



# **LocAdoc Database Design Document (DDD)**

## **Version 2.0**

**Prepared by: Abhi Jay Krishnan**

**Kim Hyeocheol**

**Rivaldo Erawan**

**Durrah Afshan**

---

## Revision History

<b>Date</b>	<b>Version</b>	<b>Description</b>	<b>Author</b>
27/09/2017	2.0	Made modification to the structure of both NoSQL database and SQLite database.	Abhi Jay Krishan

# Table of Contents

<b>1</b>	<b>INTRODUCTION .....</b>	<b>1</b>
1.1	DOCUMENT OBJECTIVES .....	1
1.2	INTENDED AUDIENCES .....	1
1.3	REFERENCES .....	1
<b>2</b>	<b>DETAILED DATABASE DESIGN .....</b>	<b>2</b>
2.1	DYNAMODB DESIGN (NOSQL DATABASE) .....	2
2.1.1	<i>Object Diagram</i> .....	3
2.1.2	<i>Data dictionary</i> .....	4
2.1.2.1	Data dictionary for Element: User.....	4
2.1.2.2	Data dictionary for Element: Password.....	4
2.1.2.3	Data dictionary for Element: File.....	4
2.1.2.4	Data dictionary for Element: Area .....	5
2.2	SQLITE DATABASE DESIGN (RELATIONAL DATABASE) .....	6
2.2.1	<i>Conceptual diagram</i> .....	6
2.2.2	<i>Description</i> .....	6
2.2.3	<i>Purpose of Tables</i> .....	6
2.2.3.1	Purpose of Area Table.....	6
2.2.3.2	Purpose of File Table .....	7
2.2.3.3	Purpose of User Table.....	7
2.2.4	<i>Relations</i> .....	7
<b>3</b>	<b>REFERENCES .....</b>	<b>7</b>
<b>4</b>	<b>APPENDIX 1 – XML SCHEMA.....</b>	<b>8</b>

# 1 Introduction

The section introduces the Database Design Document (DDD) for LocAdoc to its readers.

## 1.1 Document Objectives

This DDD for the LocAdoc software has the following objectives:

- Describe the design of a DynamoDB and SQLite database, that is, a collection of related data stored in one or more computerized files in a manner that can be accessed by users or computer programs via a database management system (DBMS). It can also describe the software units used to access or manipulate the data.
- To serve as the basis for implementing the database. It provides the acquirer visibility into the design and provides information needed for software support.

## 1.2 Intended Audiences

This DDD is intended for the following audiences:

- Technical reviewers, Supervisor and UOW staff who must evaluate the quality of this document.
- LocAdoc developers including:
  - Architects, whose overall architecture must meet the requirements specified in this document.
  - Designers, whose design must meet the requirements specified in this document.
  - Programmers, whose software must implement the requirements specified in this document.
  - Testers, whose test cases must validate the requirements specified in this document.

## 1.3 References

This DDD refers to the following references:

- Software requirement specification: SRS\_LocAdoc.docx
- Project Proposal: Project\_Proposal\_SS173D\_V1.docx

## 2 Detailed Database Design

This section describes the actual design of different databases at varying levels of abstraction. A subsection for each of conceptual, internal, logical and physical levels.

### 2.1 DynamoDB design (NoSQL database)

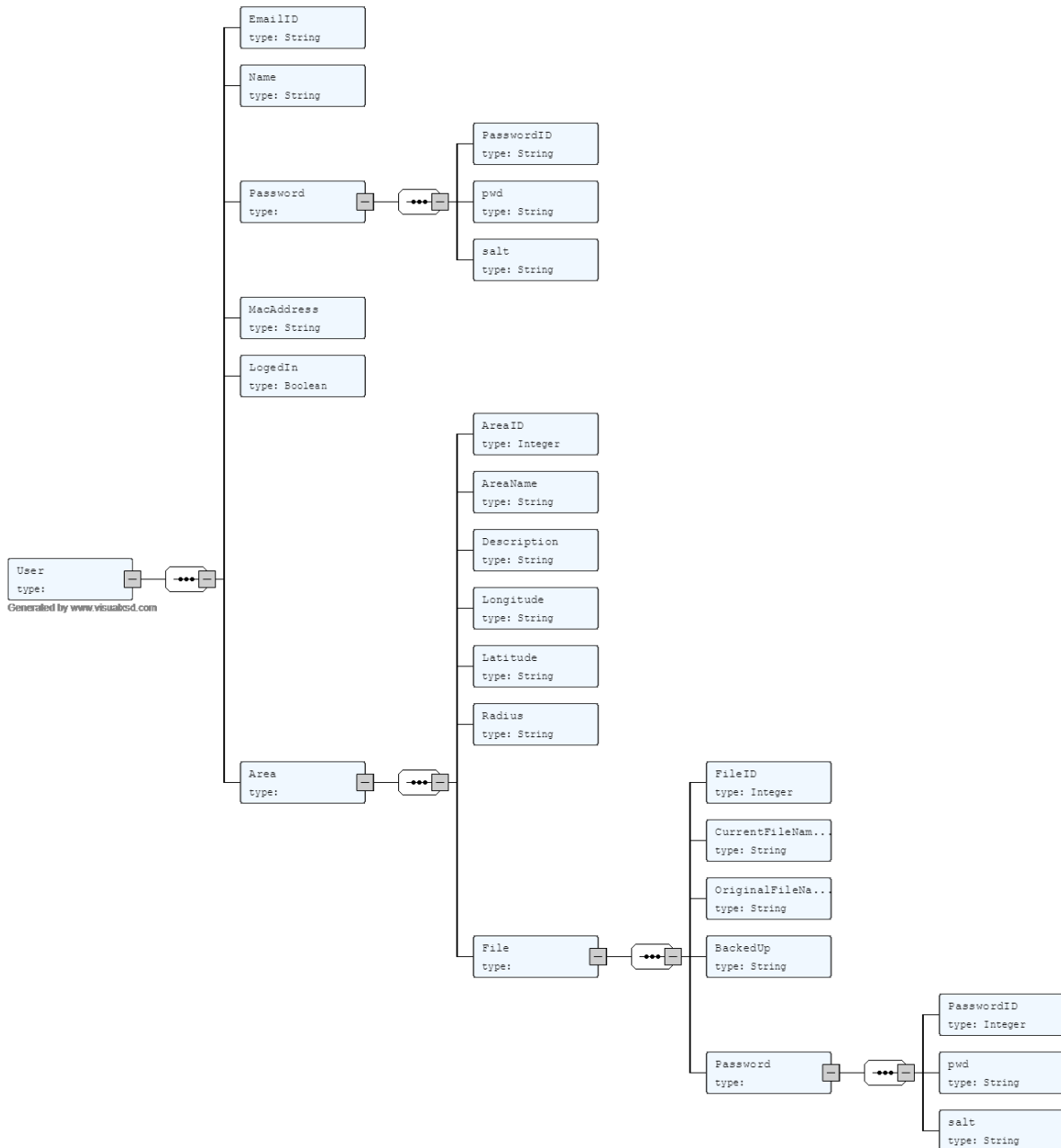


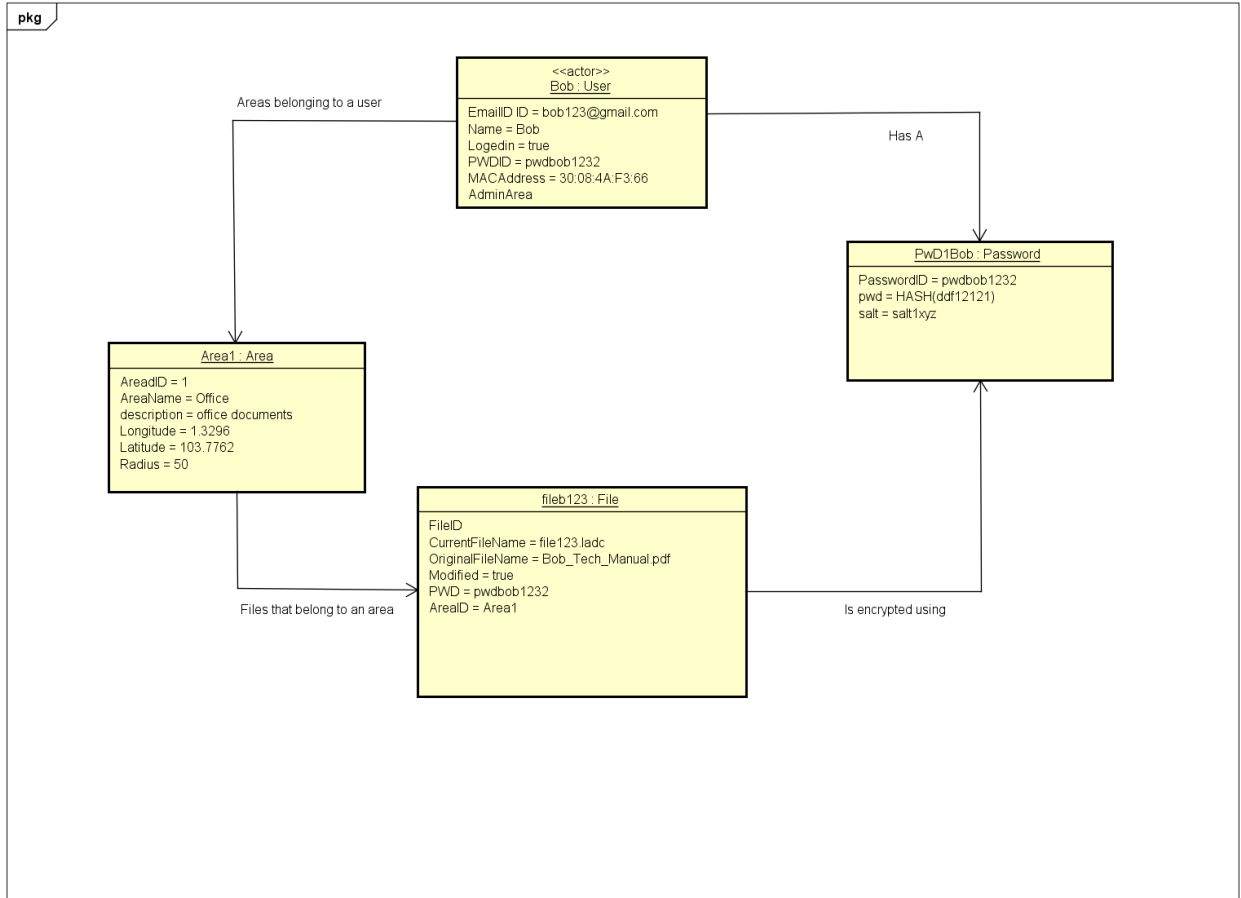
Figure 1: Database design

The diagram given above visualize the NoSQL database. This design was developed after creating a XML schema (Appendix 1) and using an online converter. [2]

The user element will hold all the user details which be mapped to there to his password and files. He will also have an option to set an user administration area.

### 2.1.1 Object Diagram

Here is an object diagram to show a given instance of the database.



This object diagram shows a sample data set and their relationships with other objects. For the propose of the object please refer to the section 2.2.3.

## 2.1.2 Data dictionary

### 2.1.2.1 Data dictionary for Element: User

Name	Data Type	Constrain	Description
Email ID (primary key)	string	Min :1, Max:1	Email ID of the user
Name	String		Name of the user
Password (Foreign Key)	Integer	Min :1, Max:1	The password of the user
LoggedIn	String		Used to flag if the person is currently logged in a devise. So the second login can be detected.
InstanceID	String		Stores the application installation instance.
AdminArea	Integer		User has the freedom to set the admin area.

### 2.1.2.2 Data dictionary for Element: Password

Name	Data Type	Constrain	Description
Password ID (primary key)	Integer	Min :1, Max:1	ID to identify the password
Password	String		Hashed Password
Salt	String		Salt to prevent repeated keys being generated for encryption due to similar password.

### 2.1.2.3 Data dictionary for Element: File

Name	Data Type	Constrain	Description
FileID (primary key)	Integer		
CurrentFileName	string	Min :1, Max:1	A new name for the file assigned by the application
OriginalFileName	String		The original file name assigned by user.

<b>BackedUP</b>	Boolean		This a variable to make sure if the data has been backed up or new.
<b>Password</b>	Integer		Password that was used to encrypt the file (password ID)
<b>Area</b>	Integer		The area where the file has been grouped in (Area ID)

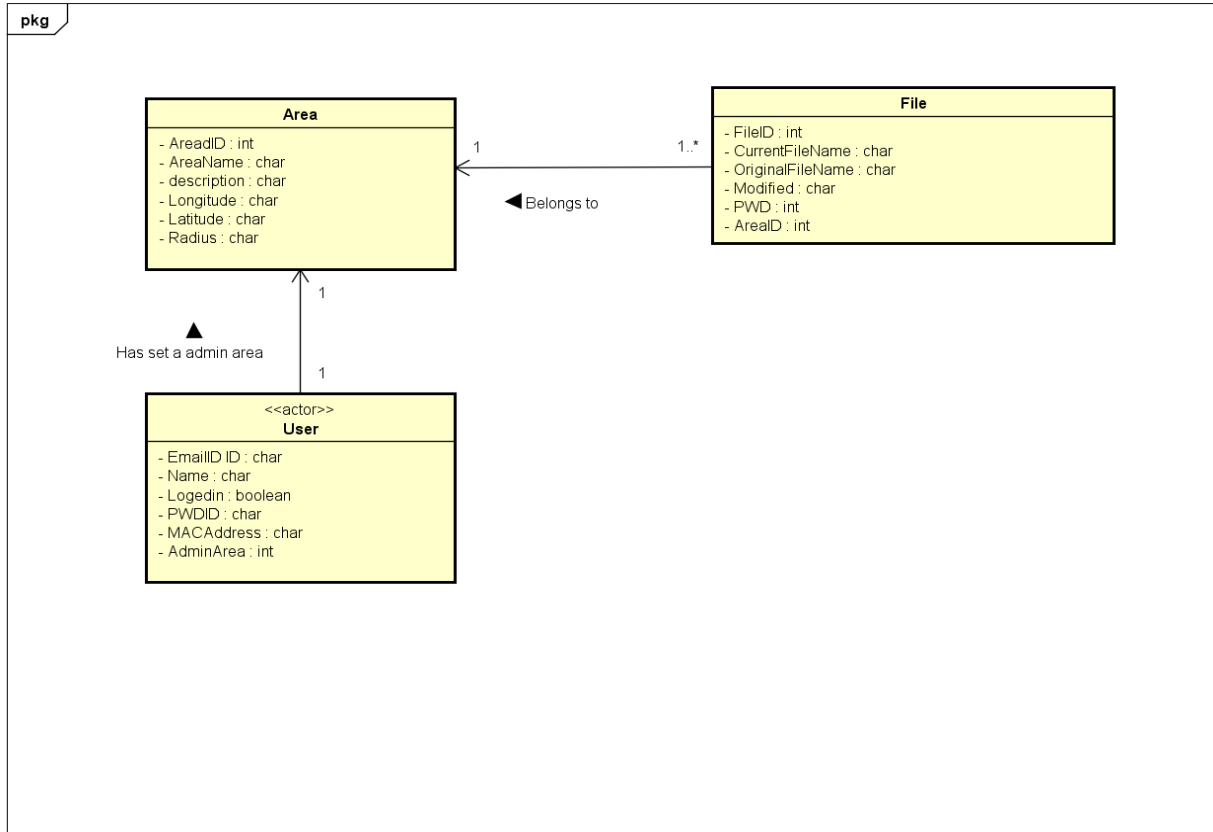
#### 2.1.2.4 Data dictionary for Element: Area

<b>Name</b>	<b>Data Type</b>	<b>Constrain</b>	<b>Description</b>
<b>Area ID (primary key)</b>	string	Min :1, Max:1	ID to identify the area
<b>AreaName</b>	string		Stores the name of the area given by the user.
<b>Description</b>	string		Stores the description created by the user for the area.
<b>Longitude</b>	Decimal		The Longitude of the first file that was created in this area
<b>Latitude</b>	Decimal		The Latitude of the first file that was created in this area.
<b>Radius</b>	Decimal		The radius around the point where the first file was created.



## 2.2 SQLite database design (Relational database)

### 2.2.1 Conceptual diagram



### 2.2.2 Description

This diagram displays the conceptual model of the SQLite database. This database will be created after the it has imported the user’s data. The user session will have the user’s details who is currently logged in. The user will have a password and an admin area where he can make changes to his account such as changing password (optional up to user to set it up). Each user will have zero or more files. The local database will only hold the Area and file information. All the data other than the primary and foreign keys will be stored after encryption using user’s password. The database contents will be decrypted when user makes request.

### 2.2.3 Purpose of Tables

#### 2.2.3.1 Purpose of Area Table

This table stores the information regarding the Area a file was created. The radius is the area around that point where the files grouped in that area can be accessed. The longitudinal and latitudinal value is used to encrypt the file in that area.

### 2.2.3.2 Purpose of File Table

This table stores all the information regarding a file used by the user. When the file is imported into the application, a new file name is generated and it is mapped with the actual table. The longitude and the latitude of the location where the file was created is also stored. The file will also will have a password which was used to encrypt the file.

### 2.2.3.3 Purpose of User Table

The user stores the details of the user currently logged into the system. The user will have a password and an admin area where he can make changes to the account. The primary key of the table will be the email ID. This table will only one record as the database only belongs to one user.

### 2.2.4 Relations

From Table	To Table	Relation
User	Area	A user may set an admin area.
Files	Area	A file belongs to an area.
User	File	A user may saved more than one file.
File	Password	A file will be encrypted using a password.
User	Password	A user has a password.

## 3 References

[1] "Visual BSD," [Online]. Available: <http://visualxsd.com/>.

[2] "Amazone DynamoDB," Amazone, [Online]. Available: <https://aws.amazon.com/dynamodb/>.

## 4 Appendix 1 – XML Schema

This XML schema was created to check if the schema was well formed.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Created with Liquid Studio 2017 - Community Edition 15.1.9.7584 (https://www.liquid-technologies.com) -->
<!-- created with XMLSpear -->
<xsd:schema elementFormDefault="qualified" targetNamespace="http://www.w3schools.com"
  xmlns="http://www.w3schools.com" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="User">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="EmailID" type="xsd:string"/>
        <xsd:element name="Name" type="xsd:string"/>
        <xsd:element name="Password" minOccurs="1" maxOccurs="1">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="PasswordID" type="xsd:string"/>
              <xsd:element name="pwd" type="xsd:string"/>
              <xsd:element name="salt" type="xsd:string"/>
            </xsd:sequence>
          </xsd:complexType>
          <xsd:key name="PK_Password">
            <xsd:selector xpath="User/Password"/>
            <xsd:field xpath="PasswordID"/>
          </xsd:key>
        </xsd:element>
        <xsd:element name="MacAddress" type="xsd:string"/>
        <xsd:element name="LoggedIn" type="xsd:boolean"/>
        <xsd:element name="Area">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="AreaID" type="xsd:integer"/>
              <xsd:element name="AreaName" type="xsd:string"/>
              <xsd:element name="Description" type="xsd:string"/>
              <xsd:element name="Longitude" type="xsd:string"/>
              <xsd:element name="Latitude" type="xsd:string"/>
              <xsd:element name="Radius" type="xsd:string"/>
              <xsd:element name="File">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="FileID" type="xsd:integer"/>
                    <xsd:element name="CurrentFileName" type="xsd:string" maxOccurs="1"/>
                    <xsd:element name="OriginalFileName" type="xsd:string" maxOccurs="1"/>
                    <xsd:element name="BackedUp" type="xsd:string"/>
                    <xsd:element name="Password">
                      <xsd:complexType>
                        <xsd:sequence>
                          <xsd:element name="PasswordID" type="xsd:integer"/>
                          <xsd:element name="pwd" type="xsd:string"/>
                          <xsd:element name="salt" type="xsd:string"/>
                        </xsd:sequence>
                      </xsd:complexType>
                    </xsd:element>
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

```

                                <xsd:key name="PK_File">
                                    <xsd:selector xpath="User/File"/>
                                    <xsd:field xpath="CurrentFileName"/>
                                </xsd:key>
                                </xsd:element>
                                </xsd:sequence>
                            </xsd:complexType>
                            <xsd:key name="PK_AREA">
                                <xsd:selector xpath="User/File/Area"/>
                                <xsd:field xpath="AreaID"/>
                            </xsd:key>
                        </xsd:element>
                    </xsd:sequence>
                </xsd:complexType>
                <xsd:key name="PK_USER">
                    <xsd:selector xpath="User"/>
                    <xsd:field xpath="EmailID"/>
                </xsd:key>
            </xsd:element>
        </xsd:schema>
```