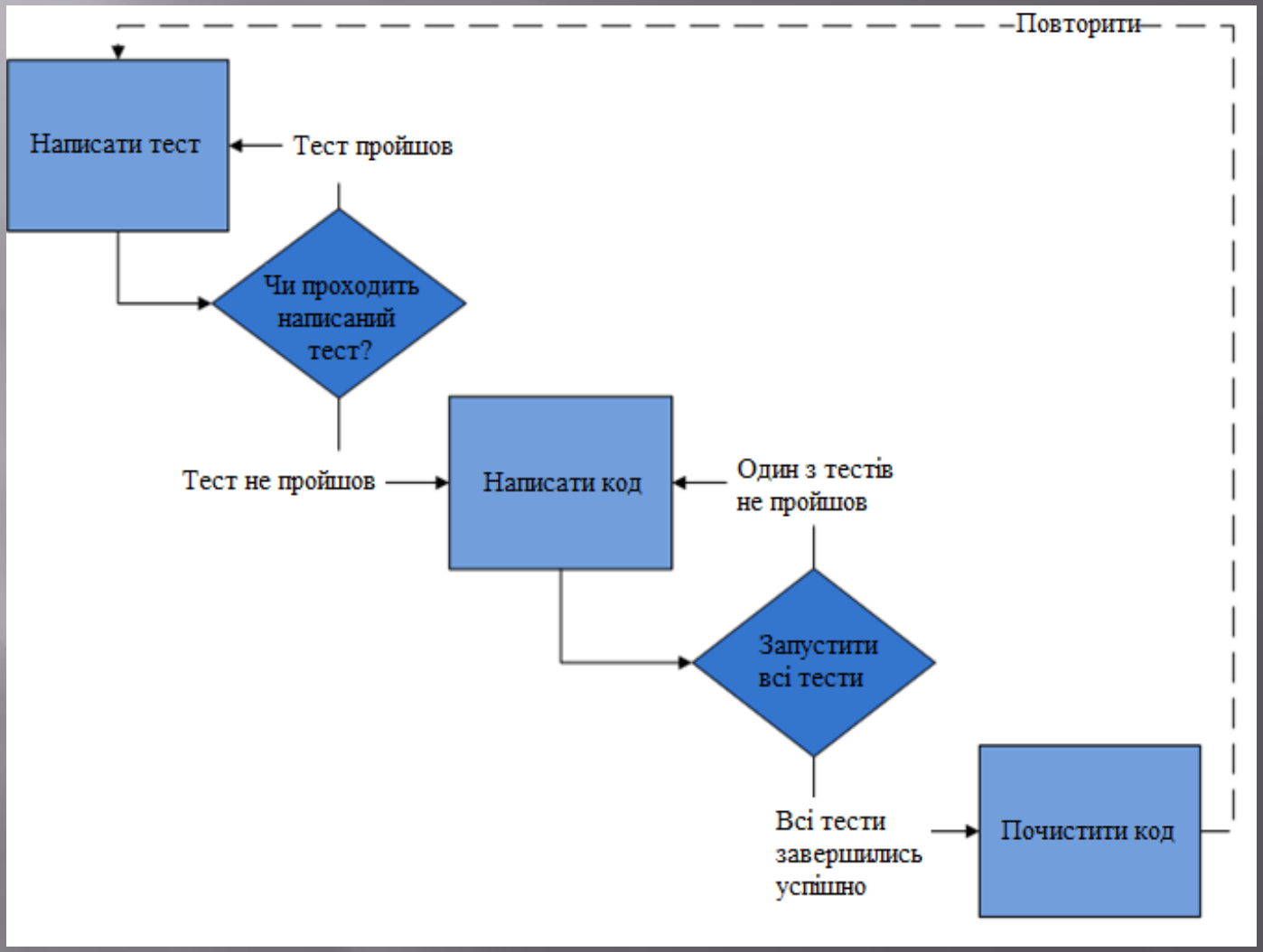
Актуальність проблеми. Загальний погляд на тестування програмного забезпечення останні роки активно еволюціонував, стаючи все більш конструктивним, прагматичним і наближеним до реалій сучасних проектів розробки програмних систем. Сьогодні тестування розглядається як діяльність, яку необхідно проводити протягом усього процесу розробки і супроводу. Воно є важливою частиною конструювання програмних продуктів. Дійсно, планування тестування повинно починатися на ранніх стадіях роботи, необхідно систематично і постійно розвивати і уточнювати плани тестів і відповідні процедури тестування. Навіть самі по собі сценарії тестування виявляються дуже корисними для тих, хто займається проектуванням, дозволяючи виділяти ті аспекти вимог, які можуть неоднозначно інтерпретуватися або навіть бути суперечливими. Але з іншого боку, необхідно усвідомлювати, що навіть якщо тести показали позитивні результати, це зовсім не означає, що отриманий продукт не містить помилок але дає впевненість, що він справляється зі своєю основною функціональністю.

Мета і задачі дослідження. Метою роботи є підвищення ефективності, стабільності, надійності розробки програмного забезпечення. Створення ефективних тестів, що забезпечать впевненість того, що програмний продукт справляється не тільки з його безпосередньою задачею, а і в різних ситуаціях виключеннях.

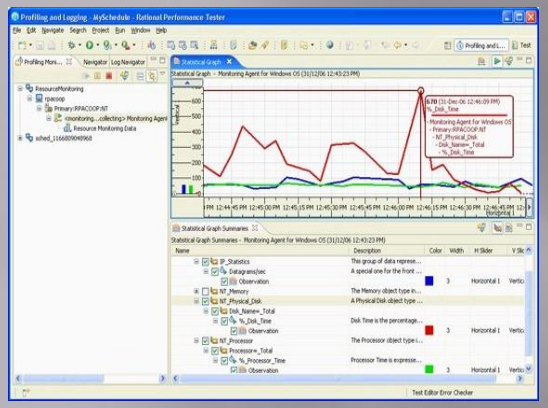
Об'єктом дослідження і розробки є процес створення програмного забезпечення, зокрема, етап тестуваннння.

Предметом дослідження є засоби автоматизації тестування клієнт - серверних систем.

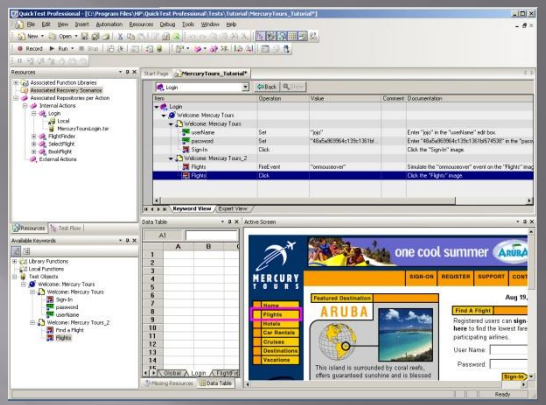
Розробка програми через тестування



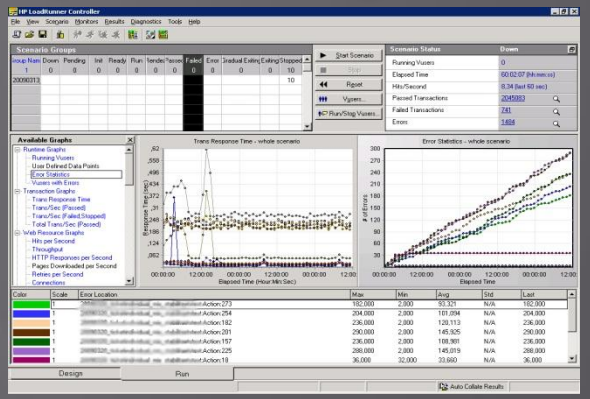
Канонічні схеми регулятора стану



IBM Rational Functional Tester

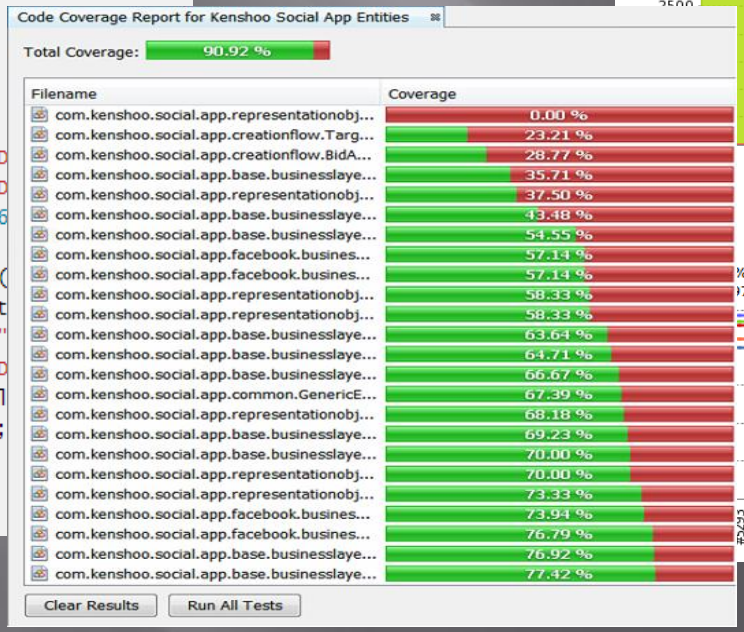


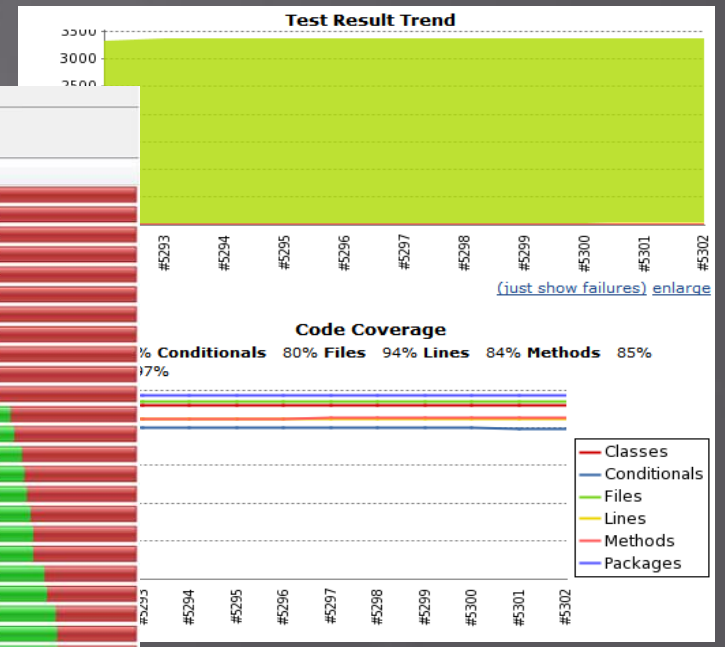
HP QuickTest Professional



HP LoadRunner

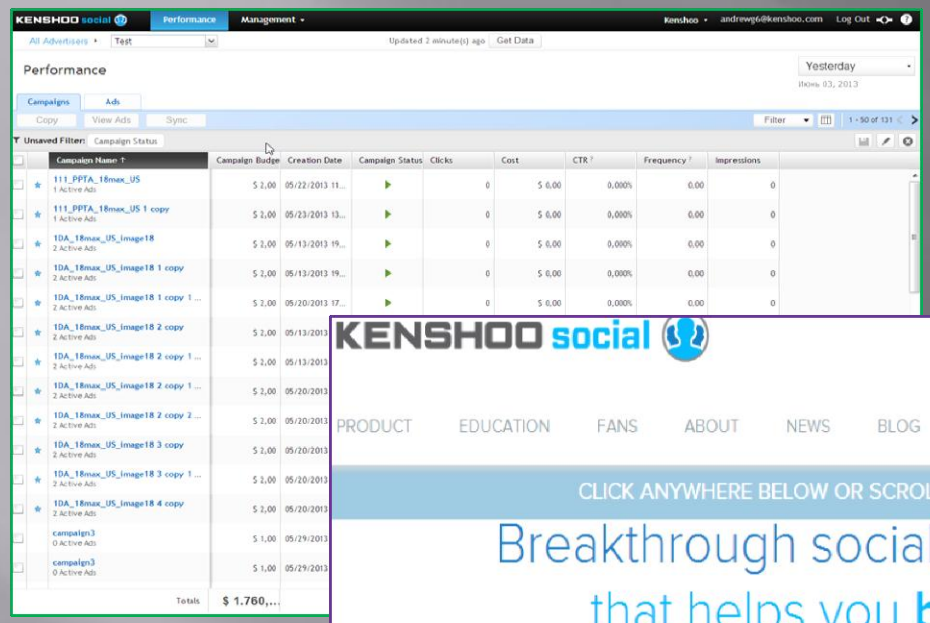
```
import static org.junit.Assert.*;
import org.junit.Test;

public class TestMoney {

    @Test
    public void testAdd() {
        Money m1 = new Money(12, "USD
        Money m2 = new Money(14, "USD
        Money expected = new Money(26
        Money result = m1.add(m2);
        assertFalse(!expected.equals(
        assertEquals(expected, result
        assertNotNull(new Money(10, "
        assertNull(new Money(10, "USD
        assertNotSame(expected, resul
        assertSame(expected, result);
    }

}
```

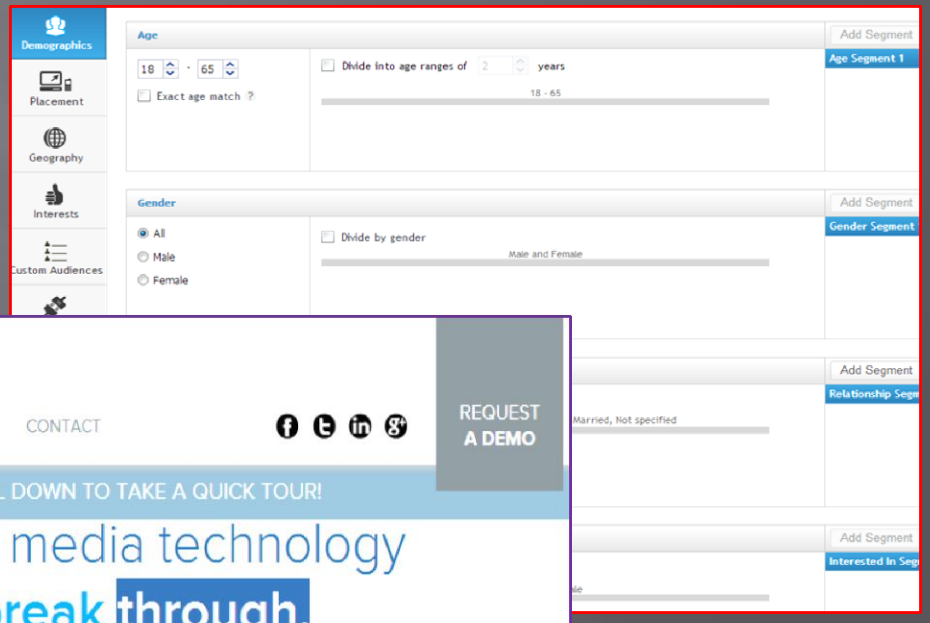


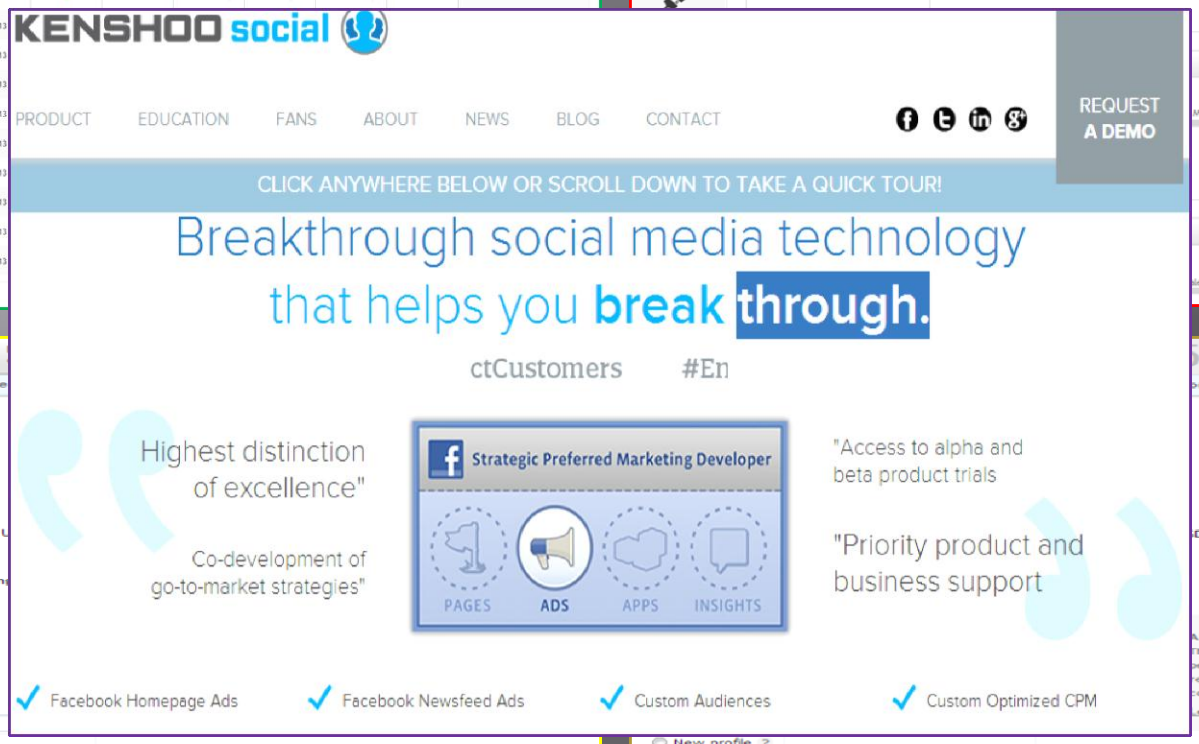


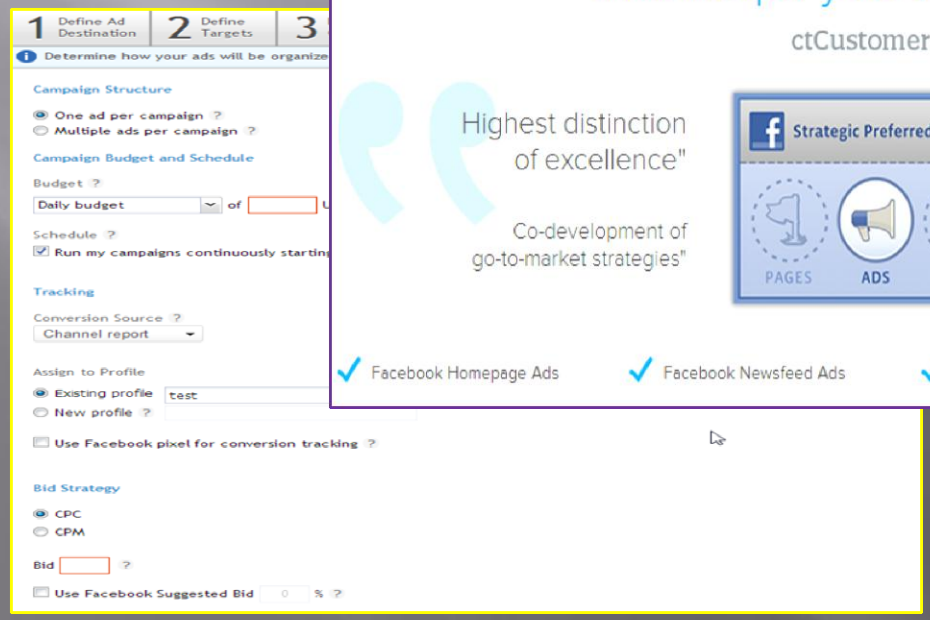
Junit тестування

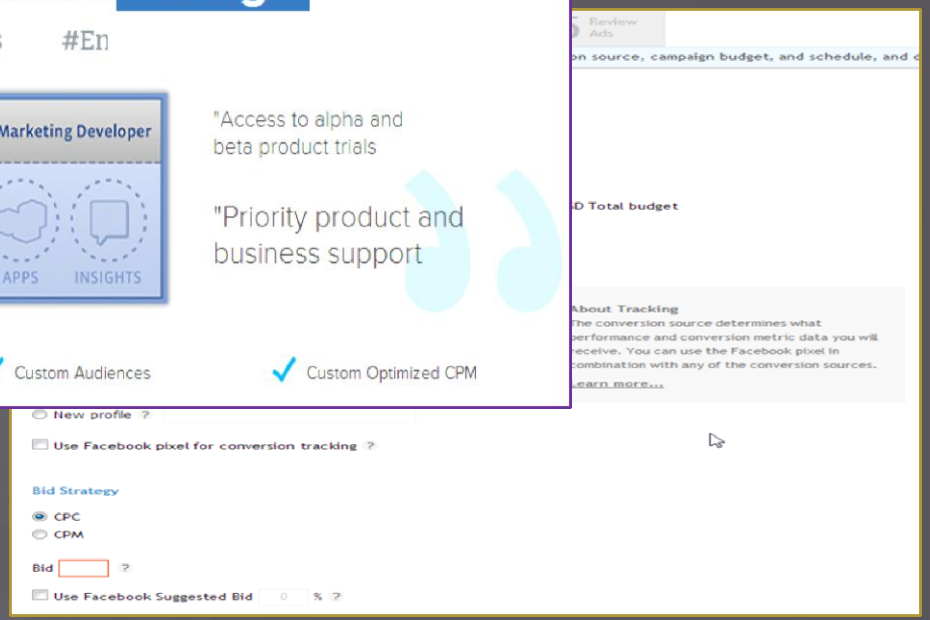
Клієнт-серверна система



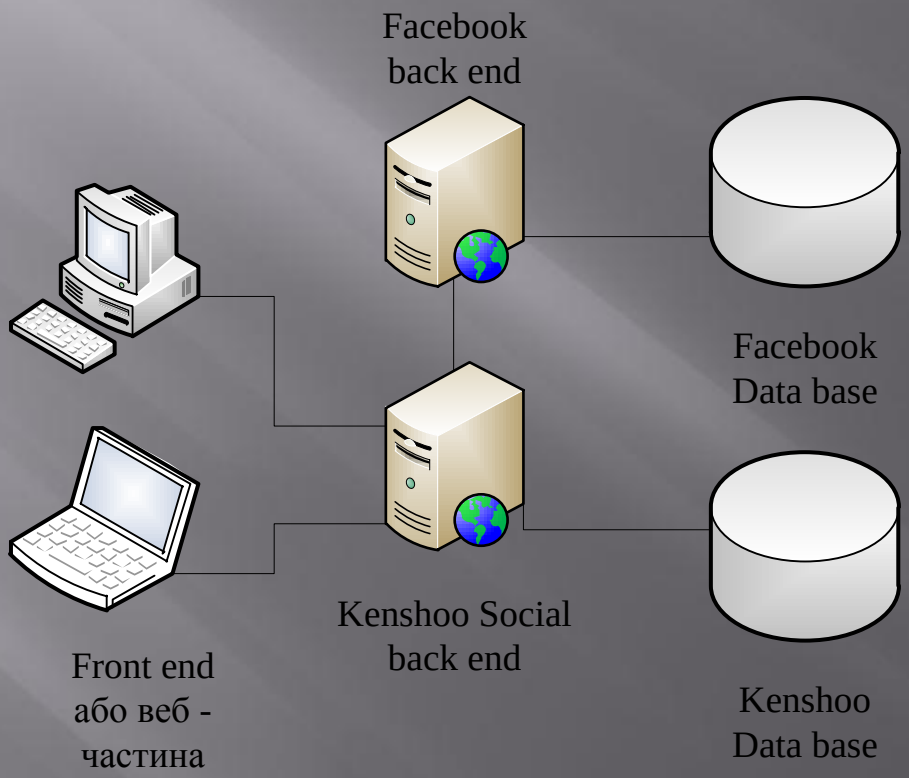




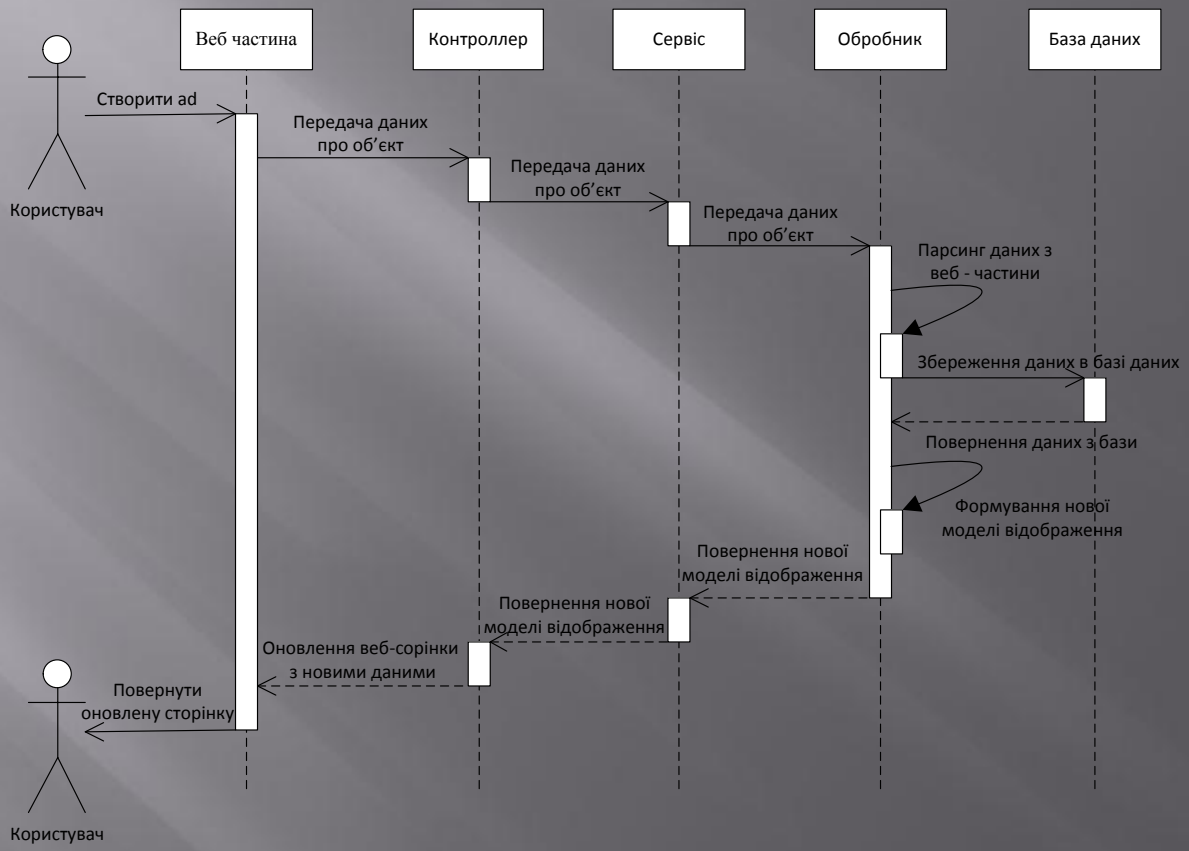




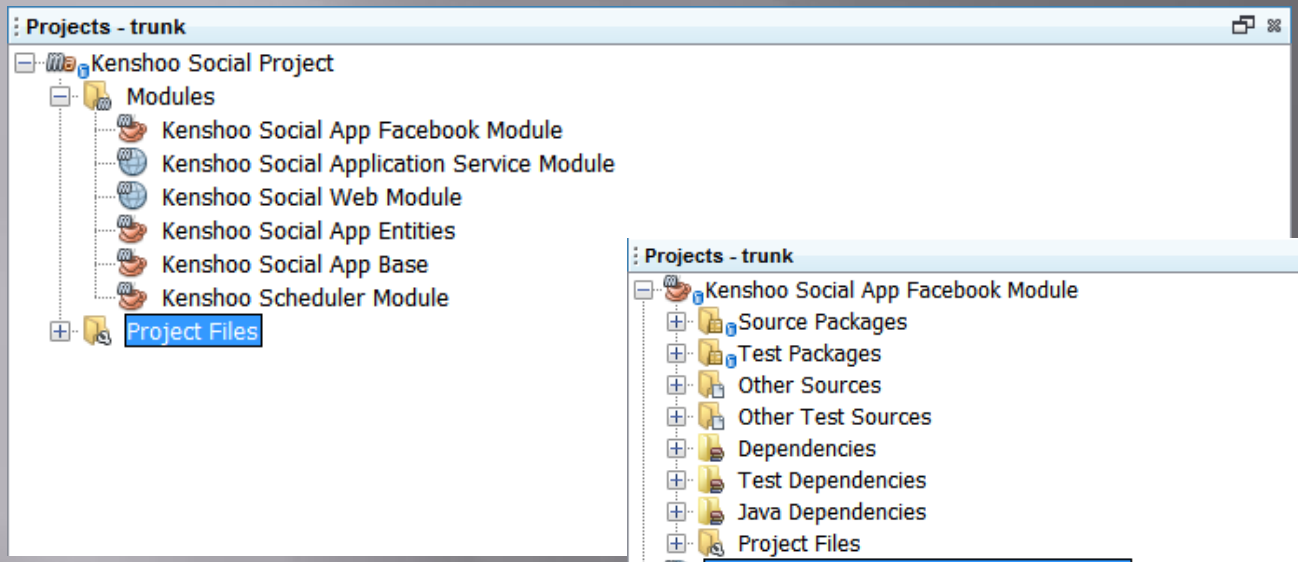
Структурна схема програмного забезпечення

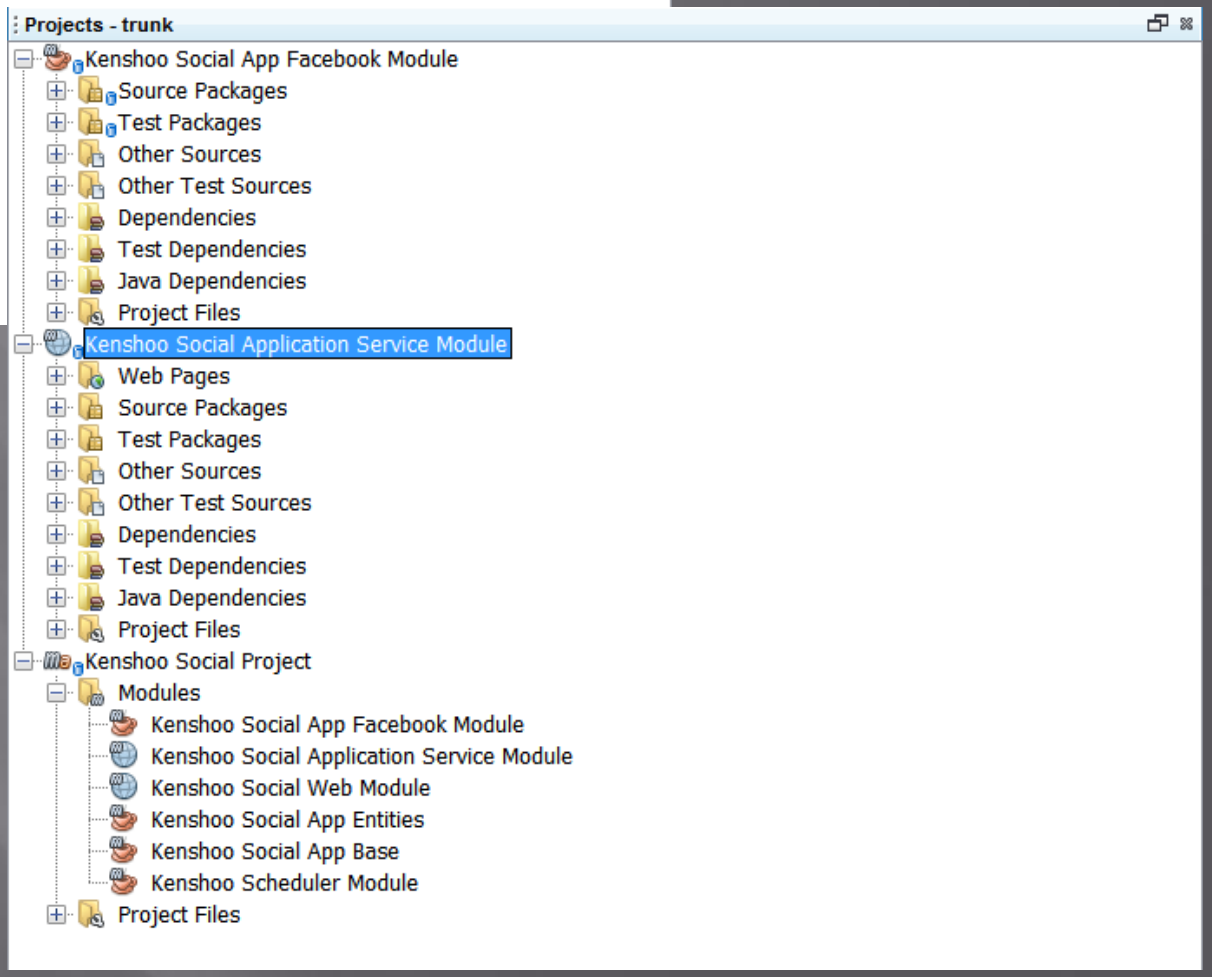


UML – діаграма діяльності



Структура ПЗ





Загальна структура тестового класу

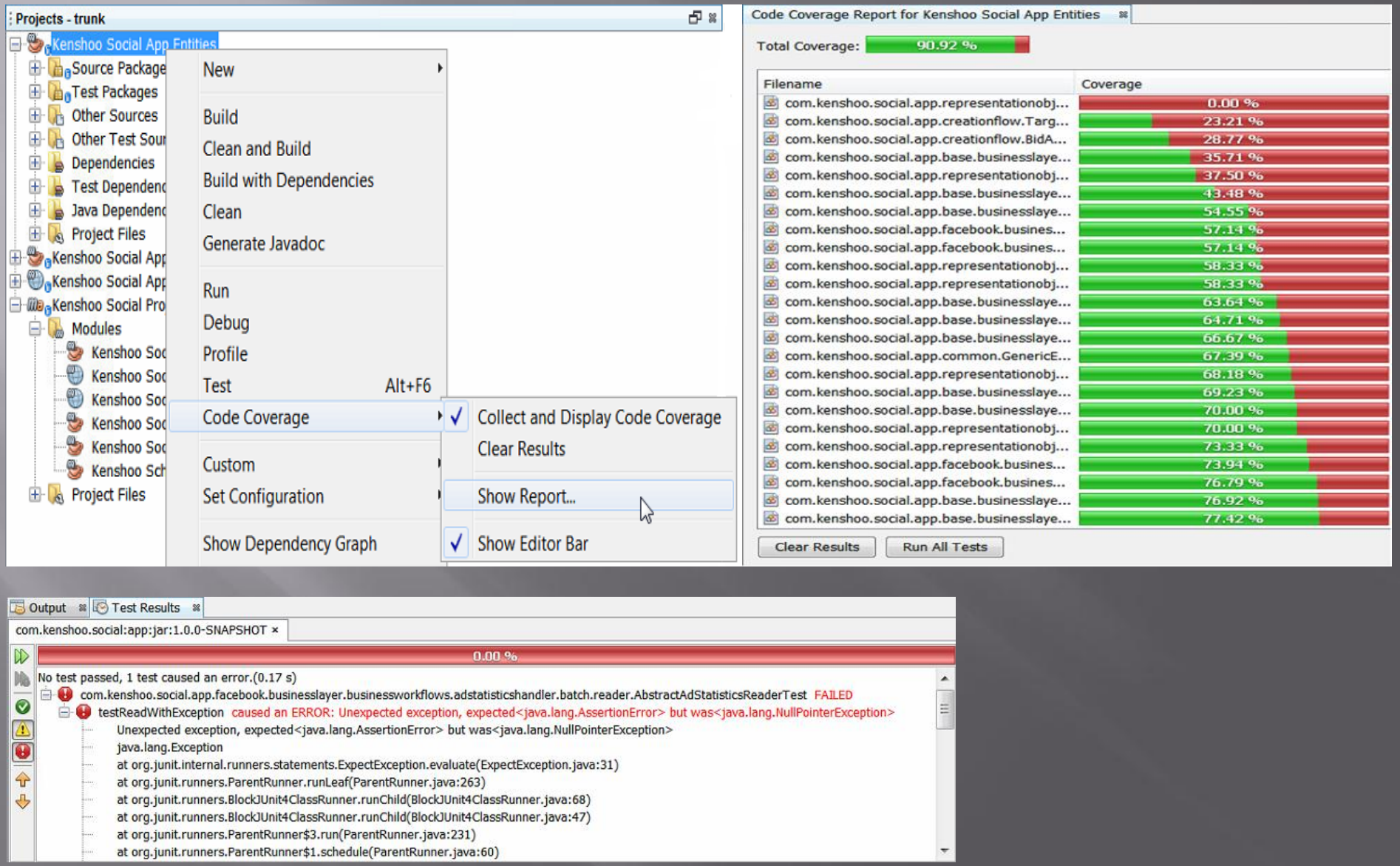
```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = {"classpath:/testContext.xml"})
@TransactionConfiguration
@Transactional(isolation = Isolation.READ_UNCOMMITTED)
```

```
@Test
public void testEditAd() {
    FacebookCampaignRO campaignRO = new FacebookCampaignRO(campaign);
    FacebookAdgroupRO adgroupRO = campaignRO.getFacebookAdgroupROs().get(0);
    FacebookCreativeRO creativeRO = adgroupRO.getFacebookCreativeROs().get(0);
    campaignRO.setId(campaign.getId());
    adgroupRO.setId(adgroup.getId());
    creativeRO.setId(creative.getId());
    Map<String, Object> editAdResult = adsController.editAd(new FacebookCampaignRO[]{campaignRO}, null);
    assertTrue((Boolean) editAdResult.get("success"));

    ReflectionTestUtils.setField(adsController, "adEditingService", null);
    editAdResult = adsController.editAd(new FacebookCampaignRO[]{campaignRO}, null);
    assertFalse((Boolean) editAdResult.get("success"));
}
```

Файл налаштування - testContext.xml

```
<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
    <property name="driverClassName" value="org.hsqldb.jdbcDriver"/>
    <property name="url" value="jdbc:hsqldb:mem:socialTest"/>
    <property name="username" value="sa"/>
    <property name="password" value=""/>
</bean>

<bean id="dataSourceInitializer" class="test.jdbc.datasource.InitializingDataSourceFactoryBean"
      autowire-candidate="false">
    <property name="dataSource" ref="dataSource"/>
    <property name="initScripts">
        <list>
            <value>schema-hsqldb.sql</value>
        </list>
    </property>
</bean>


<bean id="sessionFactory" class="org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean">
    <property name="dataSource" ref="dataSource"/>
    <property name="packagesToScan" value="com.kenshoo.social.app"/>
    <property name="namingStrategy" ref="namingStrategy"/>
    <property name="hibernateProperties">
        <value>
            hibernate.dialect=org.hibernate.dialect.HSQLDialect
            hibernate.format_sql=true
            hibernate.hbm2ddl.auto=create
            hibernate.show_sql=false
            use_sql_comments=true
            hibernate.cache.use_second_level_cache = false
        </value>
    </property>
```

Запуск тестових класів



Висновки

- В роботі проведено аналіз та розробку засобів тестування клієнт - серверних сисмтем.
- Дипломний проект складається з чотирьох розділів.
- В першому розділі подана загальна характеристика методів та засобів тестування клієнт - серверних систем. Виконано конструктивний аналіз існуючих методів, що використовуються для тестування програмного забезпечення різного рівня складності. Проведено техніко-економічне обґрунтування нової розробки. Спрощеним способом було проведено співставлення капітальних вкладень та експлуатаційних витрат нової розробки та аналога. При цьому доведено, що новий програмний продукт є економічно ефективним, оскільки забезпечує економію як на капітальних вкладеннях, так і на експлуатаційних витратах для споживача.
- В другому розділі визначено основні вимоги до клієнт-серверної системи та обрано засіб тестування, що найкращим чином підходить для нашої системи. На основі обраного засобу тестування розроблено тестові класи за допомогою яких тестується програмне забезпечення. Проведено короткий огляд клієнт-серверного проекту, засобів, за допомогою яких була реалізована така система, її загальну структуру, огляд програмного забезпечення, за допомогою якого було розроблено як сам проект так і тестові модулі. Розроблено UML-діаграми варіантів використання та діяльності.
- В третьому розділі проведено тестування складної клієнт-серверної системи тестами, що розроблені за допомогою обраного засобу тестування. Зроблено огляд програмних засобів за допомогою яких зручно керувати процесом тестування. Всі результати тестування зображені на рисунках.
- В четвертому розділі проведено аналіз економічних показників нової розробки. Розраховано кошторис витрат на розробку та порівняно експлуатаційні витрати нової розробки та аналога. Визначено економічний ефект від розробки.
- Отже у роботі проведено огляд засобів тестування складних-клієнт серверних систем, обрано найбільш зручний, та гнучкий засіб. Розроблено велику кількість тестових класів, що відповідає обраному засобу, а також виконано економічне обґрунтування розробки.