****

**LocAdoc**

**Database Design Document (DDD)**

**Version 0.1**

**Review Draft**

**Prepared by:**

**Abhi Jay Krishnan**

**Kim Hyeocheol**

**Rivaldo Erawan**

**Durrah Afshan**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1 Introduction 3

1.1 Document Objectives 3

1.2 Intended Audiences 3

1.3 References 3

1.4 Database Overview 3

1.5 Document Overview 3

2 Database-wide Design Decisions 3

2.1 Interfaces 3

2.2 Behavior 3

2.3 Appearance / Naming 3

2.4 DBMS Platform 3

2.5 Qualities 3

2.6 Distribution 3

2.7 Operations 3

2.8 Maintenance 3

3 Detailed Database Design 3

3.1 <design level> 3

3.1.1 <Name> Database 3

3.1.1.1 <Type> Logical Grouping 3

3.1.1.1.1 <Type> Field 3

4 Detailed Database Software Design 3

4.1 <Name> Software Component 3

5 Requirements Traceability 3

6 Notes 3

7 Appendices 3

# Introduction

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

## 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 and related software units. It provides the acquirer visibility into the design and provides information needed for software support.
* All sections should remain in this document. If a section is to be tailored out, the section shall remain and contain the words “Tailored out”.

## 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.

## References

This DDD refers to the following references:

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

## Database Overview

This database fills the following purposes:

* Detailed design of the database
* Data dictionary
* <Intended use>
* <Maintenance objectives>
* <Deployment locations>

## Document Overview

This DDD is organized into the following sections:

* *Introduction*, which introduces the database design for <Application> to its readers including referenced documents and an overview of the database including definition, business goals and context.
* *Database Overview*, which provides a high level description of the database including its definition, business goals, and context.
* *Database-wide design decisions,* which provides decisions about the databases behavioral design (how it will behave, from a user’s point of view, in meeting its requirements, ignoring internal implementation) and other decisions affecting further design of the database.
* *Detailed database design,* which will contain a section for each design level (conceptual, internal, logical, physical).
* *Detailed database software design,* scripts, data access components and management software.
* *Requirements traceability,* which provides traceability from the system or software requirements specification to the database software that implements it.
* *Notes,* which contains any general information that aids in understanding this document (e.g., background information, glossary, rationale). This section shall include an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document and a list of any terms and definitions needed to understand this document.
* *Appendices*, which may be used to provide information published separately for convenience in document maintenance.

# 

# Database-wide Design Decisions

This section documents decisions about the databases behavioral design (how it will behave, from a user’s point of view, in meeting its requirements, ignoring internal implementation) and other decisions affecting further design of the database.

## Interfaces

Design decisions regarding queries or other inputs the database will accept and outputs (displays, reports, messages, responses, etc.) it will produce, including interfaces with other systems and users.

## Behavior

Design decisions on database behavior in response to each input or query including actions, response times and other performance characteristics, selected equations / algorithms / rules, disposition, and handling of unallowed inputs.

## Appearance / Naming

All application will keep the databases transparent from the user and the user will not be able to directly interact with the database.

## DBMS Platform

This project will make use of two different DBMS:-

* DynamoDB for a central database which is a NoSQL database and will be used as central database for the application.
* SQLite database which will be created by the application while application is running to store temporary data. This database is created to reduce the number communication over head with the central server.

## Qualities

The DynamoDB is designed to highly scalable and quick compared to the conventional relational database [1].

## Maintenance

The DynamoDB database will be maintained by the administrator. This will scale as the number users using the app increase. The database server is maintained by Amazon web services. The SQLite database is maintained by the app itself and will be dropped when the user logs out for security purposes.

# 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.

## DynamoDB design (NoSQL database)

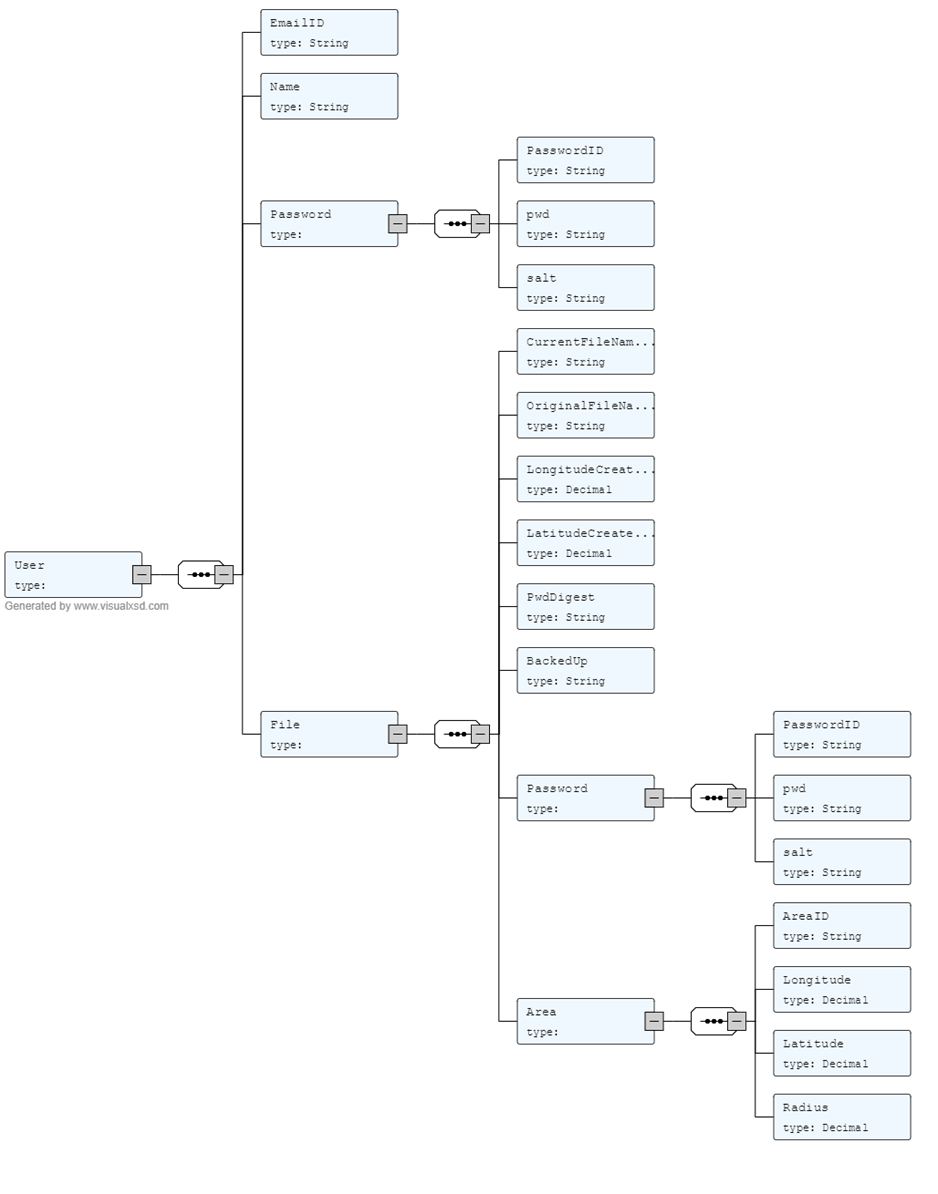
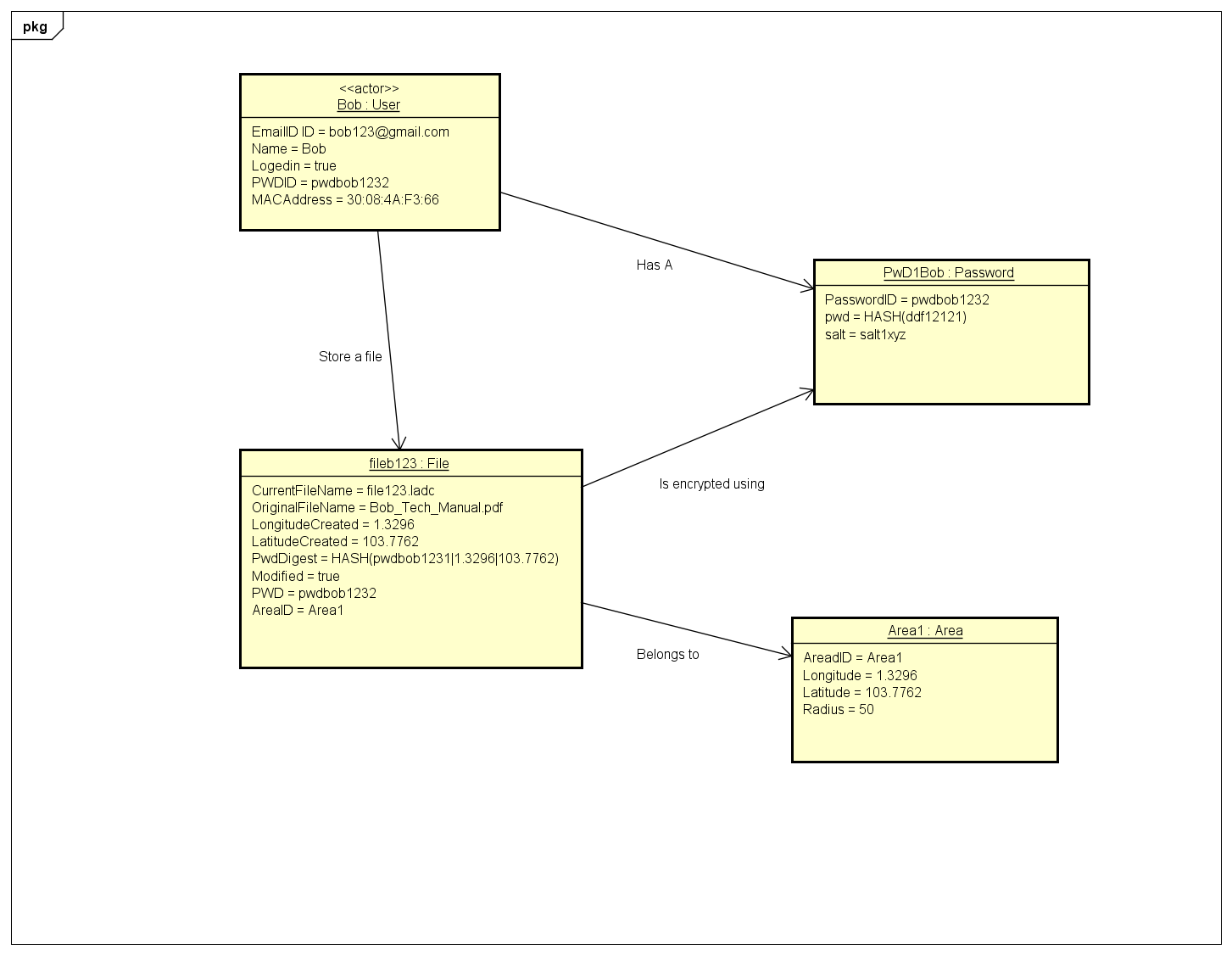


Figure 1: Database design

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]

### Object Diagram

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



### Data dictionary

#### 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) | Password | Min :1 , Max:1 | The password of the user |
| LogedIn | Boolean |  | Used to flag if the person is currently logged in a devise. So the second login can be detected. |
| MacAddress | String |  | Used to store mac address of the phone used by the user |

#### Data dictionary for Element: Password

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

# Detailed Database Software Design

This section contains subsections for each software module used within the database. This includes but DynamoDB NoSQL database design and SQLite Relational database design.

## DynamoDB Design

# Requirements Traceability

This section shall map each software component defined above in section 4 to a set of requirements stated in the SRS.

# Notes

This section shall any general information that aids in understanding this document (e.g., background information, glossary, rationale). This section shall include an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document and a list of any terms and definitions needed to understand this document.

# Appendix 1 – XML Schema

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



