Project Proposal

# Description

## Background

With the emergence and proliferation of mobile computing tools, businesses are seeking ways to make good use of the technology for the benefit of their businesses. One of the ways in which this is being done is the collection of data by mobile devices.

Many companies need to collect data from remote locations. Examples of such data collection would be client information collected by salesmen, technical consultants who wish to transfer statistics back to head office as soon as possible, and technicians filling out risk assessments for their Health and Safety department.

According to Tomlinson et al., (2009) who researched the use of mobile phones for data collection, electronic data collection is preferred over paper-based collection. This is because “errors are frequent, storage costs are prohibitive, and the costs of double-data entry are high”(Tomlinson, Solomon et al. 2009).

## Problem

In some job roles (in particular, field-based roles) it is important to collect and return information to management. Such information may be as a result of Health and Safety rules, stock recording, ad-hoc checks on any particular area, etc. Often such data collection is achieved on paper sheets and transferred to digital form. This transfer may be done by transcription, or scanning. As the collector of the information may not be the processor of the information, the information is likely to be forwarded to another person. If there are many different forms, there may be a number of unrelated people processing the received information.

A number of issues can, and do arise out of the scenario described above:

1. Use of paper forms increases the risk of transcription errors and legibility problems when entering the data into a computerised system.
2. Scanning requires the availability of related equipment and optic character recognition (OCR) software. It is often the case that forms are sent to a central office and someone must scan those forms.
3. Since data is collected and processed by different parties, management may miss opportunities to see trends and uncover important data that is only evident when all received data is analysed.

## Proposed solution

This project looks at solving the above problem for the following scenario.

A company has field engineers from whom various types of information must be collected. Each engineer is in possession of an android tablet. Form questionnaires need to be filled in on the tablets and returned to management. The forms do not have a determined structure, so one engineer may have a number of different forms to fill in and send back to management.

Management keeps a store of received forms. The forms can be viewed as PDF files or other suitable format, and information pertaining to any group of data can be analysed. The type of analysis is not predetermined but it must be possible for any reasonable type of query to be done on all feedback forms. Therefore, management should be able to select attributes that are common to more than one form. For example, if two different forms include a customer location field, they should be compatible, and if at all possible, have the same source of values.

Engineers do not always have an available Internet connection. As a result, a form template must be saved on the tablet and any data entered must be saved, at least until an Internet connection is available and can then be sent to management.

Security is not a major concern but should be implemented if possible, to protect information that is sent.

The application can be thought of as a combination of the following components:

* Desktop management application
* Main data storage
* Android client app
* Android client data storage
* Web service

# Desktop Management Application

This application will be used to:

* Create form definition files
* Create data store for each form definition file
* Send files to participating engineer’s tablets
* Remove expired forms from tablets
* Query data

# Main Data Storage

This data store will contain the data collected from each used form. This will require a table for each type of form. Some field may be selected from a permanent collection of tables. For example, a list of customers which are common across the business may exist in a permanent table which can be included on a form definition file.

# Android Client App

The client app performs the following:

* Receives and stores form definition files
* Renders the form definition files to the screen
* Saves filled out forms to storage
* Sends filled out forms to the Desktop Management Application

## Android Client Data Storage

This component is responsible for storing the definition files as well as data from filled out forms.

## Use-Case Diagram

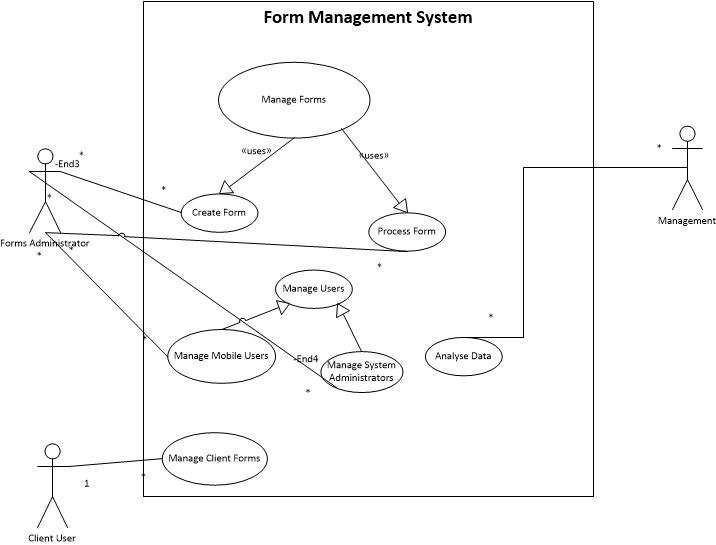


Figure : Use case diagram for form management system

## Use Case Descriptions

|  |  |  |
| --- | --- | --- |
| Use Case Name: Manage Client Forms | ID: 1 | |
| Primary Actor: Client User | |  |
| Description: This describes the process for maintaining and using forms on the mobile device. | | |
| Trigger: Client user launches client application to download forms, fill in a form, or send a filled-in form. | | |
| Relationships: Client User | | |
| Normal Flow of Events: | | |
| 1. User starts client application    1. If report definition file updates are available, user starts synchronisation 2. User selects form definition 3. User fills out form 4. User sends form | | |
| Alternate flows:  4a1. User saves form  4b1. User sends saved form  4a2. Form saved if no Internet connection  4b2. Form sent by user when Internet is available | | |

# Form Structure

A form definition file is used to provide information that is required to dynamically build a form screen on the android client device. The form definition files must contain entities for each form and each question. The entities of each form must include:

* Form identifier
* Form version number
* Form Title
* Form Description
* A number of sections, each of which contains one or more questions

The entities for each question must include:

* Question text
* Question type
  + Multiple choice
  + Multiple selection
  + Free text
  + Numeric
  + True/False
* Answers for multiple choice, and multiple selection question types

The responses to each form are saved in a separate response file which is used to send back to the data server. When a user wishes to continue editing a saved file, the response file is used to repopulate the form. The response files contain the following:

* Form identifier
* Form Version number
* Completion date
* Return Date
* User’s identifier
* Responses

## Form Definition File Format

There are a number of formats that can be used to transmit files. These can fall into two categories; human-readable and binary/non-human-readable. To simplify the design, the file format will be human-readable. Two possibilities to be considered are:

* JSON
* XML

XML: XML is a format that has been used extensively for data interchange in many applications. The design goals of XML include easy use over the Internet, ease in writing programs which utilise it, human-legibility, ease of creation, and concise format (Bray, Paoli et al. 1997).

JSON: According to Lennon, “JSON is an open, text-based, human-readable data interchange format derived from JavaScript” (Lennon 2009). Lennon also writes that JSON has a syntax that programmers of some modern languages would be familiar with, such as Java and C++ (Lennon 2009). The syntax of JSON is more compact than XML.

One of the main differences between JSON and XML is the speed of transmission of data. This is due to the compactness of the JSON syntax over the XML syntax. In research carried out by Nurseitov, Paulson et al., It was found that JSON has a lower transmission time per object than XML and uses less CPU resources (Nurseitov, Paulson et al. 2009). The difference between transmission times are fractions of milliseconds and may make little difference to each client app sending and receiving a form definition file. However if this was to be scaled to thousands of employees this could become significant. It would be important to consider the future deployment of this proposed application to include serving multiple companies. In such a scenario speed of transmission could become quite important.

One advantage that XML has over JSON is familiarity. The use of XML is so common that most developers are familiar with it; at some point in a developer’s career there is likely to have been exposure to HTML (hypertext mark-up language), AJAX (Asynchronous JavaScript and XML) and some need to transfer data between remote or disparate systems. XML has become a standard choice for data interchange for many people. However, with the ease of use of JSON, this advantage may not be much of an issue.

It should also be noted that there is support for JSON and XML built into the Android platform.

# Bibliography

Bray, T., J. Paoli, C. M. Sperberg-McQueen, E. Maler and F. Yergeau (1997). "Extensible markup language (XML)." World Wide Web Journal **2**(4): 27-66.

Lennon, J. (2009). Introduction to JSON. Beginning CouchDB, Springer**:** 87-105.

Nurseitov, N., M. Paulson, R. Reynolds and C. Izurieta (2009). "Comparison of JSON and XML Data Interchange Formats: A Case Study." CAINE **2009**: 157-162.

Tomlinson, M., W. Solomon, Y. Singh, T. Doherty, M. Chopra, P. Ijumba, A. C. Tsai and D. Jackson (2009). "The Use of Mobile Phones as a Data Collection Tool: A report from a household survey in South Africa." BMC Medical Informatics and Decision Making **9**(51).