Chapter 1

Approach

In order to research the problem formulation of this paper, a practical approach has been used. To evaluate and compare the two different methods (as described in problem formulation) a mobile application with the same requirements has been developed in each method. Both of the mobile applications make use of the same web application.

1.1 The mobile applications

The web application that is used composes of a single page and is functional for the web browser Google Chrome. The page displays a form with a text field, an image field and a location field. To get the image or the location value for the form the user can only press a button. When the image or location button is pressed the web application uses the Google Chrome's built in ability to take a picture or get the location.

The mobile applications consists of two parts. The web application described above and the mobile application encapsulating the web application. The mobile application must encapsulate the web application and display the web application exactly the same as it is displayed in a browser. The mobile application must use the mobile's native functions to get the value for the image and the location.

The text field in the web application form must function the same way as it functions in Google Chrome.

For getting the image value in the form the mobile application must give the user the option to use an existing image from the image gallery or take a picture using the camera on the phone. The image that is passed back to the web application must be of in Base64 format.

The location must be obtained using the mobile's GPS.

When the user interacts with the mobile application the interaction must be directly with the web application encapsulated within the mobile application.

In the project the following was used for developing and testing:

- Android SDK version 21
- Apache Cordova CLI version 4.0
- Google Chrome version 44.0.2403.157
- Nexus 5
- Android version: 5.1.1
- Kernel version: 3.4.0-gbebb36b

1.2 Measuring lines of code

The resulting mobile applications are measured using the software metric logical lines of code. The logical lines of code have been counted including and excluding the following:

(here we have a bullet list of what's included and what's excluded)

The mobile application are written in two different programming languages, Java and JavaScript. To compare the measurement result of logical lines of code we use the conversion table described by Galorath and Evans[**?**, p. 163]. JavaScript and Java is both third generation languages and therefore no conversion is needed.

The logical lines of code written in the web application layer is also measured in both developing methods. To give an correct result we measured the added logical lines of code that has been written as a result of the mobile application. The resulting lines in the mobile and web application layer is added to give a total result.

The architecture of the mobile applications impact the amount of lines of code. The measurement result is therefore analysed using the measurement data and our personal development experience.

1.3 Layer communication

A goal with the mobile application is to have a developer friendly way of communicating between the mobile and web application layer. The web and mobile application has a master and slave relationship. Where the web application layer acts as the master and the mobile application layer acts as the slave. To evaluate this communication a flow diagram of the web layer sending a request for native data and the mobile application layer responding with data is constructed. The diagram acts an suggestion in how developer friendly the communication is and is combined with our personal developing experiences.