

# **Functional Requirements(uWatch digital forensic tool)**

**Group Name:** MPHETamines

Martha Mohlala	10353403
Taariq Ghoord	10132806
Phethile Mkhabela	
Sboniso Masilela	
Harrison Maphuti Setati	12310043

**Git repository link:**  
[https://github.com/MPHETamines/  
MPHETamines/](https://github.com/MPHETamines/MPHETamines/)

**Date:** 20 May 2015

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Purpose . . . . .	1
1.2	Scope . . . . .	1
1.3	Background . . . . .	1
1.4	References . . . . .	1
1.5	Assumptions and Constraints . . . . .	1
1.5.1	Assumptions . . . . .	1
1.5.2	Constraints . . . . .	1
1.6	Document Overview . . . . .	2
<b>2</b>	<b>Methodology</b>	<b>2</b>
<b>3</b>	<b>Functional Requirements</b>	<b>2</b>
3.1	Context . . . . .	2
3.2	User Requirements . . . . .	2
3.3	Use Case Diagrams . . . . .	3
3.4	Class And Sequence Diagrams . . . . .	4
3.5	Activity Diagrams . . . . .	4
3.6	Functional Requirements . . . . .	4

# **1 Introduction**

[Provide an overview of the system and some additional information to place the system in context.]

## **1.1 Purpose**

[Provide an overall description of the FRD, its purpose. Reference the system name and identifying information about the system to be implemented.]

## **1.2 Scope**

[Discuss the scope of the document and how it accomplishes its purpose.]

## **1.3 Background**

[Describe the organization and its overall responsibilities. Describe who is producing the document and why.]

## **1.4 References**

[List references and controlling documents, including: meeting summaries, white papers, other deliverables, etc.]

## **1.5 Assumptions and Constraints**

[Provide a list of contractual or task level assumptions and/or constraints that are preconditions to preparation of the FRD. Assumptions are future situations beyond the control of the project, whose outcomes influence the success of a project.]

### **1.5.1 Assumptions**

Examples of assumptions include: availability of a technical platform, legal changes and policy decisions.

### **1.5.2 Constraints**

Constraints are boundary conditions on how the system must be designed and constructed. Examples include: legal requirements, technical standards, strategic decisions. Constraints exist because of real business conditions. For example, a delivery date is a constraint only if there are real business

consequences that will happen as a result of not meeting the date. If failing to have the subject application operational by the specified date places the organization in legal default, the date is a constraint. Preferences are arbitrary. For example, a date chosen arbitrarily is a preference. Preferences, if included in the FRD, should be noted as such.]

## **1.6 Document Overview**

[Provide a description of the document organization.]

## **2 Methodology**

[Describe the overall approach used in the determination of the FRD contents. Describe the modeling method(s) so non-technical readers can understand what they are conveying.]

## **3 Functional Requirements**

### **3.1 Context**

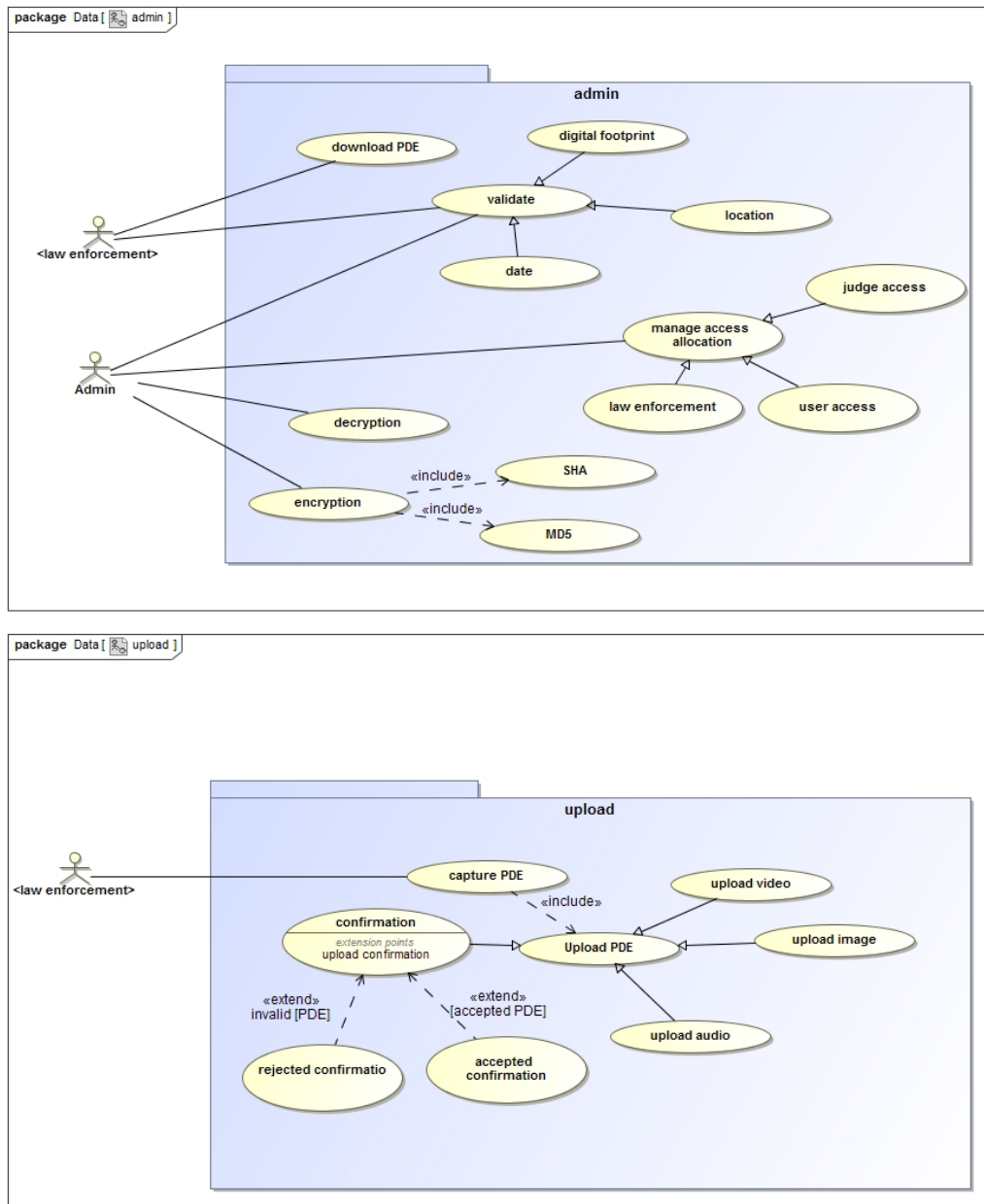
[Provide a context diagram of the system, with explanations as applicable. The context of a system refers to the connections and relationships between the system and its environment.]

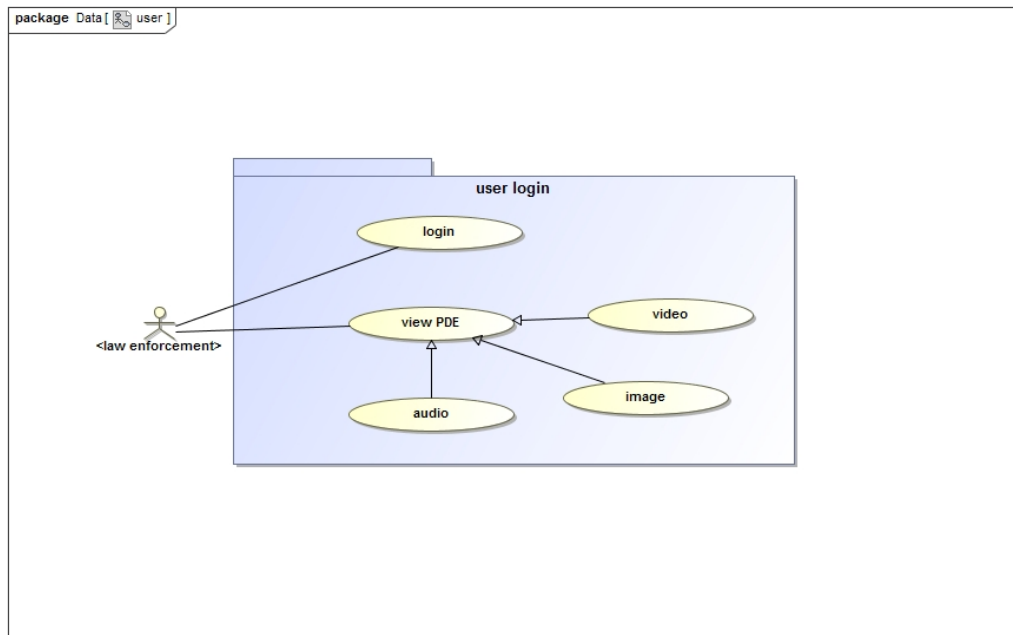
### **3.2 User Requirements**

[Provide requirements of the system, user or business, taking into account all major classes/categories of users. Provide the type of security or other distinguishing characteristics of each set of users. List the functional requirements that compose each user requirement. As the functional requirements are decomposed, the highest level functional requirements are traced to the user requirements. Inclusion of lower level functional requirements is not mandatory in the traceability to user requirements if the parent requirements are already traced to them. User requirement information can be in text or process flow format for each major user class that shows what inputs will initiate the system functions, system interactions, and what outputs are expected to be generated by the system. The scenarios should be comprehensive, to the extent that all user types and all major functions are covered. Give each user

requirement a unique number. Typically, user requirements have a numbering system that is separate from the functional requirements. Requirements may be labeled with a leading U or other label indicating user requirements.]

### 3.3 Use Case Diagrams





### 3.4 Class And Sequence Diagrams

[service contracts and their corresponding sequence diagrams to be shown here]

### 3.5 Activity Diagrams

[flow of the system in different contexts]

### 3.6 Functional Requirements

[List the functional requirements of the system.]