

Test Specification

Project 3: Decision tree engine app



1. Introduction

Information and forms to fill for all tests to be performed in Project 3: Decision tree engine app can be found within this document. This document will also include information about the purpose behind each test and results of tests that already have been carried out.

2. Background

Environmental friendliness and comprehensibility^[1] should be taken into account by the developer^[2] in every project. The comprehensibility of the applications should not be assumed by the developer, it will not automatically be comprehensible for the end-user, therefore the design needs to be tested to seize if it's fathomable for the end-user. The applications need to be tested to see if the customer is able to manage without any assistance from the developer.

One of the two applications is intended for an administrator and the other for an end-user. The administrator application is designed to function as a survey builder, where the administrator will be able to create, modify and delete surveys. Different kinds of surveys, estimated economy calculations and locally saved documents containing their results should be viewable through the end-user application.

3. High-level overview

In order to receive feedback on the design of the end-user application, a comprehensibility test was made based on the end-user smartphone application design. The outcome of the test needed to show that the design was comprehensible for others and not only for the customer.

To evaluate the administrator application the customer needs to be able to add, edit and remove surveys, questions and languages in an easy manner. The customer will be given a set of tasks and changes may be made based on the outcome.

The end-user application will be put through an intensive user testing in an attempt to break the application, to ensure correct error handling.

Another intensive user testing will be made for the database in an attempt to find the upper limit of simultaneous push and pull requests. If no corruption or error occurs the database pass this test.

Lastly the end-user application will be given to the customer in an acceptance test, where the customer will be given a set of tasks which needs to be completed without assistance in order for the application to pass. The app should be able to function offline, send a message to an e-mail and have the possibility to choose a language.

3.1 End-user site design test

Users for this test was hand picked among students registered in the course DVA313 (limited by request of secrecy by the client). They were either third year computer science bachelor students or fourth year integrated systems master students. All test persons therefore has either some or extensive knowledge in the field of GUI design, which was taken into account when the test was designed in-order to get as valuable feedback as possible.

The test person was first given a brief description of the purpose of the application. This was done in-order to give the test person a scenario as of why they're using the application.

Then the test person was given a smartphone with the illusion of the application already running, known as the Wizard-of-Oz-technique^{[3][4]}, and was told to open the automation decision tree, answer it to the end (the amount of questions was shortened down since the test users lack necessary knowledge to answer all of them) and to view the report. If the test user got stuck anywhere in the process they were given extra verbal information for the sake of being able to continue.

When the test user either aborted the test, or completed a full run, and returned back to the main menu they were given an A4 paper containing all views of the application and an ink pen while being asked to comment on all design flaws they could find.

These comments were then used in order to improve the first design draft.

The results of this test can be found in appendix 1.

3.2 Administrator site acceptance test

The client will be given the following list of tasks to complete which will cover all the implemented functionality of the program

1. Create Survey
2. Edit the survey
3. Add a question to it
4. Add possible answers to the question
5. Delete an answer
6. Delete a question
7. Delete the survey



If it takes more than an appropriate amount of time or if the client finds any of the functionality unsatisfying the test will be failed.

3.3 End-user site stress test

The interface responsiveness and error handling will be tested in an attempt to force the application to corrupt data or throw an error and cancel. This will be done by high frequency manual input in a random/unusual pattern compared to the most common program flow.

If the application remains free of corruptions and if it does not throw an exception it will have passed this test.

3.4 Database stress test

The database will be tested in an attempt to find the upper limit of simultaneous pull requests. This will be done by having several smartphones request to update from the database at the same time.

If an upper limit cannot be found or if it's deemed to be within a reasonable amount the database will have passed this part of the test.

Simultaneous push requests should also be tested to check for appropriate countermeasures against the possible corruption of data.

This test will be cleared if the data remains free of corruptions no matter if updates were discarded.

3.5 End-user site acceptance test

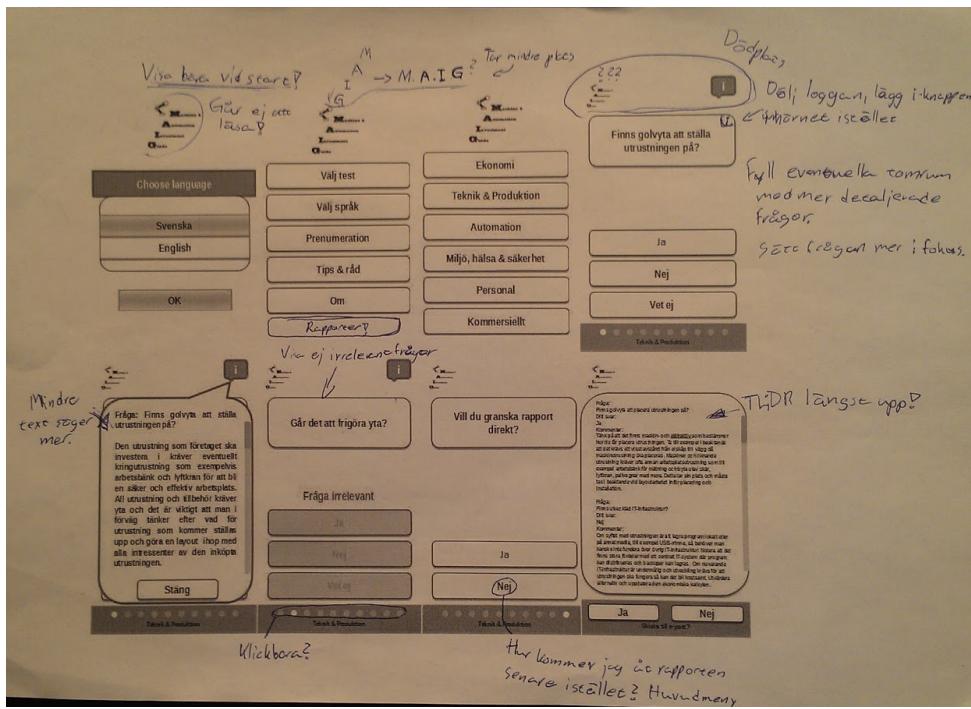
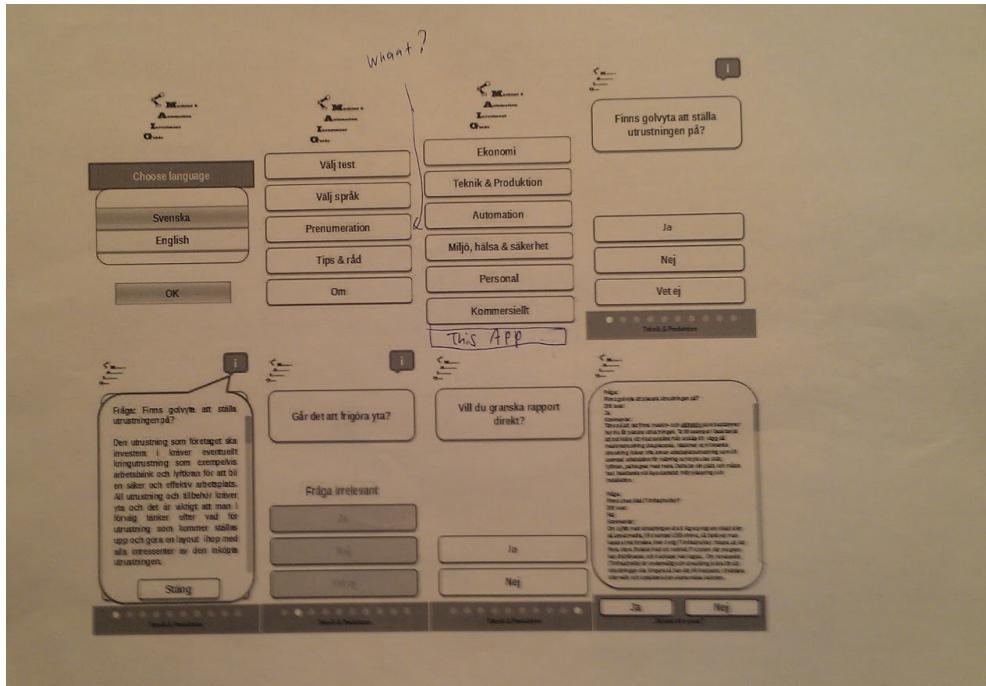
The Client will perform the following list of tasks without assistance:

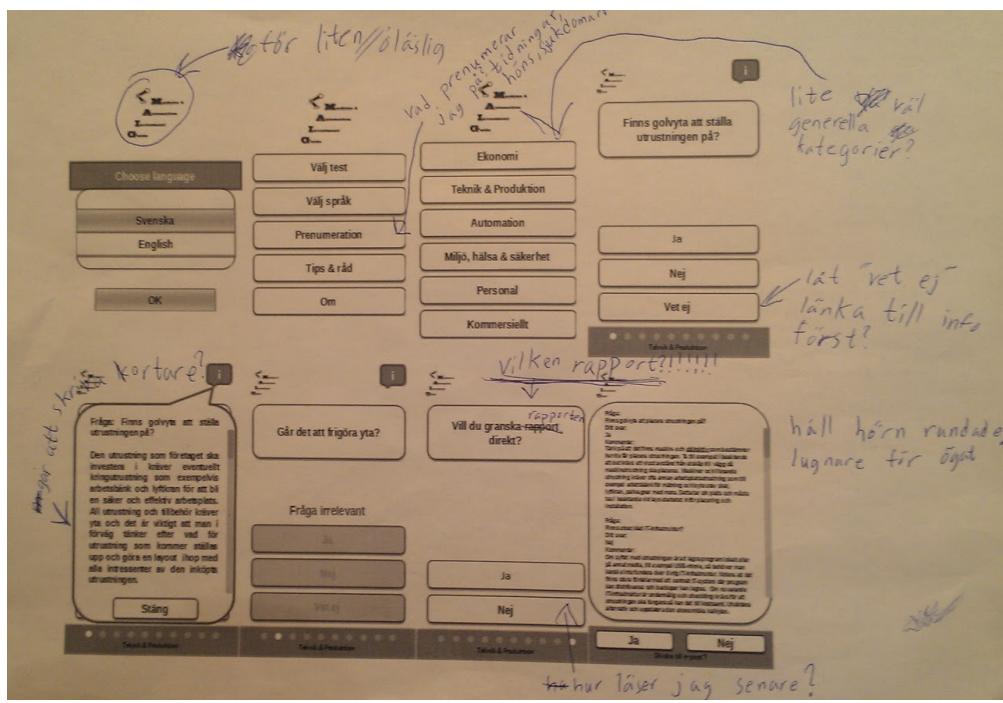
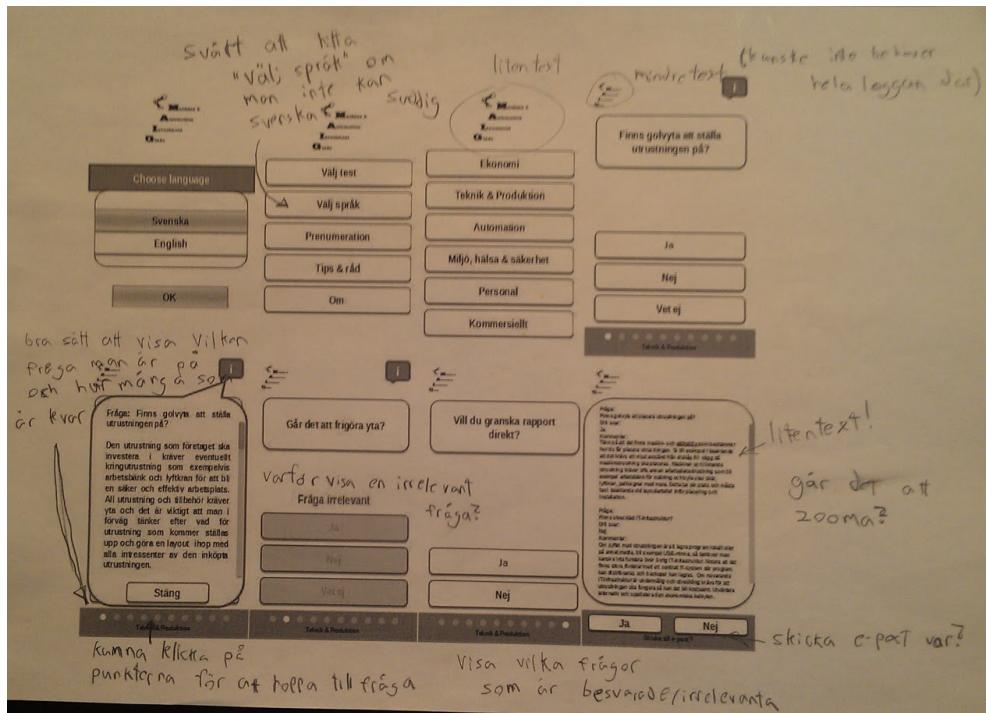
1. Start the application
2. Retrieve the latest data from the database
3. Choose a **test**
4. Read the additional information
5. Answer a question 
6. Go back and change an answer
7. Complete the survey
8. View the report
9. Email the report to yourself

If they can be completed within a reasonable amount of time without complications the test will be deemed passed as soon as the client also gives their approval.

4. References

- [1]. Benyon, David. Perception and navigation. In *Designing Interactive Systems*, 623-644, ISBN - 9780321435330, 2007.
- [2]. Wikipedia, *Software developer*. http://en.wikipedia.org/wiki/Software_developer (Accessed 2014-12-18)
- [3]. Buxton, Bill. The Wonderful Wizard of Oz. In *Sketching User Experiences*, 239-244, ISBN - 9780123740373, 2007.
- [4]. Salber, Daniel and Coutaz, Joëlle. *Applying the Wizard of Oz Technique to the Study of Multimodal Systems*. no.1. 2005.
http://link.springer.com/chapter/10.1007%2F3-540-57433-6_51 (Accessed 2014-12-17).





5.2 Administrator site acceptance test

Unsatisfying functionality	
Time	
User	

5.3 End-user site stress test

Exceptions thrown	
Corrupted data	
Time	

5.4 Database stress test

Upper limit	
Corrupted data	

5.5 End-user site acceptance test

Time	
Complication	
Approval	
Client	