

РОЗРОБКА ЗАСОБІВ ДЛЯ ТЕСТУВАННЯ КЛІЄНТ- СЕРВЕРНИХ СИСТЕМ

Ткач Є. С., гр.ЗКСУА-14сп

Вінниця ВНТУ 2015

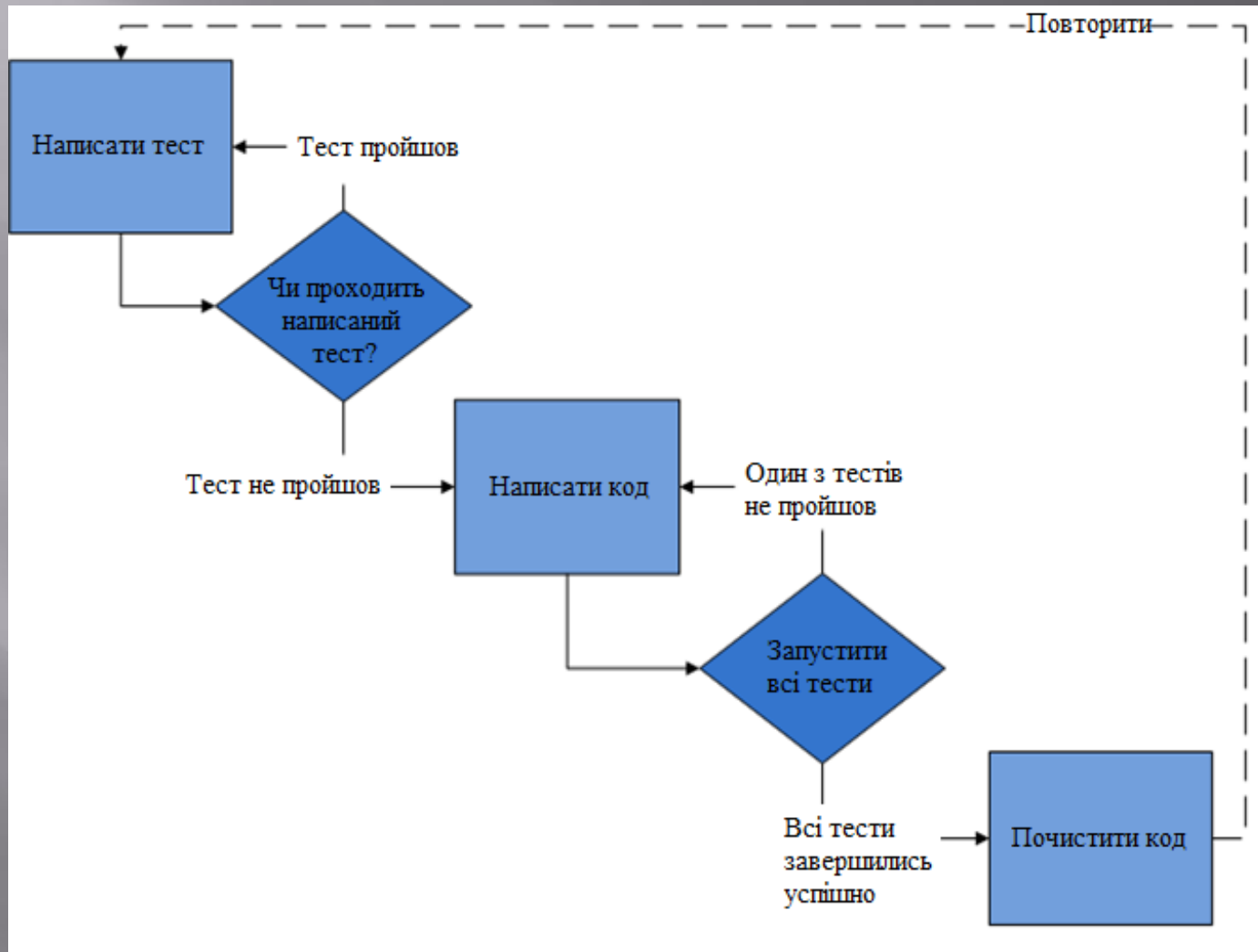
Актуальність проблеми. Загальний погляд на тестування програмного забезпечення останні роки активно еволюціонував, стаючи все більш конструктивним, прагматичним і наближеним до реалій сучасних проектів розробки програмних систем. Сьогодні тестування розглядається як діяльність, яку необхідно проводити 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 виключення.

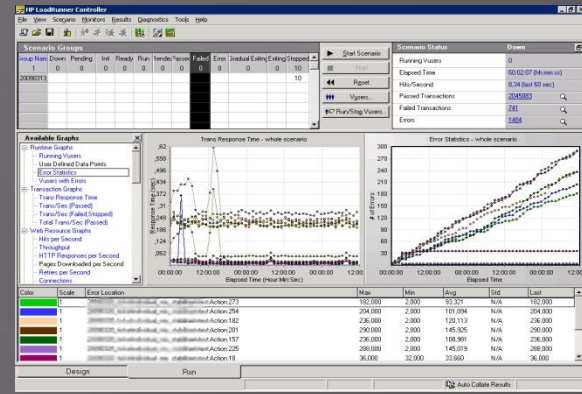
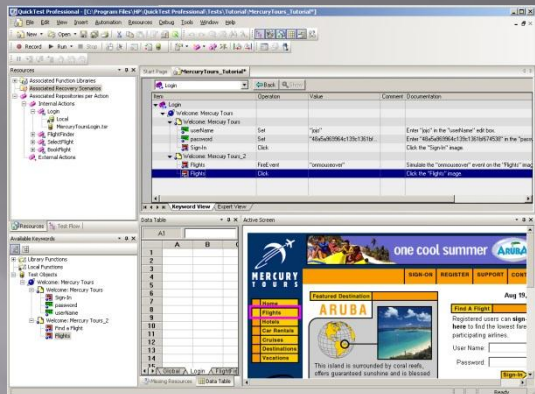
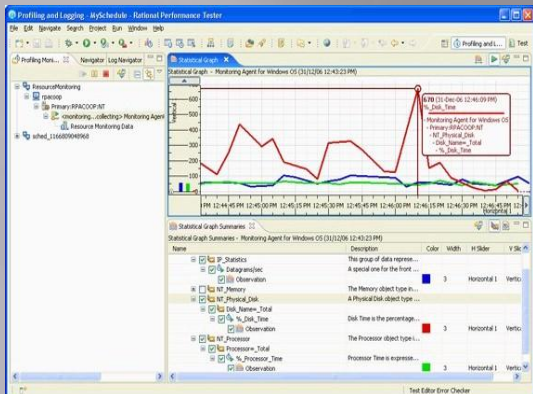
Об'єктом дослідження і розробки є процес створення програмного забезпечення, зокрема, етап тестування.

Предметом дослідження є засоби автоматизації тестування клієнт - серверних систем.

Розробка програми через тестування



Канонічні схеми регулятора стану



IBM Rational Functional Tester

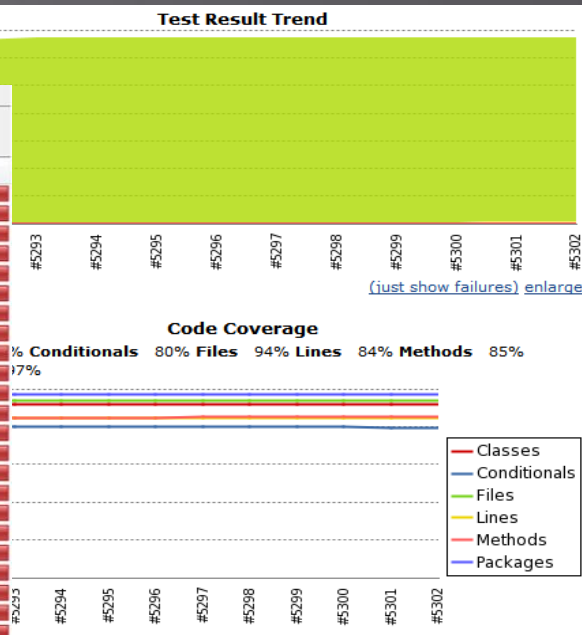
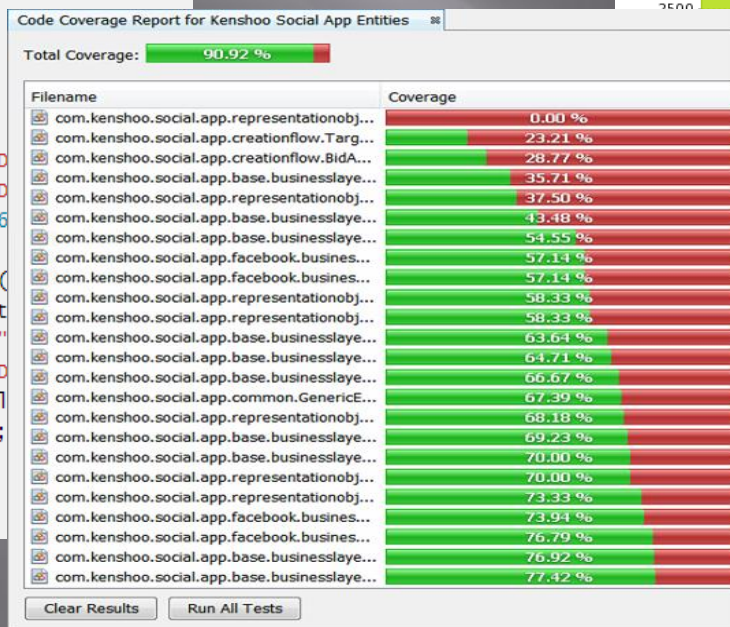
HP QuickTest Professional

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, "USD");
        Money result = m1.add(m2);
        assertFalse(!expected.equals(result));
        assertEquals(expected, result);
        assertNotNull(new Money(10, "USD"));
        assertTrue(new Money(10, "USD"));
        assertNotSame(expected, result);
        assertSame(expected, result);
    }
}
```



Junit тестування

Клієнт-серверна система

KENSHOO social Performance Management

Updated 2 minute(s) ago Get Data

Yesterday (Nov 03, 2013)

Performance

Copy View Ads Sync Filter 1-30 of 131

Unsaved Filters: Campaign Status

Campaign Name	Campaign Budget	Creation Date	Campaign Status	Clicks	Cost	CTR %	Frequency %	Impressions
111_PPTA_18max_US 1 Active Ads	\$ 2,00	05/22/2013 11...	▶	0	\$ 6,00	0,00%	0,00	0
111_PPTA_18max_US 1 copy 1 Active Ads	\$ 2,00	05/23/2013 13...	▶	0	\$ 6,00	0,00%	0,00	0
1DA_18max_US_image18 2 Active Ads	\$ 2,00	05/13/2013 19...	▶	0	\$ 6,00	0,00%	0,00	0
1DA_18max_US_image18 1 copy 2 Active Ads	\$ 2,00	05/13/2013 19...	▶	0	\$ 6,00	0,00%	0,00	0
1DA_18max_US_image18 1 copy 1 ... 2 Active Ads	\$ 2,00	05/20/2013 17...	▶	0	\$ 6,00	0,00%	0,00	0
1DA_18max_US_image18 2 copy 2 Active Ads	\$ 2,00	05/13/2013	▶	0	\$ 6,00	0,00%	0,00	0
1DA_18max_US_image18 2 copy 1 ... 2 Active Ads	\$ 2,00	05/13/2013	▶	0	\$ 6,00	0,00%	0,00	0
1DA_18max_US_image18 2 copy 1 ... 2 Active Ads	\$ 2,00	05/20/2013	▶	0	\$ 6,00	0,00%	0,00	0
1DA_18max_US_image18 2 copy 2 ... 2 Active Ads	\$ 2,00	05/20/2013	▶	0	\$ 6,00	0,00%	0,00	0
1DA_18max_US_image18 3 copy 2 Active Ads	\$ 2,00	05/20/2013	▶	0	\$ 6,00	0,00%	0,00	0
1DA_18max_US_image18 3 copy 1 ... 2 Active Ads	\$ 2,00	05/20/2013	▶	0	\$ 6,00	0,00%	0,00	0
1DA_18max_US_image18 4 copy 2 Active Ads	\$ 2,00	05/20/2013	▶	0	\$ 6,00	0,00%	0,00	0
campaign3 0 Active Ads	\$ 1,00	05/29/2013	▶	0	\$ 6,00	0,00%	0,00	0
campaign3 0 Active Ads	\$ 1,00	05/29/2013	▶	0	\$ 6,00	0,00%	0,00	0
Totals	\$ 1,760,...							

KENSHOO social

Demographics

Age

18 65

Divide into age ranges of 2 years

Exact age match ?

18 - 65

Gender

All Male Female

Divide by gender

Male and Female

Interests

Custom Audiences

REQUEST A DEMO

CLICK ANYWHERE BELOW OR SCROLL DOWN TO TAKE A QUICK TOUR!

Breakthrough social media technology that helps you **break through.**

ctCustomers #En

Strategic Preferred Marketing Developer

PAGES ADS APPS INSIGHTS

Highest distinction of excellence"

Co-development of go-to-market strategies"

"Access to alpha and beta product trials

"Priority product and business support

Facebook Homepage Ads

Facebook Newsfeed Ads

Custom Audiences

Custom Optimized CPM

1 Define Ad Destination 2 Define Targets 3 Determine how your ads will be organized

Determine how your ads will be organized

Campaign Structure

One ad per campaign ? Multiple ads per campaign ?

Campaign Budget and Schedule

Budget ?

Daily budget of

Schedule ?

Run my campaigns continuously starting

Tracking

Conversion Source ?

Channel report

Assign to Profile

Existing profile test New profile ?

Use Facebook pixel for conversion tracking ?

Bid Strategy

CPC CPM

Bid

Use Facebook Suggested Bid

Review Ads

on source, campaign budget, and schedule, and c

Total budget

About Tracking

The conversion source determines what performance and conversion metric data you will receive. You can use the Facebook pixel in combination with any of the conversion sources. [LEARN MORE...](#)

New profile ?

Use Facebook pixel for conversion tracking ?

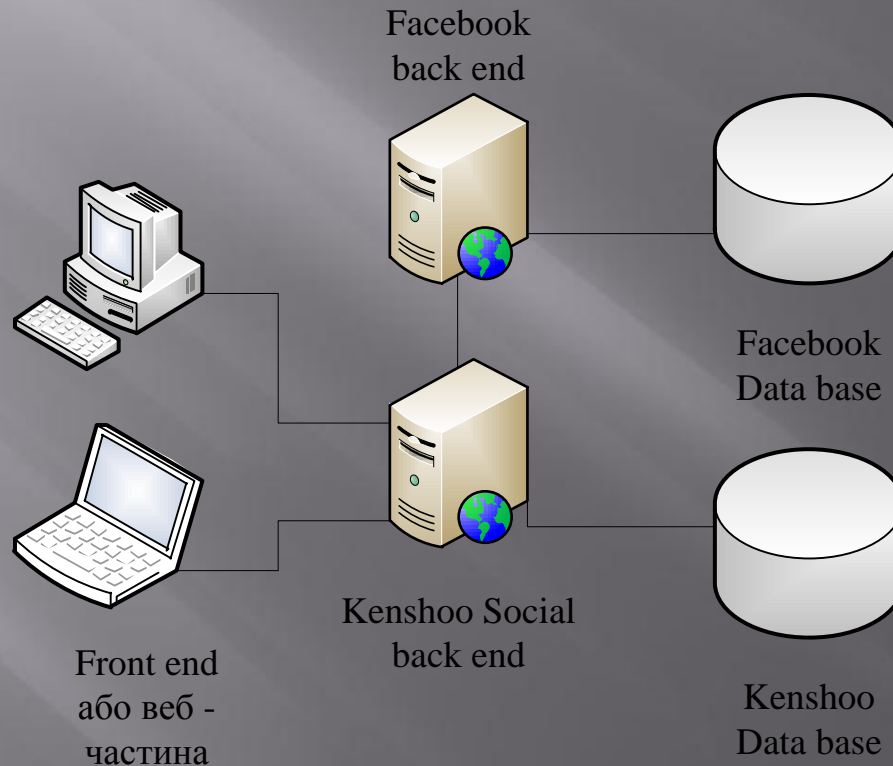
Bid Strategy

CPC CPM

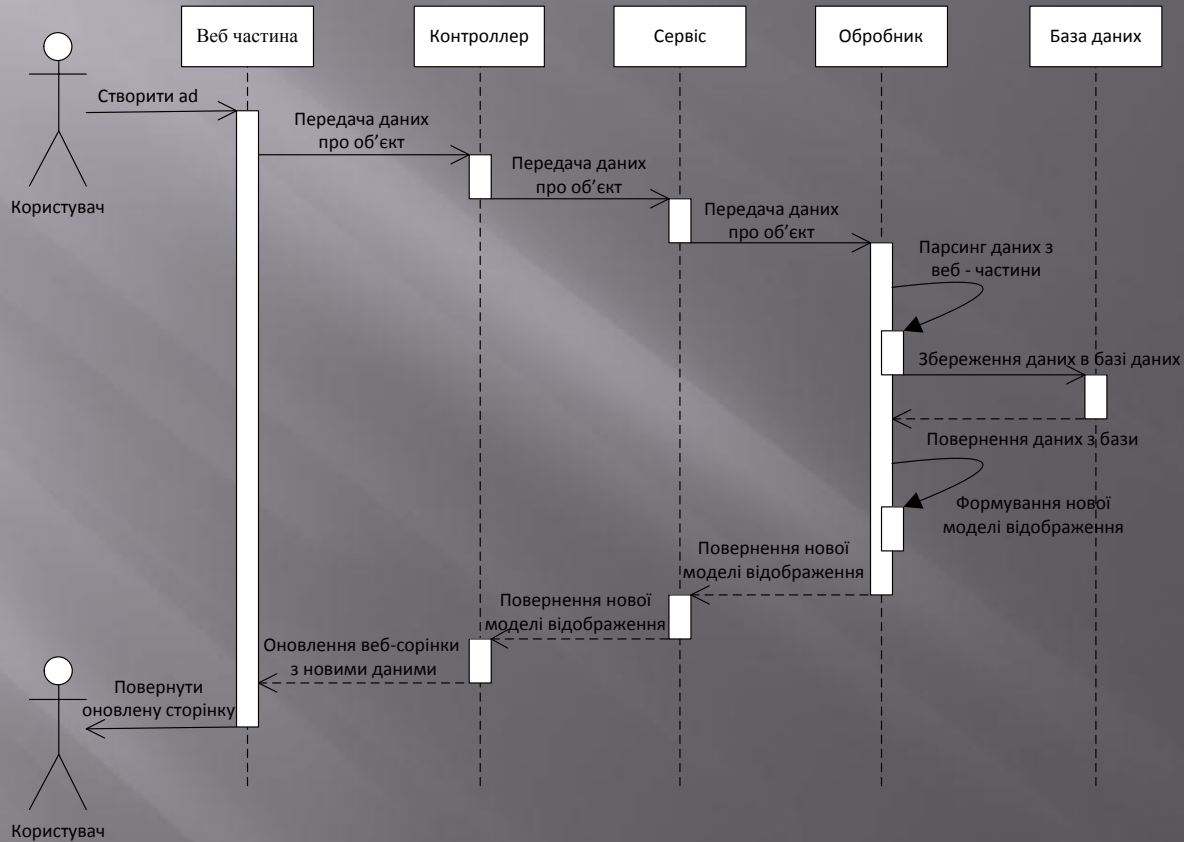
Bid

Use Facebook Suggested Bid

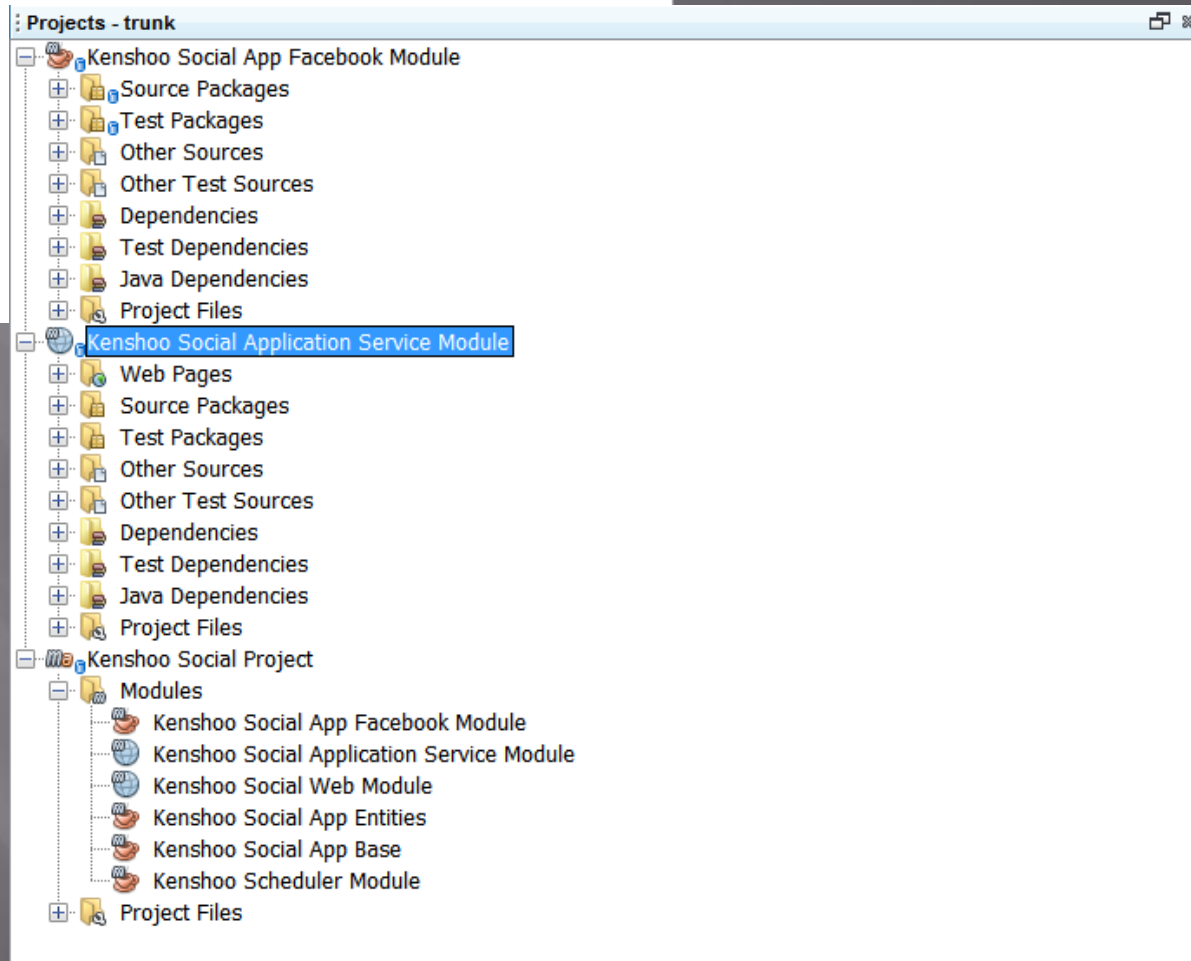
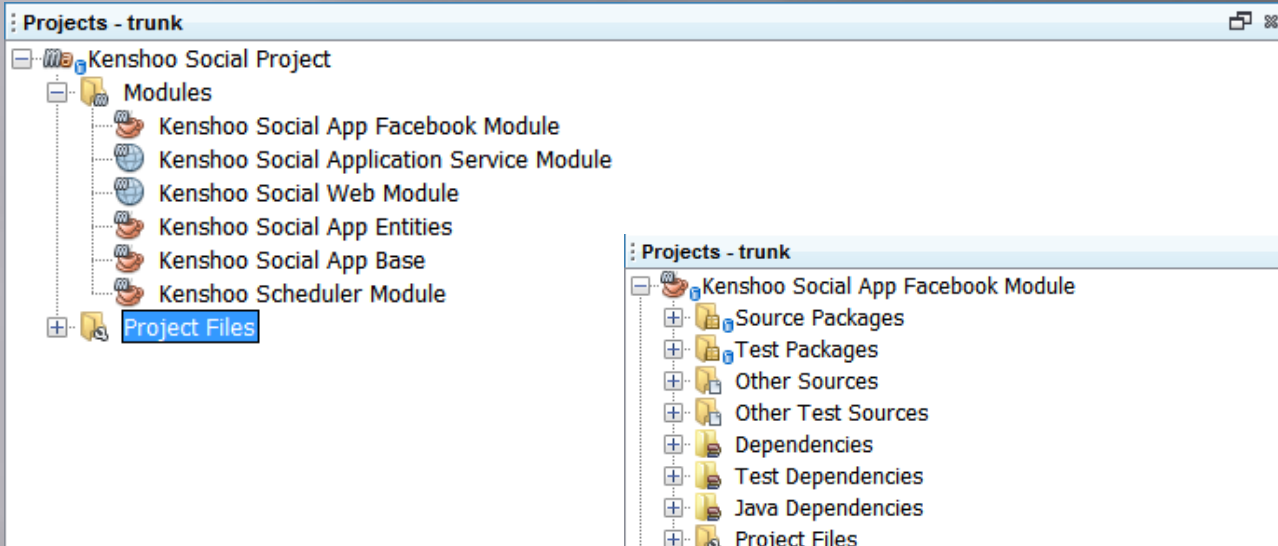
Структурна схема програмного забезпечення



UML – діаграма діяльності



Структура ПЗ



Загальна структура тестового класу

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = {"classpath:/testContext.xml"})
@Transactional
@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>
</bean>
```

Запуск тестових класів

The screenshot shows an IDE interface. On the left, a project tree is visible with 'Kenshoo Social App Entities' selected. A context menu is open over this project, listing various actions. The 'Code Coverage' option is highlighted, and a sub-menu is open showing 'Collect and Display Code Coverage', 'Clear Results', 'Show Report...', and 'Show Editor Bar'. On the right, a 'Code Coverage Report for Kenshoo Social App Entities' window is displayed. It shows a total coverage of 90.92% and a table of individual file coverage percentages.

Code Coverage Report for Kenshoo Social App Entities

Total Coverage: 90.92 %

Filename	Coverage
com.kenshoo.social.app.representationobj...	0.00 %
com.kenshoo.social.app.creationflow.Targ...	23.21 %
com.kenshoo.social.app.creationflow.BidA...	28.77 %
com.kenshoo.social.app.base.businesslaye...	35.71 %
com.kenshoo.social.app.representationobj...	37.50 %
com.kenshoo.social.app.base.businesslaye...	43.48 %
com.kenshoo.social.app.base.businesslaye...	54.55 %
com.kenshoo.social.app.facebook.busines...	57.14 %
com.kenshoo.social.app.facebook.busines...	57.14 %
com.kenshoo.social.app.representationobj...	58.33 %
com.kenshoo.social.app.representationobj...	58.33 %
com.kenshoo.social.app.base.businesslaye...	63.64 %
com.kenshoo.social.app.base.businesslaye...	64.71 %
com.kenshoo.social.app.base.businesslaye...	66.67 %
com.kenshoo.social.app.common.GenericE...	67.39 %
com.kenshoo.social.app.representationobj...	68.18 %
com.kenshoo.social.app.base.businesslaye...	69.23 %
com.kenshoo.social.app.base.businesslaye...	70.00 %
com.kenshoo.social.app.representationobj...	70.00 %
com.kenshoo.social.app.representationobj...	73.33 %
com.kenshoo.social.app.facebook.busines...	73.94 %
com.kenshoo.social.app.facebook.busines...	76.79 %
com.kenshoo.social.app.base.businesslaye...	76.92 %
com.kenshoo.social.app.base.businesslaye...	77.42 %

Clear Results Run All Tests

The screenshot shows the 'Output' window of the IDE. It displays the results of a test run for 'com.kenshoo.social:app:jar:1.0.0-SNAPSHOT'. The overall test status is '0.00 %' with a red bar. The message indicates 'No test passed, 1 test caused an error.(0.17 s)'. The failed test is 'testReadWithException' in the class 'AbstractAdStatisticsReaderTest', which caused an 'Unexpected exception, expected<java.lang.AssertionError> but was<java.lang.NullPointerException>'. The stack trace shows the error occurred in the 'evaluate' method of 'ExpectException'.

com.kenshoo.social:app:jar:1.0.0-SNAPSHOT x

0.00 %

No test passed, 1 test caused an error.(0.17 s)

com.kenshoo.social.app.facebook.businesslayer.businessworkflows.adstatisticshandler.batch.reader.AbstractAdStatisticsReaderTest FAILED

testReadWithException caused an ERROR: Unexpected exception, expected<java.lang.AssertionError> but was<java.lang.NullPointerException>

Unexpected exception, expected<java.lang.AssertionError> but was<java.lang.NullPointerException>

java.lang.Exception

at org.junit.internal.runners.statements.ExpectException.evaluate(ExpectException.java:31)

at org.junit.runners.ParentRunner.runLeaf(ParentRunner.java:263)

at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:68)

at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:47)

at org.junit.runners.ParentRunner\$3.run(ParentRunner.java:231)

at org.junit.runners.ParentRunner\$1.schedule(ParentRunner.java:60)

Висновки

- В роботі проведено аналіз та розробку засобів тестування клієнт - серверних систем.
- 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.