

**Database Dictionary Functional System Specification**

|  |  |
| --- | --- |
| Version | 0.2 |
| Release Date | 22/01/2016 |
| Release State | Draft |
| Approval State | Review |
| Author (s) | Khensani Baloyi & Kgothatso Manganye |
| File Name | G4S Deposita Database Dictionary Functional System Specification |
| Confidentiality Category | Confidential |

**Document Control**

**Version Control**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version #** | **Author** | **Version Date** | **Section Changed** | **Change Description** |
| 0.2 | Kgothatso Manganye & Khensani Baloyi | 22/01/2016 |  |  |
|  |  |  |  |  |

**Document Approval**

The list below details the project signatories. By signing below, they acknowledge that they have read, understood and are committed to the acceptance of the document, including any attached documentation or changes.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name & Surname** | **Role** | **Date Signed** | **Signature** |
| Sidalan Moodley | IT Director |  |  |
| Kutloisiso Mona | Lead Developer |  |  |
| Amahle Jenkins | Business Analyst |  |  |

**Document Distribution**

The individuals listed in the document distribution list below serve as recipients of the document only and do not serve as signatories to the document. Feedback is however welcomed as part of the review process.

|  |  |
| --- | --- |
| **Name & Surname** | **Role** |
| Khensani Baloyi | System Development Learner |
| Kgothatso Manganye | System Development Learner |
| Development Team |  |
|  |  |

**Contents**

[**Document Control** 2](#_Toc441651784)

[Version Control 2](#_Toc441651785)

[Document Approval 2](#_Toc441651786)

[Document Distribution 2](#_Toc441651787)

[**1.** **Introduction** 4](#_Toc441651788)

[1.1 Project Overview 5](#_Toc441651789)

[1.2 Benefits 5](#_Toc441651790)

[1.3 Functionality development 5](#_Toc441651791)

[1.4 Out of Scope 5](#_Toc441651792)

[1.5 Users 6](#_Toc441651793)

[**2.** **Use Cases** 7](#_Toc441651794)

[2.1 Use Case Diagram 7](#_Toc441651795)

[2.1.1 Create Database Dictionary 8](#_Toc441651796)

[2.1.1.1 Use Case Description 8](#_Toc441651797)

[2.1.1.2 Activity Diagram 8](#_Toc441651798)

[2.1.2 List Databases 9](#_Toc441651799)

[2.1.2.1 Use Case Description 9](#_Toc441651800)

[2.1.2.2 Activity Diagram 9](#_Toc441651801)

[2.1.3 Edit Database 10](#_Toc441651802)

[2.1.4 Create Tables 11](#_Toc441651803)

[2.1.5 List Tables 12](#_Toc441651804)

[2.1.5.1 Use Case Description 12](#_Toc441651805)

[2.1.5.2 Activity Diagram 12](#_Toc441651806)

[2.1.6 Edit Tables 12](#_Toc441651807)

[2.1.7 Create Table Fields 13](#_Toc441651808)

[2.1.8 List Table Fields 14](#_Toc441651809)

[2.1.9 Edit Table Fields 15](#_Toc441651810)

[2.1.10 Browse 16](#_Toc441651811)

[2.1.10.1 Use Case Description 16](#_Toc441651812)

[2.1.10.2 Activity Diagram 16](#_Toc441651813)

[2.1.11 Search 16](#_Toc441651815)

[2.1.11.2 Activity Diagram 17](#_Toc441651816)

[**3.** **Data Table Design** 17](#_Toc441651817)

1. **Introduction**
   1. **Project Overview**

The main objective of this specification is to document the requirements of a database dictionary in as much detail as possible in order for Database Administrators and Database Developers to use this as the basis of their coding. This document describes the behaviour of a database dictionary system, the process of storing and accessing information from the database dictionary and its use as a reference in understanding the structure of a database.

A Database dictionary is a storage system that is used to store names and descriptions of database objects. The system is going to assist database developers and database administrators learn and understand the structure of the database and what information gets saved into which table(s). The database dictionary will also have functions for browsing and searching for meanings of database objects.

* 1. **Benefits**

Benefits of a Database Dictionary include:

* **Ease of Use**
* It is a good tool to refer to when a user doesn’t understand the structure of a specific database.
* When a new user (Database Developer & Database Administrator) is introduced to the system, it will be a good tool to help them to identify table structure and types.
* Data dictionary will provide a point of reference for a developer when they want clarity on data objects in database.
* **Improve data management**
* When a user is inserting information in a database they will know how and where the information is stored.
  1. **Functionality development**

The functionality of the Database dictionary were agreed that the development of a web based application that is accessible online, to registered users.

* System must allow database developers to save information about the database and the data that is stored in that database,
* Provide easy access to the type of data that they should expect to see in every table, row and columns of database.
* Ability to edit information stored in database dictionary.
  1. **Out of Scope**
* System will not be used to develop the actual databases, or table and fields of the databases.
* It cannot be used to access records inside database.
  1. **Users**

The users of the Data Dictionary are Database Administrators and Database Developers.  It can also be used by programmers working on updating or replacing legacy systems to figure out the basic data structures of the systems they are renovating and the definitions of individual objects.

1. **Use Cases**
   1. **Use Case Diagram**



* + 1. **Create Database Dictionary**
       1. Use Case Description
       2. Activity Diagram

|  |  |
| --- | --- |
| **Use Case** | CREATE DATABASE DICTIONARY |
| **Purpose** | To create a database dictionary. |
| **Actors** | Database Developer |
| **Trigger** | Actor (Database Developer and the System (Logic)) |
| **Basic Flow** | 1. Open application 2. Select Add 3. Enter Name and Description 4. Click Save 5. System checks if database description is there? 6. System checks if Database Name is unique? 7. If Yes to any of the above, System will Display Error 8. If No, System will display Success Message 9. Display Database Dictionary in database dictionary Screen |
| **Alternative Flow** | User Clicks Cancel and a Database dictionary screen is displayed |
| **Exceptional Flow** | If database name entered exist and/or description is empty, system will display error message |
| **Pre-Condition** | Start-up Screen has to have an option to add database |
| **Acceptance Criteria** | 1. Database information Successfully Saved 2. Database appears in Start-up Screen |
| **Business Rule** | 1. Actor must have authority to use this function 2. Database name must be unique and have a description |



* + 1. **List Databases**
       1. Use Case Description

|  |  |
| --- | --- |
| Use Case | List Database Information |
| Purpose | Purpose of this use case is to have a list of Database information saved into the system |
| Actors | Database Developer and Database Administrators |
| Trigger | Actors (Database Developer and DBA) has to see list of saved database information |
| Basic Flow | 1. Open application 2. System generates list of saved database information 3. Database information appears on startup screen |
| Pre-Condition | Start-Up screen should contain list of saved database information |
| Acceptance Criteria | List of Saved Database information(Name and Description)  List of Database Information Should be sorted alphabetically |

* + - 1. Activity Diagram



* + 1. **Edit Database** 
       1. Use Case Description

|  |  |
| --- | --- |
| Use Case | Edit Database Information |
| Purpose | Purpose of this use case is to Edit Database information that is saved into system |
| Actors | Database Developer |
| Trigger | Actor (Database Developer) Edit Database information |
| Basic Flow | 1. Open application 2. Click Edit 3. Add Changes to Name and Description 4. Click Save 5. Changes added to selected Database Name |
| Alternative Flow | Click Cancel to exit and Reload to Start-Up Page that contain list of database information list |
| Exceptional Flow | If Database name exist and description textbox is empty system will not add changes .It will display error Message |
| Pre-Condition | Start-Up Page should have Edit Function |
| Acceptance Criteria | Database Information Successfully Changed  Start-up screen should have list of databases and the changed database information |
| Business Rule | Actor must have authority to use this function  Database name must be unique |

* + - 1. Activity Diagram



* + 1. **Create Tables**

|  |  |
| --- | --- |
| Use Case | CREATE TABLE |
| Purpose | Purpose of this use case is to add Table Information inside Database dictionary |
| Actors | Database Developer |
| Trigger | Actor (Database Developer) has to add Table information. |
| Basic Flow | 1. Open application 2. Select Database Dictionary Name 3. Click List 4. Click Add 5. Enter Table Name and Description 6. Click Save 7. System will add table information and Reload to Tables List |
| Alternative Flow | User clicks Cancel to exit and return back to the list of Tables from selected Database Name |
| Exceptional Flow | If table name exist and description textbox is empty system will not add table it will display error message |
| Pre-Condition | Database list Screen has to have an option to add table |
| Acceptance Criteria | Table Successfully added  Table added appears inside database dictionary |
| Business Rule | Actor must have authority to use this function  Table name must be unique |

* + - 1. Use Case Description
      2. Activity Diagram



* + 1. **List Tables**
       1. Use Case Description

|  |  |
| --- | --- |
| Use Case | List Tables |
| Purpose | Purpose of this use case is to list saved tables in a specific database dictionary |
| Actors | Database Developer and Database Administrators |
| Trigger | Actors (Database Developer and DBA) Table description has to be added. |
| Basic Flow | 1. Open application 2. Select Database Dictionary Name 3. Click View 4. System populates tables saved for the selected database |
| Pre-Condition | Table List Page should have a list of table from selected database |
| Acceptance Criteria | List of tables is displayed.  Table information listed and sorted by alphabetically |

* + - 1. Activity Diagram



* + 1. **Edit Tables**

|  |  |
| --- | --- |
| Use Case | Edit Tables |
| Purpose | To Edit table information that is saved into system |
| Actors | Database Developer |
| Trigger | Actor (Database Developer) has to Edit table information |
| Basic Flow | 1. Open application 2. Select Database Name 3. Click View 4. Select Table Name 5. Click Edit 6. Add Changes to Name and Description 7. Click Save 8. Changes added to selected Table Name |
| Alternative Flow | Click Cancel to exit and Reload to Table Page that contain list of table information list |
| Exceptional Flow | If Table name exist or description textbox is empty system will not add |
| Pre-Condition | Be in database Dictionary Screen |
| Acceptance Criteria | Table information successfully changed  Table page should have list of tables and the edited table information |
| Business Rule | Actor must have authority to use this function  Table name must be unique |

* + - 1. Use Case Description
      2. Activity Diagram



* + 1. **Create Table Fields** 
       1. Use Case Description

|  |  |
| --- | --- |
| Use Case | CREATE FIELD/COLUMN |
| Purpose | Purpose of this use case is to add a Field/Column Information |
| Actors | Database Developer |
| Trigger | Actor (Database Developer) has to add Field/Columns Information |
| Basic Flow | 1. Open application 2. Select Database Name 3. Click List 4. Select Table Name 5. Click List 6. Click Add 7. Enter Field Name and Description 8. Click Save 9. System will add and Reload to field/columns List |
| Exceptional Flow | If column name exist and description textbox is empty it will show an error |
| Pre-Condition | Field Page has to have an option to add field  Database and Table must exist. |
| Acceptance Criteria | Field Successfully Added  Field appears on Field list |
| Business Rule | Actor must have authority to use this function  Field name must be unique |

* + - 1. Activity Diagram



* + 1. **List Table Fields**

|  |  |
| --- | --- |
| Use Case | List Fields/Columns |
| Purpose | Purpose of this use case is to list saved fields/columns in a specific table Information |
| Actors | Database Developer and Database Administrators |
| Trigger | Actors (DBA and Database Developer) has to see list of fields/columns stored. |
| Basic Flow | 1. Open application 2. Select Database Name 3. Click List 4. Select Table Name 5. Click List 6. System Populate list of columns from selected table |
| Alternative Flow | Edit column information |
| Pre-Condition | Field List Page should have a list of columns from selected table |
| Acceptance Criteria | List of columns  Column Information listed and Sorted Alphabetically |

* + - 1. Use Case Description
      2. Activity Diagram



* + 1. **Edit Table Fields**

|  |  |
| --- | --- |
| Use Case | Edit Fields/Columns |
| Purpose | Purpose of this use case is to Edit Columns information that is saved into system |
| Actors | Database Developer and Database administrators |
| Trigger | Actor (Database Developer) has to Edit fields information |
| Basic Flow | Open application  Select Database Name  Click List  Select Table Name  Click List  Select Column Name  Click Edit  Add Changes to Name and Description  Click Save  Changes added to selected Table Name |
| Alternative Flow | Click Cancel to exit and Reload to column Page that contain list of columns information list |
| Exceptional Flow | If column name exist and description textbox is empty system will not add changes it will display error Messages |
| Pre-Condition | Columns Page should have Edit Function |
| Acceptance Criteria | Column Information Successfully Changed  Column page should have list of columns and the changed column information |
| Business Rule | Actor must have authority to use this function  Column name must be unique |

* + - 1. Use Case Description
      2. Activity Diagram



* + 1. **Browse**

|  |  |
| --- | --- |
| Use Case | BROWSE |
| Purpose | Purpose of this use case is to let user evaluate and observe the nature of database objects. |
| Actors | Database Developer and Database Administrators |
| Trigger | When actor has to categorize how they want to view database objects information |
| Basic Flow | 1. Open application 2. Click Browse tab 3. Select Database Name 4. Click Browse by Table 5. System Populate results categorized by Tables |
| Alternative Flow | User click Browse by Field to categorize by field |
| Pre-Condition | Database dictionary names has to be shown on startup page |
| Acceptance Criteria | See List of Tables When Browse by Table is clicked  See List of Fields/Columns When Browse by Field is clicked |

* + - 1. Use Case Description
      2. Activity Diagram



* + 1. **Search**

|  |  |
| --- | --- |
| Use Case | SEARCH |
| Purpose | Purpose of this use case is to search through database objects and descriptions |
| Actors | Database Developer and Database Administrators |
| Trigger | When an Actor (DBA, and Developer) want to search for a specific Database object information |
| Basic Flow | 1. Open application 2. Click inside the Search textbox and select an object on the filter 3. Enter any object name or text contained in the object name 4. Click search Button 5. System generate results matching the search text |
| Alternative Flow | User aborts search textbox and continue with other controls in Database Dictionary |
| Exceptional Flow | If the searched word doesn’t match any record system will show error message |
| Pre-Condition | Actor has to know the keyword that they want to search |
| Acceptance Criteria | Results found  View Results containing name and/or descriptions of searched object |
| Business Rule | Search box doesn’t have to be empty |

* + - 1. Use Case Description
      2. Activity Diagram

1. **Data Table Design**



**Architectural Design of the Database Dictionary**

