**Employee Management System (EMS)**

1. Purpose

The purpose of this document is to show how the EMS will be constructed. The system Design Document was Created to ensure that the EMS meet the requirements specified.

The System Design Document provides a description of the system architecture, software, hardware, database design, and security.

1. System Overview

This system will help VSC solutions to manage its branches all overs South Africa as well as its employees

1. System Architecture

* Software

Java SE 8

JSP

JavaScript

PostgresSQL

Spring MVC

JDBC

CSS

HTML

bootstrap

1. Input and Output

**Input**

Table data

* Employee Table

Name

Surname

Birthday

TelephoneNo

MaritalStatus

Location

branchID

* Branch Table

Branch Name

Postal Address

Postal Code

Branch Location

**Output**

Different fields from the above mention databases as required as well as a Map that show markers of different cities

1. Database Design

-- Table: public.Branch

-- DROP TABLE public."Branch";

CREATE TABLE public."Branch"

(

branchid integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

branch\_name character(30) COLLATE pg\_catalog."default",

postal\_address character(30) COLLATE pg\_catalog."default",

branch\_location character(30) COLLATE pg\_catalog."default",

"postal code" character(30) COLLATE pg\_catalog."default",

CONSTRAINT "Branch\_pkey" PRIMARY KEY (branchid)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public."Branch"

OWNER to postgres;

**-- Table: public.Employee**

**-- DROP TABLE public."Employee";**

**CREATE TABLE public."Employee"**

**(**

**name character(30) COLLATE pg\_catalog."default",**

**surname character(30) COLLATE pg\_catalog."default",**

**telephoneno character(30) COLLATE pg\_catalog."default",**

**location character(30) COLLATE pg\_catalog."default",**

**maritalstatus character(30) COLLATE pg\_catalog."default",**

**branchid integer,**

**birthday character(30) COLLATE pg\_catalog."default",**

**empid integer NOT NULL,**

**CONSTRAINT "primary" PRIMARY KEY (empid),**

**CONSTRAINT branch\_key FOREIGN KEY (branchid)**

**REFERENCES public."Branch" (branchid) MATCH SIMPLE**

**ON UPDATE NO ACTION**

**ON DELETE NO ACTION**

**)**

**WITH (**

**OIDS = FALSE**

**)**

**TABLESPACE pg\_default;**

**ALTER TABLE public."Employee"**

**OWNER to postgres;**

**-- Index: fki\_branch\_key**

**-- DROP INDEX public.fki\_branch\_key;**

**CREATE INDEX fki\_branch\_key**

**ON public."Employee" USING btree**

**(branchid ASC NULLS LAST)**

**TABLESPACE pg\_default;**

**CREATE OR REPLACE FUNCTION getEmp\_procedure()**

**RETURNS character varying as $Employees$**

**BEGIN**

**SELECT e.name,e.surname,e.telephoneNo,e.location,b.branch\_name,b.branch\_location**

**from public."Employee" e, public."Branch" b**

**where e.branchid=b.branchid**

**RETURN Employees**

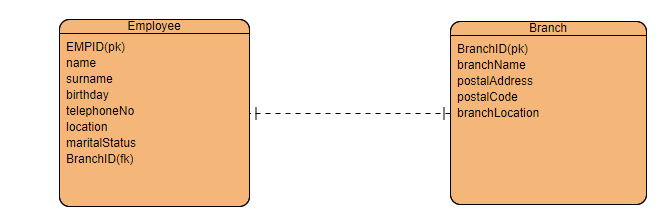
**END;**

**$Employees$ LANGUAGE sql**

**Functional Requirements**

* Add Employee details
* Add Branch details
* View branch details
* View Employee details
* Generate 4 digit ID and merge with initials
* Display city on Map
* Use bootstrap and CSS
* Edit Employee
* Delete Employee
* Update Employee

**ER Diagram**



Relationship label is: **belongs to**