TensionCamApp v0.1: Test Report

1 Introduction
An application that analyses the pressure on washers in a wind power station.

1.1 Purpose of application
The idea is to provide an easy and efficient way to measure the pressure on the washer and compare differences over time. This is done by capturing a photo from the mobile device on a number of white dots on a cap. This photo is sent through the application to a web server where the analysing software is located which analyses the photo and sends back the result to the user.

1.2 Characteristics
The application provides basic functionality of what is required but is deliberately developed to make it easy to add other functionality onto after user reviews.

2 Test environment
Test has been performed manually due to the flexibility is provides in the working progress.

2.1 Hardware specifications
The test has both been performed in an Android Emulator on an OS X system and on a LG Nexus 4. Some tests, both automatic and acceptance, requires an SD card and those has been ran on the mobile device but would most like work as well with an emulated SD card.

2.2 Software specifications

OS X versions
- 10.8.3 Mountain Lion (mainly)
- 10.6.8 Snow Leopard (partially)

Android versions
- (mainly)
- 4.1.2 Jelly bean (partially)

3 Bugs and limitations
No known bugs or limitations in the application. However the application relies and depends on external software in the analyse program which, although tested throughout and trusted, cannot be accounted for.

4 Test specifications
Since the program does not rely on uncontrolled input from the users the test mainly consists of acceptance test that insures convenience for the user. This is partially motivated by that a lot of classes and methods depend on each other over the borders. Therefore it was more important to insure that the process was impeccable rather than single methods, which also could be seen as a consequence of our client focus. The process performance was easier to conclude through acceptance test rather than automatic tests. Automatic test has

of course been constructed, mainly when necessary and possible to secure that a specific method performs the wanted action.

4.1 Acceptance tests

The acceptance tests are described in the document “Acceptance tests” in the path TensionCamApp/docs

4.2 Automatic tests

Automatic tests are included in a separate project folder called TensionCamAppTester. To run the tests using Eclipse ADT, you should mark the project folder and choose to run it as “Android JUnit Test”. Eclipse will then automatically run included tests and display the test results.

The tests can also be tested individually. To do so; expand the project folder in the package explorer, mark desired test, and choose to run it as “Android JUnit Test”.

Note that the main project ‘TensionCamApp’ needs to be included in the build path for the ‘TensionCamAppTester’.

4.2.1 Unit tests

In this project used to test methods in controller and util classes that are hard to conclude if they work correctly or not by acceptance tests

4.2.2 GUI tests

Mainly used to insure functionality beyond acceptance testing in those cases when a correct implementation is crucial for the usability.

5 Test report

5.1 Functional tests

From the document “Acceptance Test” in TensionCamApp/docs.

Test ID	Result	Comment
TC01	Passed test	
TC02	Passed test	
TC03	Passed test	
TC04	Passed test	
TC05	Passed test	
TC06	Passed test	
TC07	Passed test	
TC08	Passed test	
TC09	Passed test	

5.2 Unit Test Report

From the test in the folder “TensionCamAppTester” in TensionCamApp/

Test ID	Result mobile device	Result emulator	Comment
TC01	Passed test	Passed test	

TC02	**Passed test**	**Passed test**	
TC03	**Passed test**	**Passed test**	
TC04	**Passed test**	**Passed test**	
TC05	**Passed test**	**Passed test**	
TC06	**Passed test**	**Passed test**	
TC07	**Passed test**	**Passed test**	