**SOFTWARE TESTING AND QUALITY ASSURANCE PROJECT PROPOSAL**

**Title: CRM CODE TESTING**

**Team Members: Sayali Pendharkar (M08844056)**

**Sanjay Alexander Kalla (M08843598)**

**Instructor: Prof. Vignesh Subbian**

The goal of this project is to comprehend novel software testing tools and frameworks like RSpec, Selenium and Watir to test the system (SUT) and conduct software inspection on the same.

We have decided to perform these operations on ‘Fat Free CRM’ [1] which is our system under test – It is an open source, customer relationship management (CRM) platform built using Ruby on Rails framework.

About the Software under Test - Fat Free CRM is an open source customer relationship management platform. It finds its application mostly in small or medium sized businesses. It can run on windows, Mac and Linux. It can be installed with complete features at no cost. It allows users to manage leads, accounts, contracts, quotes and sales pipelines. The program comes with a number of benefits that provide users with a large range of options without spending anything. [2]

Fat Free CRM provides many benefits for web-based management. It allows users to send mass emails, create tickets and manage meetings all in one program. The software also provides customized access depending on each user’s role.

The system has approximately 29500 lines of Ruby code, 4370 lines of HTML code, 2601 lines of CSS code and 10000 lines of JavaScript code.

Our team has decided to choose this as our term project due to its fast growing popularity as well as to learn something new and to test it. Furthermore, with this we as a team are also trying to create a professional environment wherein there are new challenges every day for a test engineer. We are basically going to do the following:

Software Testing -

1. Conduct manual software inspection on Dashboard, Tasks, Leads and Users module.
2. Use a static analyzers such as Rubocop [5] to perform static analysis on the SUT and the results will be further analyzed to determine any correlation with results from manual inspection.
3. Use Metric Fu, a compilation of several tools to help find areas of code that could be improved [7].
4. Code Coverage - We intend to use SimpleCov [3] to determine, analyze, and interpret structural coverage of the system.
5. We have precisely generated automated test cases and results using Selenium, which is a suite of tools to automate web browsers**.** Our focus was to test the GUI and present a concise analysis of the same. We used Capybara which is acceptance test framework for web application. Moreover, it also gives a feel of simulation experience for real user.

Quality Assurance -

1. Rack-Mini-Profiler [5] is a tool used for finding bottlenecks of your applications. It does a live speed analysis of how long it took for the request to be processed and how much of that time it was doing various renderings, database queries and DOM loading.
2. We will perform analysis on the Github commits to generate repository statistics using GitStats [6].

**References -**

[1] **Github Link:** <https://github.com/fatfreecrm/fat_free_crm>

[2] **Guide for the application:** <http://guides.fatfreecrm.com/>

**Demo for the application:** <http://demo.fatfreecrm.com/login>

**Login Details:** username: aaron; password: aaron

[3] **SimpleCov:** https://github.com/colszowka/simplecov

[4] **Selenium:** <http://www.seleniumhq.org/>

[5] **Quality Assurance Tools:** <https://infinum.co/the-capsized-eight/articles/top-8-tools-for-ruby-on-rails-code-optimization-and-cleanup>

[6] **Gitstats:** <http://gitstats.sourceforge.net/>

[7] **Metric\_fu:** <https://github.com/metricfu/metric_fu>