



GO Tyme
Database Design Document (DDD)
Version 1.0

Prepared by: Mark Rid Ramirez

Revision History

Date	Version	Description	Author

Table of Contents

- 1 INTRODUCTION..... 1
 - 1.1 DOCUMENT OBJECTIVES..... 1
 - 1.2 INTENDED AUDIENCES..... 1
 - 1.3 REFERENCES..... 1
- 2 DETAILED DATABASE DESIGN 2
 - 2.1 DYNAMODB DESIGN (NOSQL DATABASE) 2
 - 2.1.1 Object Diagram 2
 - 2.1.2 Data dictionary..... 2
 - 2.1.2.1 Data dictionary for Element: userinfotable 2
 - 2.1.2.2 Data dictionary for Element: debitcard 2
 - 2.1.2.3 Data dictionary for Element: deposits 3
 - 2.1.2.4 Data dictionary for Element: gobalancedeposits 3
 - 2.1.2.5 Data dictionary for Element: transactionhistory 3
 - 2.1.2.6 Data dictionary for Element: claimhistory 3
 - 2.2 SQLITE DATABASE DESIGN (RELATIONAL DATABASE)..... 5
 - 2.2.1 Conceptual diagram 5
 - 2.2.2 Description 5
 - 2.2.3 Purpose of Tables 5
 - 2.2.3.1 Purpose of Area Table 5
 - 2.2.3.2 Purpose of File Table Error! Bookmark not defined.
 - 2.2.3.3 Purpose of User Table 6
 - 2.2.4 Relations 6
- 3 REFERENCES..... 7
- 4 APPENDIX 1 – XML SCHEMA..... 7

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.

2.1.1 Data dictionary

2.1.1.1 Data dictionary for Element: userinfotable

Name	Data Type	Constrain	Description
accountnumber (primary key)	Char	Min :12, Max:12	Account number of the user
username	Varchar	Min :1, Max:20	Userame of the user
userpassword	Varchar	Min :1, Max:30	The password of the user
balance	DECIMAL	Min :1, Max:30	Account Balance of the user
gobalance	DECIMAL	Min :1, Max:30	GO Save Balance of the user
phonenummer	Char	Min :11, Max:11	Phone number of the user
email	Varchar	Min :1, Max:100	Email address of the user
birthday	Date		Birth date of the user

2.1.1.2 Data dictionary for Element: debitcard

Name	Data Type	Constrain	Description
accountnumber (Foregin key)	Char	Min :12, Max:12	Account number of the user
cardnumber (primary key)	Char	Min :16, Max:16	Cardnumber of the user

expires	Char	Min :5, Max:5	Debit card ex
cvv	Char	Min :3, Max:3	Salt to prevent repeated keys being generated for encryption due to similar password.

2.1.1.3 Data dictionary for Element: deposits

Name	Data Type	Constrain	Description
accountnumber (Foregin key)	Char	Min :12, Max:12	Account number of the user
deposit_amount	Decimal	Min :1, Max:30	Amount deposited by the user
deposit_date	Datetime		Deposit date of the user

2.1.1.4 Data dictionary for Element: gobalancedeposits

Name	Data Type	Constrain	Description
accountnumber (Primary key)	Char	Min :12, Max:12	Account number of the user
deposit_amount	Decimal	Min :1, Max:30	Amount deposited to GO Save by the user
deposit_date	Datetime		Date deposited

2.1.1.5 Data dictionary for Element: claimhistory

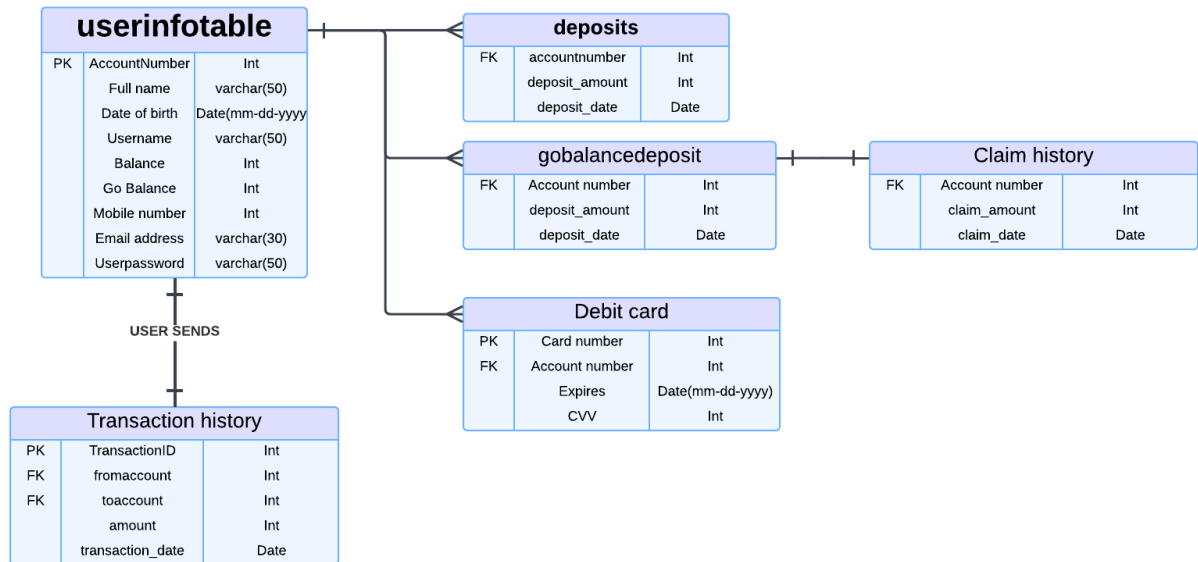
Name	Data Type	Constrain	Description
accountnumber (Primary key)	Char	Min :12, Max:12	Account number of the user
claim_amount	Decimal	Min :1, Max:30	Claimed amount by the user from GO Save
claim_date	Datetime		Claim date by the user from GO Save

2.1.1.6 Data dictionary for Element: transactionhistory

Name	Data Type	Constrain	Description
transactionID (Primary key)	Char	Min :6, Max:6	TransactionID is automatically generated when a transaction is made
Fromaccount (Foregin key)	Char	Min :12, Max:12	Sender
Toaccount (Foregin key)	Char	Min :12, Max:12	Receiver
amount	Decimal	Min :1, Max:30	Amount sent
transaction_date	Datetime		Date of transaction

2.2 MySQL 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 when the user signs up. The user can login to access the savings account, GO Save, debitcard and other features. The user can do transactions from Debit card, Send, GO Save and it will be recorded in Transaction history along with its details.

2.2.3 Purpose of Tables

2.2.3.1 Purpose of userinfotable Table

This table stores the core information of users in the system. It includes details such as the user's account number, username, account name, password, balance, GoBalance (a specific type of balance), phone number, email, and birthday. The accountnumber is the primary key, ensuring each user has a unique identifier.

2.2.3.2 Purpose of deposits Table

This table records the deposit transactions made by users. It stores the account number of the user, the amount deposited, and the date and time of the deposit. The accountnumber is a foreign key that links to the userinfotable, ensuring deposits are associated with valid users.

2.2.3.3 Purpose of gobalancedeposits Table

This table specifically tracks deposits made into the GoBalance account of users. It stores the account number, the deposit amount, and the date and time of the deposit. The combination

of accountnumber and deposit_date serves as the primary key, ensuring no duplicate records for the same deposit event.

2.2.3.4 Purpose of debitcard Table

This table stores the debit card information associated with user accounts. It includes the account number, card number, expiration date, and CVV code. The cardnumber is the primary key, ensuring each card is unique, while the accountnumber is a foreign key linking to the userinfotable.

2.2.3.5 Purpose of transactionhistory Table

This table records all transactions between user accounts. It stores the transaction ID, the source account (fromaccount), the destination account (toaccount), the transaction amount, and the date and time of the transaction. The transactionID is the primary key, ensuring each transaction is uniquely identified. Both fromaccount and toaccount are foreign keys referencing the userinfotable.

2.2.3.6 Purpose of claimhistory Table

This table tracks the claims made by users. It stores the account number, the claim amount, and the date and time of the claim. The combination of accountnumber and claim_date serves as the primary key, ensuring no duplicate claims are recorded for the same user at the same time. The accountnumber is a foreign key referencing the userinfotable.

2.2.4 Relations

From Table	To Table	Relation
deposits	userinfotable	deposits.accountnumber references userinfotable.accountnumber
gobalancedeposit	userinfotable	gobalancedeposit.accountnumber references userinfotable.accountnumber
debitcard	userinfotable	debitcard.accountnumber references userinfotable.accountnumber
transactionhistory	userinfotable	transactionhistory.fromaccount and toaccount reference userinfotable.accountnumber
claimhistory	userinfotable	claimhistory.accountnumber references userinfotable.accountnumber

3 References

[1] " GoTyme Bank, "GoTyme Philippines," [Online]. Available: <https://www.gotyme.com.ph>.

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"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<!-- Root element for the database -->
```

```
<xsd:element name="UserDatabase">
  <xsd:complexType>
    <xsd:sequence>
      <!-- UserInfoTable -->
      <xsd:element name="UserInfoTable" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="AccountNumber" type="xsd:string"/>
            <xsd:element name="Username" type="xsd:string"/>
            <xsd:element name="AccountName" type="xsd:string"/>
            <xsd:element name="UserPassword" type="xsd:string"/>
            <xsd:element name="Balance" type="xsd:decimal"/>
            <xsd:element name="GoBalance" type="xsd:decimal"/>
            <xsd:element name="PhoneNumber" type="xsd:string"/>
            <xsd:element name="Email" type="xsd:string"/>
            <xsd:element name="Birthday" type="xsd:date"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>

      <!-- Deposits -->
      <xsd:element name="Deposits" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="AccountNumber" type="xsd:string"/>
            <xsd:element name="DepositAmount" type="xsd:decimal"/>
            <xsd:element name="DepositDate" type="xsd:dateTime"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>

      <!-- GoBalanceDeposit -->
      <xsd:element name="GoBalanceDeposit" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="AccountNumber" type="xsd:string"/>
            <xsd:element name="DepositAmount" type="xsd:decimal"/>
            <xsd:element name="DepositDate" type="xsd:dateTime"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>

      <!-- DebitCard -->
      <xsd:element name="DebitCard" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="AccountNumber" type="xsd:string"/>
            <xsd:element name="CardNumber" type="xsd:string"/>
            <xsd:element name="Expires" type="xsd:string"/>
            <xsd:element name="CVV" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>

      <!-- TransactionHistory -->
      <xsd:element name="TransactionHistory" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
```

```
<xsd:element name="TransactionID" type="xsd:string"/>
<xsd:element name="FromAccount" type="xsd:string"/>
<xsd:element name="ToAccount" type="xsd:string"/>
<xsd:element name="Amount" type="xsd:decimal"/>
<xsd:element name="TransactionDate" type="xsd:dateTime"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>

<!-- ClaimHistory -->
<xsd:element name="ClaimHistory" maxOccurs="unbounded">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="AccountNumber" type="xsd:string"/>
      <xsd:element name="ClaimAmount" type="xsd:decimal"/>
      <xsd:element name="ClaimDate" type="xsd:dateTime"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
</xsd:schema>
```