



UNIVERSITEIT VAN PRETORIA  
UNIVERSITY OF PRETORIA  
YUNIBESITHI YA PRETORIA

Faculty of Engineering, Built Environment and  
Information Technology

---

# System Requirements Specification

## Cerebero



---

Frederick Ehlers	11061112
Jacobus Marais	15188397
Rikard Schouwstra	15012299
Victor Twigge	10376802

---

## Stakeholders

---

Computer Science Department  
of University of Pretoria:

Vreda Pieterse

---

eCivix

Daniël Eloff Chairperson

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Testing</b>	<b>2</b>
2.1	Mocha . . . . .	2
2.1.1	Why Mocha? . . . . .	2
<b>3</b>	<b>Reporting</b>	<b>2</b>
3.1	Reporting . . . . .	2
<b>4</b>	<b>Appendix</b>	<b>2</b>
4.1	Github . . . . .	2
4.2	Trello . . . . .	2

# 1 Introduction

We Cerebero are working together with eCivix to create a new web based game. The idea of the game is to create an election simulator to teach High school students how elections work and what their vote essentially means in the greater scheme. The user will create party that they will control. The game will revolve around the party gaining funds and man power to do campaigns and gain more funds and man power to run bigger and more effective campaigns. The user with the score at the end of the game wins. The user will be playing against an AI (Artificial Intelligence) player that we will program. The AI will try be more effective/ more successful than the user. There will be a leader-board with all the users' scores and at the end of the client's event a winner will be chosen for a prize.

## 2 Testing

### 2.1 Mocha

#### 2.1.1 Why Mocha?

1. Testing the AngularJS front end we will be making use of the testing framework called [Mocha](#). We choose Mocha as it runs on NodeJS and uses JavaScript to do the tests, which to our benefit because of the fact that AngularJS also works with JavaScript.
2. This will then allow us to test our database Create, Read, Update and Delete operations that are carried out from the front end to ensure they are successfull and fail when they are supposed to.
3. Mocha is strong but also deverse since we can utilise Mocha to do code coverage as well. Thus allowing us to see how well our tests cover all our code, only then we will truly know if we are testing for everything.
4. There are numerous other reasons you want to use Mocha to improve our code:
  - (a) It works with Continues Integration. Therefore our CI will not crash with Mocha tests.
  - (b) Supports node debugging
  - (c) Support for extensible reporting
  - (d) Has copious amounts of documentation and support available
  - (e) Much more benefits, feel free to see [Mocha](#) for more benefits.

This will provide us with a high quality product due to powerful technology of Mocha and thus quality of the test and their results will pass all quality expectations.

## 3 Reporting

### 3.1 Reporting

## 4 Appendix

### 4.1 Github

[Github](#)

### 4.2 Trello

[Trello](#)