Software Requirements Specification
Forensic Medicine Mobile Application

Version: 1.0

Organization:
University of Pretoria: TCP Solutions

GitHub:

https://github.com/CollenMphabantshi/TCP-Solutions

Authors:
Mr Mphambantshi C (10404687)
Mr Legodi PT (29302732)
Miss Sikhitha TP (10346504)

May 22, 2014

Name	Date	Changes
TP Solutions	14 May 2014	Vision and Scope
TP Solutions	15 May 2014	Addition to vision and scope and quality requirements.
TP Solutions	16 May 2014	Software Architecture
TP Solutions	23 May 2014	Final Version

Contents

1 Overview

This document provides the overall vision and scope of the Forensic Medicine Mobile Application project. It explains and illustrates what the system will do and look like. This document basically provides the skeleton of our project. It includes the scope limitations and exclusions which will help guide the stakeholders on what is expected and not expected. This document also include use case diagram which will help explain and show the whole system.

1.1 Document conventions

Documentation formulation: LaTeX

Unified Modelling Language: version 2.0

2 Vision and Scope

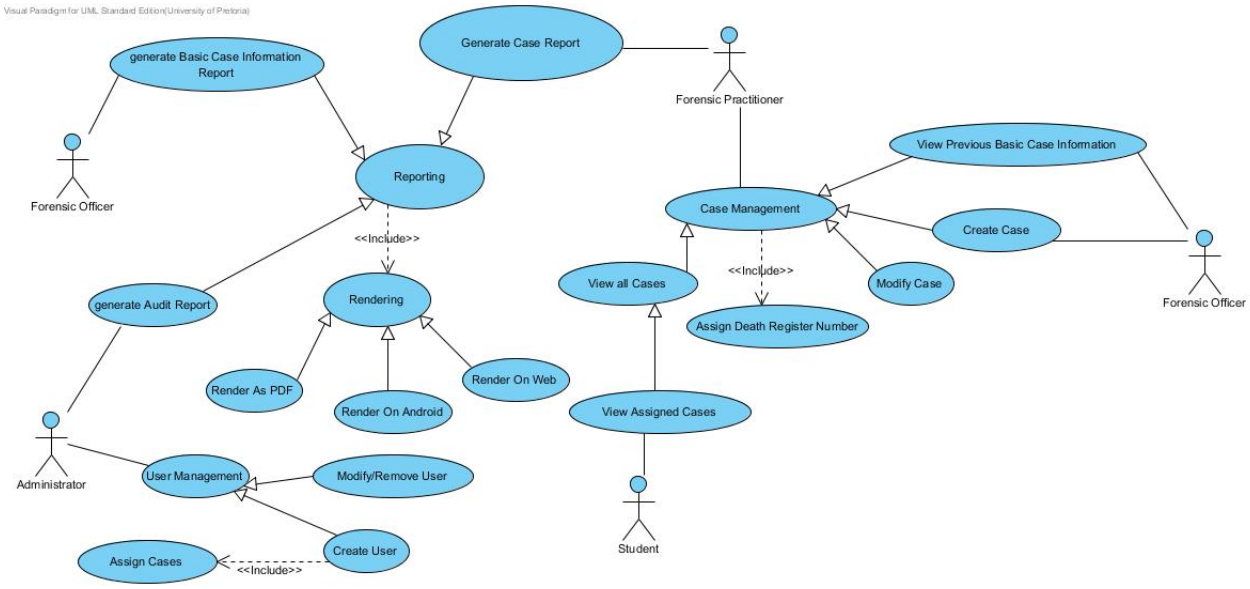

Figure 1: The Scope of the system

The proposed system is the death scene register that allow:

- Forensic officers to:
  - Capture data from death scene – the FO's will gather information on every scene based on the template it has on the mobile application.
  - View basic information. The FO views personal details of the deceased and police officer who was at the scene.
- Forensic practitioner to:
  - Generate reports – FP's will generate web, android and pdf reports specifically to their needs e.g. generate report of all hanging cases 2014.

  - View all cases – every scene stored on the database they should be able to view them.
  - Edit case information. If there was any errors made on the form such as spelling errors FPs should be able to correct them.
  - Manage cases. FPs will dictate if the case is natural and non-natural death and do other functionalities.

- Students to:
  - View all the cases cleared to them – this is for research purpose only.

- Administrator to:
  - Add new users.
  - Remove users.
  - Edit users – change personal details and access rights.
  - View audit report.

3 Scope Limitations and Exclusions

Pictures that demonstrate how the incident happed are excluded on this phase, maybe they can be added at a later stage.

4 Architecture requirements

4.1 Access channel requirements

It is going to be accessed by humans using android and web application.

4.2 Quality requirements

- **Performace**
  - The system should process all the reports within 10 seconds.
  - It should send the information to the server within seconds.
- **Reliability**
  - The system should be up and running all the time.
  - Easy and fast access to the database.
- **Scalability**
  - The system should be able to handle all death scenes captured information.
  - It should allow additional templates.
- **Security**
  - The system is accessible to users who are authorized.
  - System users will have different permission.
  - Information about death scenes will be encrypted.
  - Information stored by forensic officers will not be edited after the submission.
- **Flexibility**
  - If something happen when the forensic officer is capturing information, it should automatically be stored in the server.
- **Maintanability**
  - The system will be maintained every time the client needs new changes.
- **Auditability**
  - The system should record all the changes made to the data stored, by showing whom, when and what was changed.
  - It will also show old and new values.
- **Usability**
  - Users should be able to use the system without prior training.
  - The system will be in English.

4.3 Integration requirements

- Database will be created from scratch.
- The android application will be connected to the web service and the web service connected to the server.

4.4 Architecture constraint

- The device that will be used is Asus nexus 7
- Android SDK
- MySQL
- HTML5,PHP,apache(Afrihost)
- Java, JavaScript
- Ajax, jQuery
- The mobile client must be running on an android application.

5 Software Architecture Documentation

5.1 Architecture requirements

1. Architectural scope

   The database will run on Afrihost

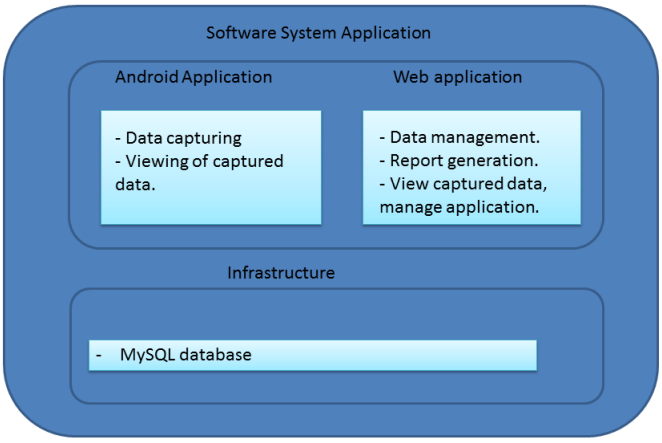

Figure 2:Architectural Scope

Android app
- The android application will be used to capture information on death scenes.
- View information based on clearance.

Web app
- will be used for data management.
- Report generation.
- System administration.

Infrastructure
- Data storage, MySQL database on Afrihost.

2. Quality requirements

Security
- Only authorised people should be able to have access to the system.
- Only administrator should register people.

Auditability
- Any change made to data stored should be recorded.
- Record what, who and when changes were made.

Performance
- Data should be sent in real time e.g. from forensic officer to forensic practitioner should receive it within 10 seconds.

Reliability
- The server should run all the time (24/7 - 365) and the connection should always be active.
- Only administrator should register users.

Usability
- All the users should be able to use the system without any prior training.

3. Integration and Access channel

   Access Channel • Accessible by humans through the following channels: Admin – the access it through web application and FO, FP and students – they show access the system through mobile application.

   Integration Channel • The new SQL database will be created in Afrihost.

4. Architectural constraints

   The system will use the following constraints • Android SDK
   - REST web services
   - The system will be deployed in Asus Nexus 7 OS Android 4.1 jelly bean.

   Technologies to be used • Java, PHP, HTML, JavaScript, MySQL, JQuery, CSS

5.2 Architectural Pattern

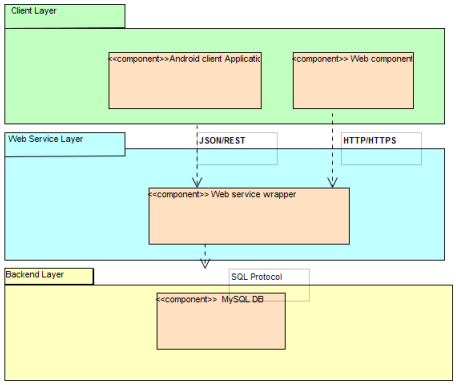

Figure 3: Architectural layered pattern

The Architectural pattern • Provides access to humans - Client
- Provides Functionality and objects required to client layer -¿ Web service
- Host database - Backend layer

The communication protocol are also shown. They include • HTTP/HTTPS from the browser to the web module.
- Provides Functionality and objects required to client layer - Web service
- JSON/REST/HTTP/HTTPS for the web services between the Android application and the database.

5.3 Use of reference architectures and frameworks

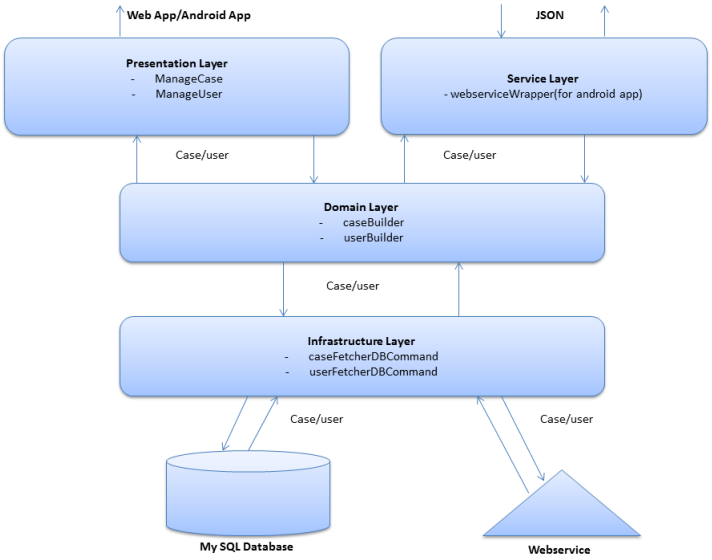

Figure 4: Three Layer Architecture

Presentation Layer • This User interface Layer. The UI is responsible for creating and displaying the user interface and handling user interaction. It's going to be in Android App and Web App. It gets data from Domain layer.

Service Layer • This is the Web Service Layer. Responsible for showing web service API and returning method results as JSON. It gets data from Domain Layer.

Domain Layer • This is the Business logic, it is responsible for business logic of the application. All functions and objects used are going to be modelled here. It gets data from Infrastructure layer.

Infrastructure Layer • It responsible for querying database, calling web service and send emails.

6 Functional Requirements

6.1 Introduction

6.2 Required functionality

6.3 Use case prioritization

6.4 Use case/Services contracts

7 Open Issues

8 Glossary

Forensic officer (FO) – a specially trained crime scene officer that collects the finding evidence that will be analyzed back at the lab by forensic scientist or forensic practitioner.

Forensic practitioner (FP) - also referred to as crime scene investigators and forensic science technicians examine pieces of evidence to provide crucial support in criminal investigations. Their professional expertise is sought in laboratories, crime scenes and courtrooms.

Stakeholders - is anybody who can affect or is affected by an organization, strategy or project. They can be internal or external and they can be at senior or junior levels.

Students – honors and masters students who are doing research as part of their studies.

MySQL MySQL (Structured Query Language) is an open-source relational database management system.

PDF Portable Document Format is a file format for capturing and sending electronic documents in exactly the intended format.

UML Unified Modelling Language is a general-purpose modelling language in the field of software engineering. It provides a set of graphic notation techniques to create visual models of object-oriented software-intensive systems.