**Documentation**

**Application Brief** and **Technologies used**:

Explain the purpose and goal of the application

Tools used: MySQL Workbench

tool1: MySQL Workbench

tool2: Visual Studio

tool3: Azure Linux VM

tool4: Docker Container

tool4: Jenkins

tool:5 GitHub

tool6: Jira

**ER Diagram:**

Idea: original web app idea was to create a Vet website where owners can input the information of their pets which will then be allocated to a doctor. This will create a one to many relationship between the pet and owner (I.e. one owner can have many pets) and also a one to many relationship between the doctor and pet (i.e. one doctor has multiple pet (patients))

Screenshot of the ER diagram

Diagram

Description automatically generated

**DataBase:**

CREATE TABLE doctor (

doc\_id INT PRIMARY KEY,

doc\_forname VARCHAR(20),

doc\_surname VARCHAR(20),

);

CREATE TABLE owner (

own\_id INT PRIMARY KEY,

own\_forname VARCHAR(20),

own\_surname VARCHAR(20),

);

CREATE TABLE pet (

pet\_id INT PRIMARY KEY,

pet\_name VARCHAR(20),

FOREIGN KEY(own\_id) REFERENCES owner(own\_id) ON DELETE CASCADE,

FOREIGN KEY(doc\_id) REFERENCES doctor(doc\_id) ON DELETE CASCADE

);

**App functionnality:**

Function One : to create a table for table ‘Doctor’

Function Two : to create a table for table ‘Owner’

Function Three : to create a table for table ‘Pet’ which has foreign keys linking to the two previous mentioned tables

**Technical Description:**

Graphical user interface, text

Description automatically generated

**Pipeline(Explain and stages):**

Stage 1(Explain and code): starting from Azure – create Linus VM

Stage 2: install python ‘sudo apt-get install python3’ & ‘sudo install flask’ (for our flask app)

Stage 3: create directory ‘mkdir Jenkinfile’ , create vi ‘vi Jenkins\_install’ – with Jenkin installation – followed through to make Jenkins account (this is so we can use the Jenkins through our Linux VM instead of installing it in onto our local desktop so we can run the pipeline remotely)

Stage 4: Install docker and login in using ‘sudo docker login’ (containerising our app will allow us to keep our environments consistent and it’s also makes the process faster)

Stage 5: edit file /etc/sudoers with Jenkins ‘NO PASSWD’ and add additional default for username authentication

Stage 6: once in Jenkins create new pipeline

Stage 7: Tick GitHub Project and add GitHub link , for Pipeline – Pipeline script from SRC and add same HTTPS GitHub link, change \*/master to \*/main (this will build the web app)

Stage 8: Click Build pipeline and wait for the project to be built and deployed (you can then run the public IP address of the VM with the port number to see your website)

**Draw a blueprint of the tools used, their connection(using Paint):**

**e.g.,**

Diagram

Description automatically generated

**Pipeline(Screensnaps of Console output, and success output)**:

e.g.:

**Future probable updates:**

Update 1: Create a database connection with the Flask Api

Update 2: Also include none relational database (i.e. a review page)

**Video**

* Uploaded onto GitHub