



Bachelor Thesis Project

Title of your thesis project

- *Optional subtitle*



Author: Christer Hamberg
Supervisor: Johan Hagelbäck
Semester: VT 2017
Subject: Computer Science

Abstract

The report shall begin with a summary, called abstract. The abstract shall not be longer than a paragraph, and is not divided into more than one piece. It shall contain:

- A short background description to the area of your project
- A description of your research problem
- A motivation why this problem is interesting to investigate
- What you have done to answer the problem
- A short summary of your results

From reading the abstract the reader should clearly understand what the report is all about. The purpose of the abstract is to make the reader interested in continue reading the report, if it covers something that the reader wants to know more about.

Keywords: fill in some keywords for your work here. Examples:
software architectures, adaptive systems, network intrusion detection, ...

Preface

You can have a preface in the report if you want, but it is not necessary. In this you can write more personal reflections on your thesis project. In the preface you can also take the opportunity to thank the people who have been particularly helpful during the report writing, for example if you had any contact with a company that helped with the project, people that guided or helped you during the project, or your family and friends that supported you during the project. The preface shall not be longer than half a page.

Contents

1	Introduction	6
1.1	Background	6
1.2	Previous research	9
1.3	Problem formulation	11
1.4	Motivation	11
1.5	Research Question	11
1.6	Scope/Limitation	12
1.7	Target group	13
1.8	Outline	13
2	Method	15
2.1	Scientific Approach	15
2.2	Method Description	15
2.3	Reliability and Validity	15
2.4	Ethical Considerations	15
3	Implementation	16
4	Results	17
5	Analysis	19
6	Discussion	20
7	Conclusion	21
7.1	Future Research	21
	References	22
A	Appendix 1	24

1 Introduction

In July 1969 a man stepped on the moon for the first time. Almost 50 years later with all the technology advances available, we still struggle to run regression testing of interactive web applications in a successful and cost efficient way.

The repetitive task of regression testing can be one of the costliest testing activities performed in the software development project. The same test cases are repeated being executed in order to ensure that introduced code changes does not break anything in the application. Due to the frequency and repetitive nature of the task the execution depends heavily on automation of test execution in order to achieve both speed and cost efficiency in the regression test activity.

Automated testing of single-page applications (SPA) is hard to achieve as test cases regularly breaks, test execution takes long time and test cases can result in flaky results. As a result of these problems testing is often split in different parts, which are tested separately from each other.

This thesis studies the problems observed during regression testing of a commercially available web application and proposes solutions on how to overcome these issues. With the intention to avoid splitting up the testing of the different components separately from each other.

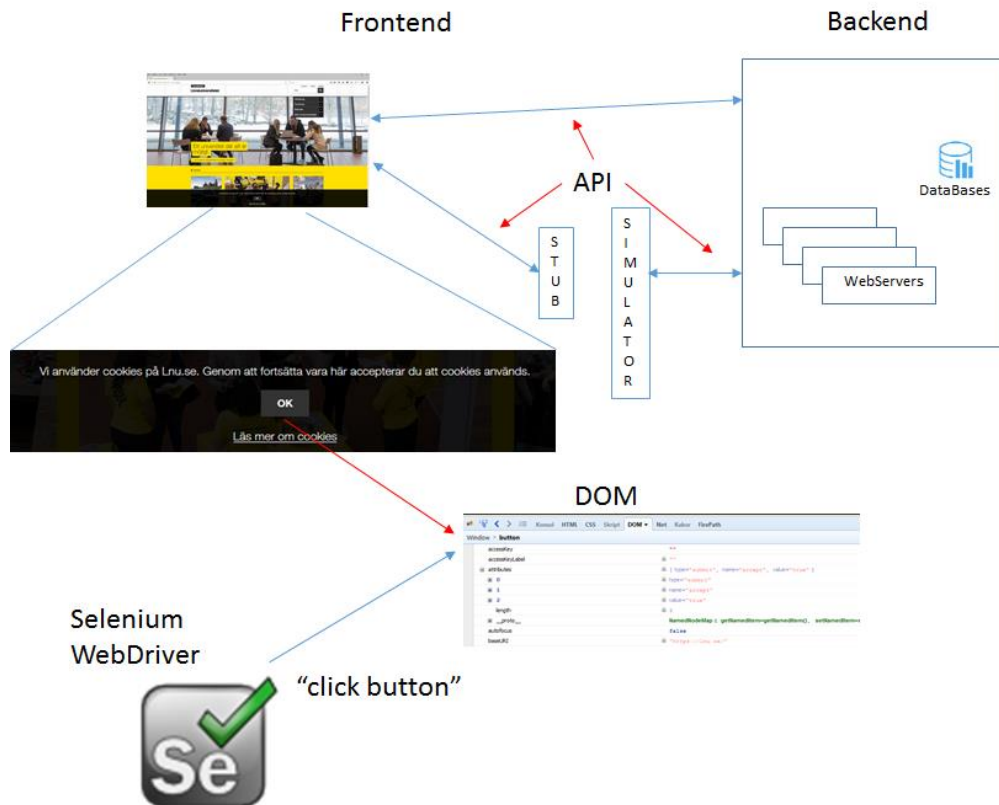
In this chapter you shall give an introduction to your thesis project. It shall start with a broad overview of what your project is all about. Similar to the abstract, the introduction shall make the reader interested in continue reading your report. Don't be too detailed here; there are plenty of opportunities to add details in later chapters.

1.1 Background

<Fix the missing links>

An SPA is logically divided into two different entities [Fig 1], the frontend and the backend. The backend implements the business logic needed by the system with for example file/database access to retrieve data needed for the realization of the functions.

The frontend implements the graphical user interface, used by the user to interact with the system. In an SPA the GUI is realized as a web page running in an ordinary web browser. The communication between the frontend and backend is done over an application programming interface (API). Which enables the frontend to send and receive requests and data from the backend. SPAs implemented today are implementing the same kind of functionalities as ordinary desktop applications are, and could be seen as ordinary desktop applications.



[Fig 1.] Test Automation of SPA

Regression testing

Testing is done in order to build confidence that the software works as intended. When a piece of code is modified, regression test is performed and as described in the IEEE definition a selected subset of test cases is executed.

However regression testing is not done in order to test that the code change functions as intended. Instead regression testing focuses on making sure that the changed code did not break any existing functionality.

Development of a software application is an iterative process, causing numerous software changes on already tested and approved code. These iteration causes a need for re testing of the code, regression testing. Due to the repetitive nature of the regression test activity [REF xxx] it is one of the most expensive test activities, consuming upto 80% of the testing budget [REF yyyy].

The IEEE definition of regression testing is *Selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements.*

Interactive web applications

An interactive web application is in most cases logically separated in two parts. A backend that implements the functional/business logic of the application and a frontend which implements the graphical user interface (GUI) that the user uses to interact with the system.

The logical separation is needed and used for security reasons, because the frontend does not have direct access to persistent data storage.

Backend

The backend system in a web application implements the business logic. It also provides the application with the needed access to persistent data storage, often implemented using some kind of database.

Frontend

The frontend implements the GUI needed for the user to interact with the web application. A web browser is ordinarily used for the access to the application. However the web browser itself does not implement the logic of the GUI. What it does is providing the infrastructure needed for a single-page application (SPA) to run in. The SPA is the web page which contains the code needed for the GUI logic used by the user when interacting with the web application.

API

An application programming interface (API) is used for sending of commands and data between the frontend and backend.

STUB and Simulator

A STUB and/or Simulator is used when testing the components separately from each other. They act as the “other” entity and responds to API requests in a predefined manner.

Document Object Model

The web browser keeps track of the web elements implemented by the SPA in an internal register, document object model (DOM). Hence different components such as buttons, text fields etc. that are available in the GUI, the locations and states of them is stored in the DOM.

Selenium WebDriver

Selenium WebDriver is an open source application that provides a programmable interface to interact with the web browser. The user can control various web elements over this interface, such as clicking buttons, entering text in text fields or validate results etc. I.e. automatically perform the tasks a user would do while using the web application.

The web browser stores the available web components in an internal register (DOM). It is the DOM that Selenium WebDriver interacts with to do the requested tasks.

The regression test activity of an SPA

Regression testing of an SPA is often divided into 3 different activities.

1. Testing of the frontend (GUI) using various techniques, where both the look and feel of the application is tested, as well as how to navigate, usability and clarity in the information presented. The testing is
2. Test of the backend (business logic) is normally done by triggering the API requests that the GUI would do. However as this is done without using the GUI, the problems with the navigation and slowness of the application running in the browser is avoided.
3. A limited test consisting of some test cases that tests that the two entities (frontend and backend) can communicate with each other and some scenarios are tested.

After you have described your project, you shall continue with writing a background to the area your project is in. Here you describe theories necessary to understand your project and explain terms you will use in the report.

Example: if you do a project that is about evaluating software architectures, you describe what a software architecture is, why it is important to design an architecture that suits a specific software system, methods for evaluating and comparing different architectures, etc.

1.2 Previous research

<Fix the missing Links on Monday>

<Check the tårta på tårta between chapters, now there is a lot of repetitive parts> Can some of these be removed? Some rephrasing would be needed.

<Add the first results and analysis of the implementation>

<Copy the new rearranged format for the lit study>

Testing is a well-researched area focusing on what to test, how to identify overlapping test cases, how to ensure that test cases are maintained and methods to keep them up to date and functioning have been addressed.

In the scope of what to test, different methods that can be applied to select the appropriate test cases to use are presented. Also different approaches to limit

the number of overlapping test cases and methods for test coverage has been researched.

The research of GUI testing is also covering multiple aspects of GUI testing, however with little coverage of the using of the GUI in regression testing.

Comparative research of different tools have been outlined in xxxx, and how creation of test cases with a GUI can be made is outlined in two main tracks:

- Recording and Playback
- Separate navigation from the flow

Recording and Playback also describes different methods that can be applied to make sure that the test cases also functions when the GUI or the navigation has changed in form of implementation of self healing test execution.

Most of the research is however done on smaller applications created specifically for the area of research instead of applying the research on commercially available systems.

< New research >

Previous research has focused on methods to limit the scope of the testing in various ways. Trial used have normally been done on smaller systems created for the purpose of the research. When GUIs have been used it is mostly a desktop application and not a web application that is used.

What is new in this report is the selected approach, to look at the possibility to combine test cases. Covering a larger scope of testing per used test case. And to research the problems that arises during the execution of the test cases rather than the problem of the scope of testing or the mass of test cases.

1. Studying the problems that arises during the execution of the test case
2. The study conducted on a commercially available web
3. Measures to address issues that arises are also tested and evaluated during the study

the problem by identifying the limiting factors, which

Here you briefly describe what others have done in the field or how others have attempted to explain or solve the same or similar problem as you are

investigating. It is okay to refer to tech articles and online blogs and portals, but you must also refer to published articles in the field. To find related articles, use the search tools listed [here](#).

1.3 Problem formulation

The activity of regression testing is testing that nothing has broken in the existing functionality of the application. However when the testing of an SPA is split into separate standalone activity the interoperability between frontend and backend components is not tested, or only partly tested.

This introduces a risk that interoperability issues are never found. The share volume of test cases also increases. The navigation performed in the GUI and the business logic provided in the backend results in partly overlapping test cases. The time that might be gained by regression testing the components separately comes with a cost in form of additional test cases and a risk that interoperability failures are never discovered during testing of the SPA

By studying and addressing the factors limiting the use of the GUI in regression testing of a commercially available web solution the intention is to show that the total amount of test cases needed can be reduced if the GUI is taken into use. The test coverage is also improved and fewer test cases are executed and maintained.

Here you give a detailed description of the research problem you intend to investigate. You can re-use the problem formulation from your project plan. You can read more about research problems [here](#).

1.4 Motivation

Upto 80% of the testing budget can be spent on regression testing. As the usage of the real GUI in driving the testing is difficult it is vital to understand the factors causing these difficulties. By addressing them with proper measures the need to divide the testing activity into separate entities (frontend and backend) decreases. Reducing partly overlapping test cases and increasing the reliability in the test results, where interoperability is also tested.

Here you motivate why your research problem is interesting for science, society or industry. You can re-use the motivation part from your project plan.

1.5 Research Question

RQ1.	What are the top 2 issues using Selenium WebDriver that cause longer test case execution times when test are run using
-------------	------------------------------------------------------------------------------------------------------------------------

	a browser, compared to testing the same functionality with APIs, and what can be done to minimize the time differences?
RQ2.	What are the top 2 issues using Selenium WebDriver causing flaky (unreliably) test results, and how can these be overcome?
RQ3.	What are the top 2 issues using Selenium WebDriver causing test cases to break when GUI modifications are done, and how can these be overcome?

The expected outcome of the project is knowledge about the main issues preventing the Web GUI from being used as driver in regression testing of the business requirements in a web based application. Thus preventing the whole application to be tested together.

The project also expects to propose measures to address the issues found and show the difference in the results when those measures are applied.

Your research problem is broken down into one or more research questions (RQ). RQs state exactly what you want to investigate in your thesis project. You have already defined research questions in your project plan. Copy and paste them here. You can read more about research questions [here](#).

RQ1	Research question 1...
RQ2	Research question 2...

You are also required to make statements about tentative and expected answers to your research questions (called propositions). What do you think your project will result in?

Don't mention anything about research method here. It will be covered in the next chapter.

1.6 Scope/Limitation

There are many different test solutions available, providing different kind of functionality. This project limits the use of test tools to Selenium WebDriver for communication with the web browser.

The report doesn't compare different web browser driving tools. The aim with the report is to understand the under laying factors that motivates testing the two entities separately.

It is assumed that a lot of different factors come in play, however the project focuses on the factors that are related to execution time of the test case, robustness and reliability of the test result.

There are a lot of different SPAs available on the market and this project uses one of them. Svenskaspels' implementation of a few of their betting games. Some of the findings made might not be applicable to other SPAs, as this depends entirely on the nature of the issues that are found.

The test execution is for sure expected to be slower when tests are run via a web browser compared to using direct API calls for testing of the business logic. However a decent "good enough" execution time is expected to be achieved.

You cannot solve everything. Here you describe what you do, and what you don't do, in your project. Limitations can for example be that you only compare some frameworks of all frameworks available on the market, that you only suggest an architecture for a specific software product and not a general architecture, or that you only include university students in a study and not a broader population sample.

1.7 Target group

The target group is Software Testers of web applications that faces similar issues and any test organization that wishes to enhance the confidence of their regression testing.

Here you outline which target group that might be interested in your thesis. If you, for example, do a project about software architectures, a target group can be professional developers and architects that work with similar software systems as the system you investigated.

1.8 Outline

In the following chapter the method used to approach the raised questions is outlined. The method describes both how data is collected, as well as how the proposed measures are applied.

The used implementation of the FIA application is described in the Implementations chapter, followed by a listing of the results collected using this application.

The proposed measures applied to address any issue are described under Analysis and Discussion, following by a short discussion regarding the results received when utilizing these measures.

Here you outline the rest of the report. It shall contain which chapters that will follow, and what each of them is about.

2 Method

In this chapter you shall describe the scientific approach that will be used to answer your research problem.

2.1 Scientific Approach

Here you define which formal research method(s) that shall be used to answer your research question(s). Research methods are divided into quantitative and qualitative methods. Quantitative methods result in numerical data, and qualitative in non-numerical data. You can read about different research methods [here](#).

2.2 Method Description

When you have defined which research method(s) that will be used to answer your research question(s), you shall continue by describing how the research methods will be applied in your project. If you for example shall conduct an experiment you describe your *independent* and *dependent* variables, how many times the experiment will be repeated, which software tools that will be used to run the experiment, etc. If you for example shall conduct a survey you describe how participants will be selected, how the questionnaire will be designed, how it shall be distributed to the participants, how data will be analyzed, etc.

2.3 Reliability and Validity

Here you discuss the reliability and validity of your project. You can read about reliability [here](#) and about validity [here](#). Discuss if you have any reliability issues or validity threats in your project here.

2.4 Ethical Considerations

You are required to discuss any ethical considerations (if any) in your project. If you do an experiment you will most likely not have any ethical considerations, but in a survey ethical considerations can for example be how you make sure that the privacy of the people participating in the study is not violated (by for example removing names from the gathered data).

3 Implementation

It is common that you will develop something in your project. It can be a mobile app, a stand-alone application, a website, a game, etc. In this chapter you describe the software you have implemented.

In some projects you don't develop anything, for example if you do a systematic literature review. In this case you remove this chapter.

4 Results

In this chapter you show and describe your results. You shall only show the raw results without any analysis, and you shall not put any conclusions or opinions in the description of the results. Try to be as objective as possible. An example of results from an experiment comparing five sorting algorithms is shown in Table 4.1 below.

Run	Bubble	Quick	Selection	Insertion	Merge
1	17384	24	3258	3	30
2	17559	21	3386	3	27
3	17795	19	3344	4	28
4	17484	20	3417	3	28
5	17642	19	3358	3	30
Average	17572.8	20.6	3352.6	3.2	28.6

Table 4.1: Execution times for the five sorting algorithms on 100 000 random numbers between 0 and 10 000.

What you show heavily depends on the type of research and what type of data you collect. Numerical data can for example be shown in both tables and graphs. A complementary graph for the sorting algorithms example is shown in Figure 4.1. For a questionnaire you can show the frequency (how many participants that selected the same answer) of each possible answer to a question.

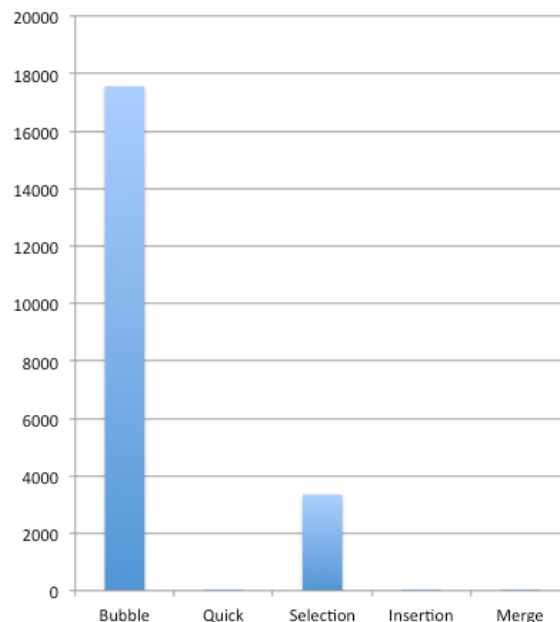


Figure 4.1: Execution times for the five sorting algorithms shown as a graph.

Note that Tables and Figures shall be labeled with chapter.number, for example Table 4.1 and Figure 1.6.

5 Analysis

Here you give meaning to and your own opinions of the results. What conclusions can you draw from the results? It is important that you don't draw any conclusions that cannot be backed up by your data. Consider using statistical tests to back up your claims. You can read about statistical testing [here](#).

6 Discussion

Here you discuss your findings and if your research question(s) have been answered. Think of research as a feedback loop. You define a problem, find a method of approaching it, execute the research and gather data. The data is then used to answer your research problem, thus creating the loop.

You shall also discuss how your findings relate to what other researchers have done in the area. Are your results similar to the findings in the related research you described in the Previous Research section?

This chapter is typically written in the present tense, while the previous chapters typically are written in past tense.

7 Conclusion

In this chapter you end your report with a conclusion of your findings. What have you shown in your project? Are your results relevant for science, industry or society? How general are your results (i.e. can they be applied to other areas/problems as well)? Also discuss if anything in your project could have been done differently to possibly get better results.

This chapter is also written in present tense.

7.1 Future Research

You cannot do everything within the limited scope of a thesis project. Here you discuss what you would do if you had continued working on your research project. Are there any open questions that you discovered during the project work that you didn't have time to investigate?

Here you shall include a list of all references used in your report. The reference list shall use the IEEE format. You can read about IEEE referencing [here](#). In the reference list below you can find examples of how to list a webpage [1][2], a journal article [3], a book [4] and a conference proceeding (article) [5].

References

[1] Linnaeus University. (2015) Course Room for Degree Projects. [Online]. Available: <https://mymoodle.lnu.se/course/view.php?id=5297#section-4>

[2] Monash University. (2015, Oct. 13) Citing and referencing: IEEE. [Online]. Available: <http://guides.lib.monash.edu/citing-referencing/ieee> ^[1]_[SEP]

[3] C. Lynch, “Big data: How do your data grow?” *Nature*, vol. 455, pp. 28–29, 2008.

^[1]_[SEP]

[4] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2010.

^[1]_[SEP]

[5] D. Agrawal, S. Das, and A. E. Abbadi, “Big data and cloud computing: current state and future opportunities,” in *Proceedings of the 14th International Conference on Extending Database Technology*, 2011, pp. 530–533.

[REF xxx] S. Biswas, R. Mall, M. Satpathy and S. Sukumaran Regression Test Selection Techniques: A Survey. Informatica 35 (2011), p. 289-321

<correct the missing link>

[REF yyy] P.K. Chittimalli and M.J. Harrold Recomputing coverage information to assist regression testing. IEEE Transaction of Software Engineering, 35 4 (2009), p.452-469.

<correct the missing link>

[REF zzzz] https://en.wikipedia.org/wiki/Application_programming_interface

[REF zzzz1] https://en.wikipedia.org/wiki/Single-page_application

[REF zzzz2] <http://www.seleniumhq.org/>

[REF zzzz3] <https://www.w3.org/TR/DOM-Level-2-Core/introduction.html>

[A2] Single-page application [Online]

https://en.wikipedia.org/wiki/Single-page_application



A Appendix 1

In the appendix you can put details that does not fit into the main report. Examples are source code, long tables with raw data and questionnaires.