

---

Capita Selecta: Software Engineering

# Project Software Engineering

Deliverable 1

---

**Christophe De Troyer**      **Maarten Vandercammen**

DATE: *16th June 2015*

DUE : *17th June 2015*

## CONTENTS

<b>1 Detailed test strategy</b>	<b>3</b>
<b>2 Accounts</b>	<b>3</b>
<b>3 Snippets</b>	<b>3</b>

## 1 DETAILED TEST STRATEGY

In order to assure proper workings of the back-end it is needed to create unit tests that cover all of the controllers. Since this sprint will only implement features of the high priority slice only two controllers need proper testing: the account controller and the snippet controller.

## 2 ACCOUNTS

**UNIT TESTS** The account controller's functionality is comprised of two important operations: logging in and registering a new account.

**Login** To test the login process it requires some special testing. The implementation notifies the user of a wrong password or wrong email address. As such tests need to be written to assure that the system detects which piece of information is wrong: the username or the password and notifies the user as such. And finally we need to test that proper login also work.

**Registration** To test the registration process one has to test the registration process. However, this process is not contained within the controller but within the Custom-MembershipProvider. The only thing to test in the HomeController itself was

## 3 SNIPPETS

**UNIT TESTS** The main code unit to test for snippets was again the snippet controller. The implemented features in this sprint were the listing of snippets and the creation of snippets.

**Listing** To test the creation of snippets one has to check the validity of the input at the controller. Hence, tests need to be created to edit snippets and to view snippets. The latter is a simple test to make sure the controller does a proper redirect or returns the proper snippet. The former needs to verify the user can only request to edit or create snippets he has access to and that the edits are processed properly by the controller. The storage of snippets can not be tested however. The application uses Entity Framework as its backend and thus we find that this software should not be tested.

**Editing** To properly unit test the listing of snippets it suffices to test that the controller returns a list of all the valid snippets when a user requests a list.