them off the feature list makes it easier to see the amount of progress done and thus motivates greatly for further effort.

2.2 Modifications to the Development Lifecycle

Even though feature-driven development seemed to be the best possible development process for my project, I still felt I had to modify it slightly, especially due to the time frame in which I had to finish my application.

As mentioned above, FDD focuses on writing the application feature by feature until the list of features has all of them marked as 'finished'. Each of those features has six 'milestones' – states in which a project may be – and which are supposed to help keep track of the progress made throughout the whole project. However, in my opinion using this in my project would not be beneficial for me at all as I am the only person developing the application – thus I will always be informed about the progress I made and the amount of work still to be done. With this in mind, I decided to omit this part of FDD and simply create a list of features which I would mark as finished as I went through it.

It is also important to clearly state the development strategy I chose. As in most projects, and also in this one, the most important objective was to create a working piece of software; this was my goal as well. Thus, I had to focus on developing the most important features of my application first – with "the most important" being those which would prove that the application actually exists, can be run and "does something". This means that my strategy was to start working on the most basic features first, and attempt to add flavour later, depending on the time left until the hand-in date of the project, and not the other way around. Therefore, if I stayed behind the schedule, I would still most probably have at least part of the program working as intended.

2.3 Tools

The below sub-sections list the tools which were used when developing this project. Each of these comes with a short description and a justification for the use of this tool specifically as well as how the tool would improve the process.

2.3.1 Programming Language

Creating an application for the Android operating system does not give one any choice of the programming language to be used as the only one that actually can be used is Java. However, it is not as simple as it may sound: in order to run a program under the Android, its code has to use the Android framework which does not have exactly the same API as the Standard Edition of Java – the reason for this being the Dalvik Virtual Machine used by Android to run the applications instead of the Java Virtual Machine. In fact, I would risk stating that it could be described as learning a different programming language with the same syntax.

2.3.2 Integrated Development Environment

Developing an Android application requires the use of Android SDK which works quite well paired with the Eclipse IDE – thus it seemed only straightforward to choose this IDE over any other. Moreover, Eclipse provides a means of creating a user interface for