ClassMate

Test Plan

# Table of Contents

[Table of Contents](#h.hi8r4vqg1i9x)

[Introduction](#h.x15xppjvyla)

[Required Resources](#h.8wyzz5lq9w4i)

[Hardware](#h.f020qdwsyqj6)

[Software](#h.7bjkwrp3vodu)

[Frameworks](#h.l4ghcl5vy9zi)

[Testing](#h.wahievo62gwz)

[What Will Be Tested](#h.czz4ganx5vab)

[Potential Future Tests](#h.xe8tk0f0ediz)

[Results](#h.o9plmm1zrwc)

# Introduction

*ClassMate* is an application developed by Justin Coschi, Kyle Zimmerman, and Sean Coombes as the solution to the team project assignment in Conestoga College’s PROG 3180 - Mobile Application Development course. *ClassMate* is a way for students to organize all of the assignments, tests, and exams received in any of their courses and have a single timeline of due dates. This document outlines the required resources and test cases used for the development and deployment of *ClassMate*.

# Required Resources

## Hardware

The hardware required for the development of *ClassMate* are:

* Computer for software programming
* Mobile device (Android) for deployment testing
* Media cables to connect development environment and target platform (optional)

## Software

The software required for the development of *ClassMate* are:

* A text editor, preferably capable of basic HTML, CSS, and JavaScript markup distinction
  + i.e. SublimeText, Visual Studio
* Google Chrome browser to test functionality during development
* Server or emulator to test deployment

## Frameworks

Frameworks required for the development of *ClassMate* are:

* jQuery
* jQuery Mobile
* jQuery Validation
* WebSQL
* Ghost Inspector/Selenium
* NUnit

# 

# 

# Testing

## What Will Be Tested

The test plan will test the user’s inputs. These include:

* Registering a user
* Logging a user in
* Creating a course
* Creating an event
* Logging a user out

## Potential Future Tests

In the future, we may need to test other requirements. These include:

* Viewing correct details of a course
* Viewing correct details of an event
* Up-/Downvoting

# Results

The results of our testing, using the NUnit framework, is shown below.

