

A laboratory exercise in testing database applications

Javier Tuya, Claudio de la Riva, José García-Fanjul
University of Oviedo – SPAIN

tuya@uniovi.es, <http://www.di.uniovi.es/~tuya/>

WTST, January, 2008, Melbourne, Florida



Supported by: Department of Science and Technology (Spain)
IN2TEST (TIN2004-06689-C03-03), Test4SOA (TIN2007-67843-C06-01),
RePRIS (TIN2005-24792-E, TIN2007-30391-E)

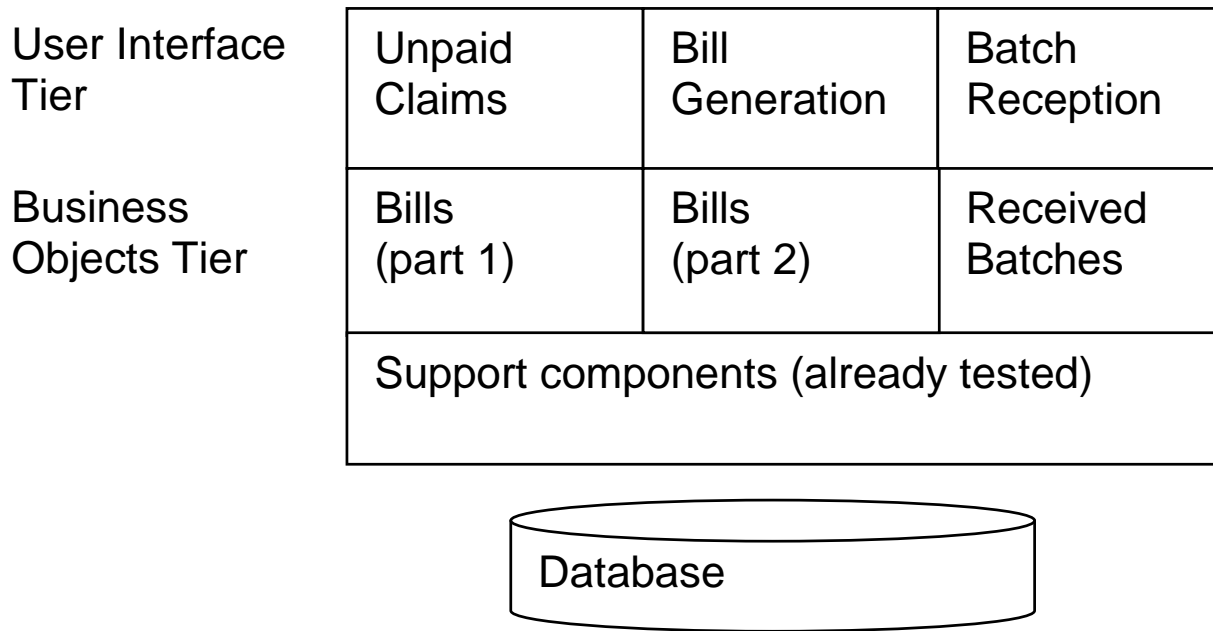
Scope



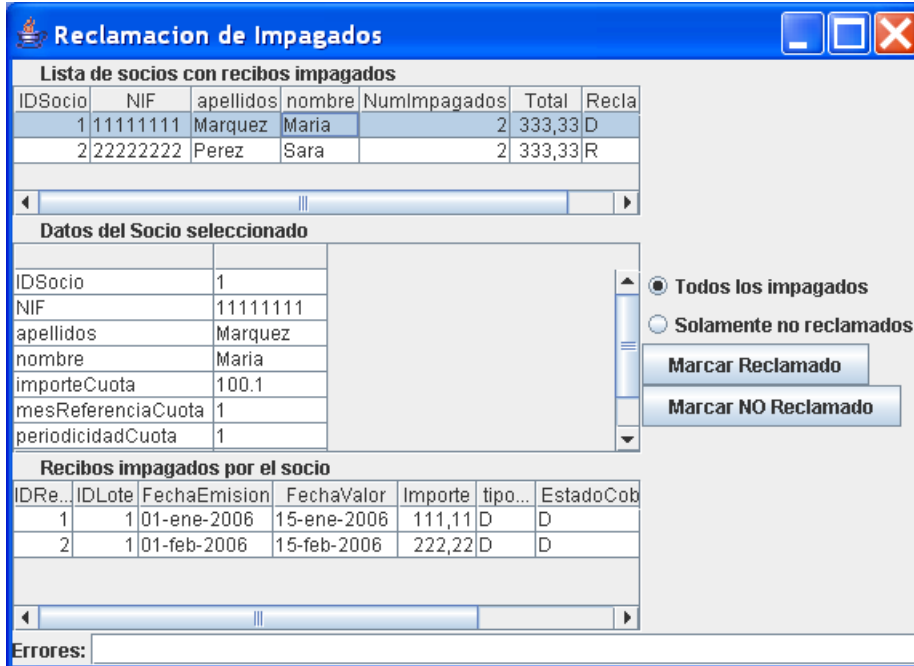
- Graduate courses in Software Engineering
 - A module in software testing (concepts, black/white-box tech.)
 - Exercises using small artifacts (short spect, piece of source code)
 - Problem: How to get a bigger picture of testing
- Laboratory exercise - testing in the context of:
 - A database application
 - Working in teams
 - Functional and unit testing
 - Test automation
 - Integration of tools
 - 12 lab-hours plus homework plus 4+2 training

Tested Artifacts - Structure

- Application: Payments by direct debit
 - Two tiers, three modules / tier
 - Unit of assignment: module



Tested Artifacts – User Interface



Reclamacion de Impagados

Lista de socios con recibos impagados

IDSocio	NIF	apellidos	nombre	NumImpagados	Total	Recla
1	11111111	Marquez	Maria	2	333,33	D
2	22222222	Perez	Sara	2	333,33	R

Datos del Socio seleccionado

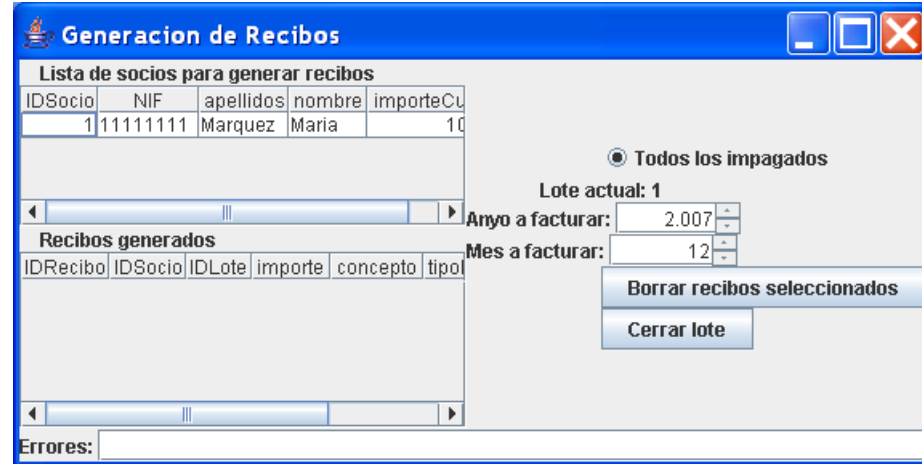
Todos los impagados
 Solamente no reclamados

Marcar Reclamado
Marcar NO Reclamado

Recibos impagados por el socio

IDRe...	IDLote	FechaEmision	FechaValor	Importe	tipo...	EstadoCob
1	1	01-ene-2006	15-ene-2006	111,11	D	D
2	1	01-feb-2006	15-feb-2006	222,22	D	D

Errores:



Generacion de Recibos

Lista de socios para generar recibos

IDSocio	NIF	apellidos	nombre	importeCu
1	11111111	Marquez	Maria	10

Todos los impagados

Lote actual: 1

Ano a facturar: 2.007
Mes a facturar: 12

Recibos generados

IDRecibo	IDSocio	IDLote	importe	concepto	tipol
----------	---------	--------	---------	----------	-------

Borrar recibos seleccionados
Cerrar lote

Errores:



Recepcion de Lotes

Lista de lotes en proceso

ID	Lote	Num	Fecha	Estado	Incidencias
20	222	1	15-ene-2006	E	No existe lote
99	999	0	15-ene-2006	E	No existe lote

Todos
 Solamente los erroneos

Devoluciones para el lote seleccionado

ID	Recibo	Socio	Importe	Concepto	Emitido	Valor
20	9	9	11,2	concepto	15-ene-2006	01-ene-2006 11112

Validar lotes recibidos

Errores:

Tested Artifacts – Doc & Tools



□ Documentation:

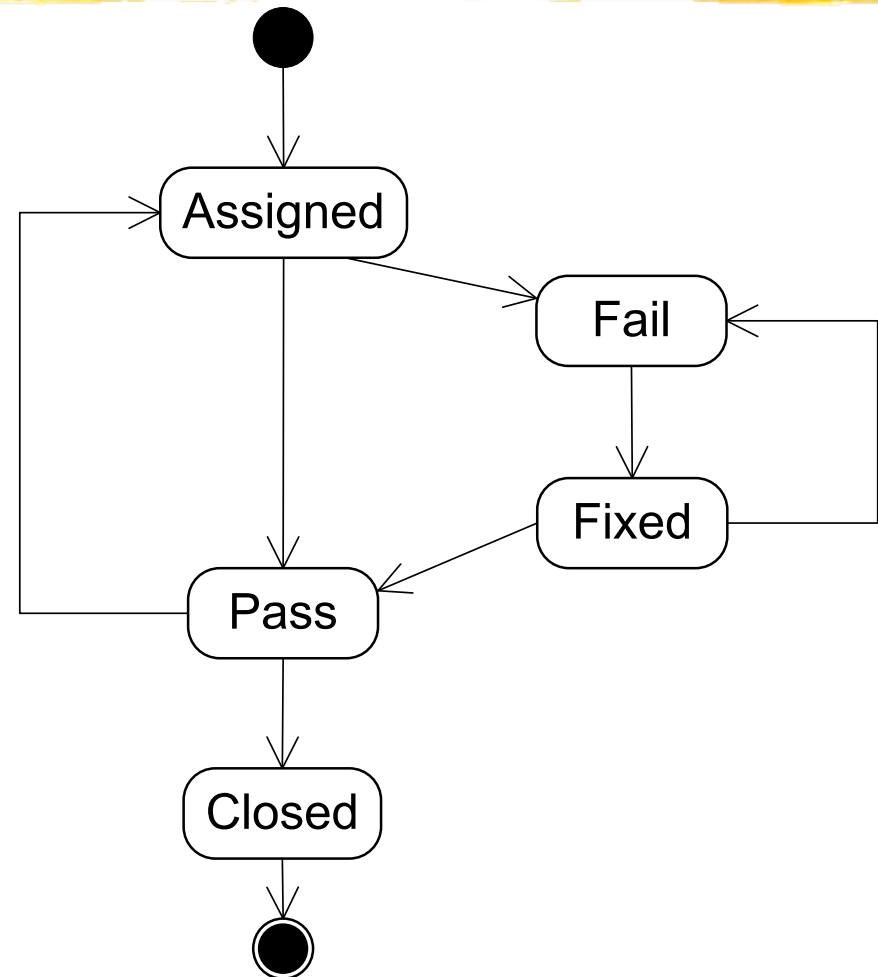
- Work procedures (3 pages)
- Work instructions (6 pages)
- Use cases (2 pages)
- Data model (3 pages)

□ Tools:

- Eclipse with JUnit
- CVS
- Helpdesk (software bug reporting database)
- Clover (code coverage)
- Data load support methods (not DBUnit)

Test Process

- ❑ Assignments student (testing)
 - 1 business component
 - 1 user interface component
- ❑ Three students per project
- ❑ Roles (alternative)
 - Tester
 - Developer
- ❑ Workflow (controled by the helpdesk system)



Discussion (1)



- ❑ Before beginning: highly motivating
- ❑ After beginning: difficult
 - Specifications do not tell everything: Doc. Fragmented into use cases, database and code. Effort in reading and synthesizing.
 - Specifications are ambiguous or apparently inconsistent: use cases and method comments use natural language. Effort to remove ambiguities.
 - No failures are found: but at least the injected faults are present. Effort to develop more effective test cases.
 - Reported bug are not always understandable: Develop the ability to communicate effectively, failures must be precisely reported in the helpdesk.

Discussion (2)



- ❑ At the end: good experience
- ❑ Some issues to discuss. How to effectively teach to avoid:
 - Irrelevant test cases: no testing the validation of user interface and database fields. Focus on the behaviour of the application and on the database states and changes.
 - More white-box than black-box (business processes): Source code is available. Some test cases designed only to cover the code, forgetting key issues about the specified behaviour.
 - Difficulties to automate functional tests (user interface): Overhead imposed by the automation of the test cases often hinders the task of designing good test cases.
 - Poorly documented and difficult to maintain test cases: Many tests that perform very small database loads using a large amount of source code.
 - Communication problems: Problem reports without enough information. Overhead in discussions.