**Version History**

|  |  |  |
| --- | --- | --- |
| Date | Name | Reason for Changes |
| 01/05/2013 | Dominic Lee | First version of document |
| 05/05/2013 | Delvin Varghese | Added critical analysis |
| 09/05/2013 | Dominic Lee  Delvin Varghese | Corrected details added “Other Aspects” |

# Critical Analysis

This document lists some of the limitations we faced while undertaking this project. While the key aspects of the project were completed successfully, it was not without its difficulties. One of the main difficulties we faced was our workforce itself. The original team consisted of 5 members. However, two members dropped out subsequently without ANY input whatsoever.

One of the requirements of the project was access to a Mac/UNIX system. We initially fulfilled this requirement as a member of our team had a Mac. Also our design methodology was based on equal work from all members of team so we allocated 2 members to the applet, 1 member to the server and 2 members to the communication gateway (i.e. Kannel). However as both these members dropped out (and one of these members had the Mac) we suffered a substantial setback. Despite this, we had a meeting with our client, and we were generously helped by Simulity to use some pre-set configurations which helped us from stretching ourselves, making up for the lost team members.

Another factor to be taken into consideration is that all of the software / hardware we used belonged to Simulity.

Examples of this would include a virtual device used throughout the company to test parts of code. This meant that we had to be careful with every instance of it we copied and were careful to remove all copies now that the product is finished.

So due to security considerations and so as not to cause any harm to their business, we were restricted (and rightfully so) in our access to the server. Originally, the plan was to implement our own server but as previously mentioned, because of the mitigating circumstances regarding the workforce, Simulity allowed us to use their server, which we could deploy our web service to with little modifications. We were also given a proprietary software owned by Simulity which they use to format the sim cards to industry specifications. Again due to licencing reasons, we are unable to reproduce it here.

**Other Aspects**

There was a very steep learning curve for us having to learn Java Card as well as learn the industry specifications. Learning the language Java Card was the first aim we had. It was difficult because we had problems with finding helpful examples of Java Card code to learn from as it is a lot less frequently used than the basic Java. The reason we needed to learn JavaCard was because it is a language in its own right, this is because the only allowed variable types are short, bite and Boolean. This is because of 2 reasons, the first being the limitations of the language the second being limitations of the embedded device itself. The reason the language limits the usable variables is to increases efficiency and the device limits the usable variables because of architectural limitations.