



Software Engineer group 2: Salesmen

---

# Software Test Document

---

*author:* Patrick Provinciael



**Abstract:** This document will outline the various tests and testing procedures required for the Salesmen Project.

*11-12-2009: Addition of JUnit*

# History of versions

Version	Date	Author(s)	Changes
0.1	29-11-2009	Patrick	First Draft
0.8	06-12-2009	Patrick	Draft before QA
1.0	07-12-2009	Patrick	QAed version
1.2	11-12-2009	Patrick	Addition of JUnit

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Scope . . . . .	3
1.2	References . . . . .	3
1.3	Definitions and acronyms . . . . .	3
<b>2</b>	<b>Test plan</b>	<b>4</b>
2.1	Introduction . . . . .	4
2.2	Test items . . . . .	4
2.3	Features to be tested . . . . .	4
2.4	Features not to be tested . . . . .	4
2.5	Approach . . . . .	4
2.5.1	Kind of tests . . . . .	4
2.5.2	Unit tests . . . . .	5
2.5.3	Integration tests . . . . .	5
2.5.4	Regression tests . . . . .	5
2.5.5	Interface tests . . . . .	5
2.6	Item pass/fail criteria . . . . .	5
2.7	Suspension criteria and resumption requirements . . . . .	5
2.8	Test deliverables . . . . .	6
2.9	Test tasks . . . . .	6
2.10	Environmental needs . . . . .	6
2.11	Responsibilities . . . . .	6
2.12	4.2.13 Stafng and training needs . . . . .	6
2.13	Schedule . . . . .	6
<b>3</b>	<b>Overview of the integration tests</b>	<b>7</b>
3.1	Scope . . . . .	7

# Chapter 1

## Introduction

### 1.1 Scope

The purpose of this document is to describe the Software Test Plan(STD) for the Salesmen Project (1). It will outline the various testing mechanisms and the responsibilities of these testing mechanisms.

Chapter 2 outlines the testing plan, while chapter 3 details the the integration tests. The results of the various integration tests will be included in this document.

### 1.2 References

- Salesmen Project, [http://wilma.vub.ac.be/~se2\\_0910](http://wilma.vub.ac.be/~se2_0910)
- Software Design Document
- Software Requirements Specification
- Software Quality Assurance Plan
- JUnit Testing Framework, <http://junit.org/>
- Google Code Repository

### 1.3 Definitions and acronyms

- QAM: Quality Assurance Manager
- SQAP: Software Quality Assurance Plan
- SCMP: Software Configuration Management Plan
- SPMP: Software Project Management Plan
- SDD: Software Design Document
- SRS: Software Requirements Specification
- STD: Software Test Document
- JUnit: Java Unit testing Framework

# Chapter 2

## Test plan

### 2.1 Introduction

The purpose of this chapter is to outline the various testing mechanisms and the responsibilities of these mechanisms.

### 2.2 Test items

The following items must be tested.

- All methods and classes as outlined in the SDD (2)
- All requirements of the system as detailed in the SRS (3)
- The quality requirements as detailed in the the SQAP (4)
- Every piece of code, part of the completed system.

### 2.3 Features to be tested

Every feature implemented in the final completed system must be tested. These must be tested as soon as possible and whenever changes are made.

### 2.4 Features not to be tested

Every feature which is not (yet) implemented.

### 2.5 Approach

Several methods will be used to test the various features.

#### 2.5.1 Kind of tests

The following tests will be run.

- Unit testing of all bussiness logic with JUnit.

- internal testing of the various tools used. This will be done with the testing tools provided by these tools. When no testing facilities are provided, testing of these tools will be done within the integration tests.
- Integration tests.

### 2.5.2 Unit tests

The QAM will foresee the possibility of writing Unit tests. Each programmer is responsible for writing these tests.

These unit tests will be created to run with the JUnit testing framework (5).

Whenever changes are made, the QAM will run all Unit Tests. The programmers are encouraged to run the full Unit Testing suite as well.

### 2.5.3 Integration tests

Integration tests will be run whenever a new stable version of the system is released. The purpose of the integration tests is to ensure that the system works as a whole. Integration tests are focused on testing whether the requirements, as outlined in the SRS, are incorporated. More details regarding integration tests can be found in chapter 3.

### 2.5.4 Regression tests

Whenever a new iteration is about to start, there will first be a round of regression tests. This means that all functionality added in the previous iteration is tested to ensure that nothing was broken in the previous iteration.

### 2.5.5 Interface tests

It is important to assure that the User Interface is userfriendly. To ensure this fact, the following methods will be used.

- The interfaces will be developed with userfriendliness in mind.
- The interfaces will be tested by the other projectmembers. Each will evaluate if the interface was userfriendly enough. If necessary, the interface will then be adjusted and a new round of Interface testing will start.
- When all projectmembers deem the interface userfriendly, the User Interface will be tested by one or more external parties. These persons must all have different backgrounds of the targetted audience. This to ensure that every person will find his needs in the system.

## 2.6 Item pass/fail criteria

For every test, the pass criteria is the lack of errors or other problems.

## 2.7 Suspension criteria and resumption requirements

Testphases can be suspended in the following cases.

- When a unit test fails. If the error appeared in code that was recently written by the tester, he is encouraged to track down the error and fix this. When the error appears in code not written by the test, an issue must be made on the Google Code Repository website (6).

- When an integration test fails. In this case, an issue has to be made on the Google Code Repository Website. The integration test must start over again when the issue is resolved.

## 2.8 Test deliverables

For every code file, there must be a test file. This test file must detail every test applicable on this code. Whenever the code is changed, the test file must be reviewed to ensure all functionality is still tested.

Every other test result will be included in an appendix of this document.

## 2.9 Test tasks

The unit tests are the responsibilities of the developers. It is the responsibility of the QAM to assure that these tests are complete. Integration tests will be mainly completed by the QAM, but individual projectmembers are encouraged to run their own integration tests.

## 2.10 Environmental needs

Every person is required to have the JUnit plugin for Eclipse, in order to perform the various unit tests.

## 2.11 Responsibilities

The unit tests are the responsibilities of the developers. It is the responsibility of the QAM to assure that these tests are complete. It is also the responsibility of the QAM to assure that everything else is tested.

## 2.12 4.2.13 Stafng and training needs

When needed, the QAM will explain the creation and use of the JUnit testing framework to the individual members.

## 2.13 Schedule

Whenever anything new is added to the system, there must be tests.

## Chapter 3

# Overview of the integration tests

### 3.1 Scope

The purpose of this chapter is to document the various integration tests required. These tests will be conform with the SRS (3) and thus, they will each contain a reference to the correspondent SRS Requirement. Every Requirement will be tested for a possible negative and positive result.