**PETx**

**TEAM ID: BSEF19-17**

**SESSION: BS SE FALL 2019 - 2023**

**PROJECT ADVISOR: NATALIA CHAUDHARY**

**SUBMITTED BY**

|  |  |
| --- | --- |
| **Student Name** | **Roll Number** |
| ATA UL MOHSIN | BSEF19M031 |
| HASSAN AHMAD SARFRAZ | BSEF19M033 |
| MUHAMMAD SAAD | BSEF19M037 |
| MUHAMMAD SALEH BUTT | BSEF19M047 |



**PUNJAB UNIVERSITY COLLEGE OF INFORMATION TECHNOLOGY,**

**UNIVERSITY OF THE PUNJAB, LAHORE.**

**Faculty of Computing & Information Technology**



Title: **“PETx” FYP Documentation**

Group ID: **BSEF19-17**

|  |  |  |
| --- | --- | --- |
| **Sr #.** | **Roll Number** | **Student Name** |
| 1 | BSEF19M031 | Ata Ul Mohsin |
| 2 | **BSEF19M033** | **Hassan Ahmad Sarfraz (Lead)** |
| 3 | BSEF19M037 | Muhammad Saad |
| 4 | BSEF19M047 | Muhammad Saleh Butt |

Project Supervisor: Ma’am **Natalia Chaudhary**

Date of Submission: **June 15, 2023**

###### STATEMENT OF SUBMISSION

This is to certify that the students:

* ATA UL MOHSIN - BSEF19M031
* HASSAN AHMAD SARFRAZ - BSEF19M033 (Team Lead)
* MUHAMMAD SAAD - BSEF19M037
* MUHAMMAD SALEH BUTT - BSEF19M047

have successfully completed the final project named as **PETx**, at the Punjab University College of Information Technology, University of The Punjab, Lahore, to fulfill the partial requirement of the degree of **Bachelor of Science in Software Engineering**.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project Office Supervisor

PUCIT, Lahore





###### Proofreading Certificate

## It is to certify that I have read the document meticulously and circumspectly. I am convinced that the resultant project does not contain any spelling, punctuation or grammatical mistakes as such. All in all I find this document well organized and I am in no doubt that its objectives have been successfully met.

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Business Communication and Technical Writing,

Lecturer, PUCIT

###### Acknowledgement

We truly acknowledge the cooperation and help make by Ma’am **Natalia Chaudhary**, **Lecturer** of **PUCIT, University of the Punjab**. She has been a constant source of guidance throughout the course of this project. We would also like to thank for her help and guidance throughout this project. We are also thankful to our friends and families whose silent support led us to complete our project.

* Mr. Ata Ul Mohsin
* Mr. Hassan Ahmad Sarfraz
* Mr. Muhammad Saad
* Mr. Muhamad Saleh Butt

Date: June 15, 2023

###### Abstract

An abstract is necessary at this stage to give an understanding of the initial requirements of the system. This will show what high-level requirements the proposed system must address. This abstract will act as a foundation for the future analysis of the system.

### Rescue System

Rescue system is the prime feature of our project. This module will require emergency situation information from the informer or the person who told the system about distress. This information should be in a form having location, contact number, and related information if any about the situation. Once an emergency event is initiated, it’ll alert the volunteers and nearby rescue teams to move toward the location. Here only one team will be assigned the task of rescue and that team or volunteer will complete the job. If the task is not completed by a volunteer, a team will be assigned the job then.

### Veterinary Doctor Appointment System

The logged-in customer or pet owner will visit the page of available doctors and will select a doctor according to his requirements. Then, he’ll be asked to schedule an appointment from the available time slots of a doctor. If he selects the time, the doctor will be notified about the scheduled appointment and then the person will move to the payment page which will be the banking site for the transaction.

From the doctor's side, he’ll be able to see scheduled appointments from his side and will be able to change availability hours and fees.

### Social Feed

This will serve as a community for pet owners. They’ll be able to see posts of other users in their feed and like, and comment on them. These posts will be focused on pets only just like a Facebook group. Our first priority in this regard will be to integrate this module with Facebook APIs. Otherwise, we shall develop our own social media system.

###### Table of Contents

[STATEMENT OF SUBMISSION 3](#_heading=h.c6irmztns7iq)

[Proofreading Certificate](#_heading=h.5z1eno6i0gl) [**4**](#_heading=)

[Acknowledgement 5](#_heading=h.y7jn3ozez7tl)

[Abstract 6](#_heading=h.f4p9msakpxs)

[Rescue System 6](#_heading=h.nmf14n)

[Veterinary Doctor Appointment System 6](#_heading=h.37m2jsg)

[Social Feed 6](#_heading=h.1mrcu09)

[Table of Contents 7](#_heading=h.mftny7ib5znv)

[Project Overview Statement 9](#_heading=h.60c29y89w839)

[High Level System Components 9](#_heading=h.4dspm910egtu)

[List of optional functional units 9](#_heading=h.1t3h5sf)

[Gantt chart 10](#_heading=h.1pxezwc)

[**Iteration One 11**](#_heading=h.49x2ik5)

[**Iteration Two 12**](#_heading=h.i9gbh2lfy8w3)

[**Iteration Three 12**](#_heading=h.2p2csry)

[**Iteration Four 13**](#_heading=h.147n2zr)

[**Application Architecture 14**](#_heading=h.wm1nn3es42pp)

[Introduction 14](#_heading=h.ul7an0w702ws)

[High-Level Architecture Overview 14](#_heading=h.5e0dkexdbma)

[Component Breakdown 14](#_heading=h.h7a3frf8ux1n)

[3.1 Rescue Service 14](#_heading=h.u7f5gzgj7ms8)

[3.2 Ecommerce Shop 15](#_heading=h.6aqkvzqmgzsu)

[3.3 Doctor Appointment 15](#_heading=h.mugcy22qhq6r)

[3.4 Social Media Blogging 15](#_heading=h.r7cvzxjhmwqq)

[Front-end Architecture 15](#_heading=h.k3rap0azcuei)

[4.1 App.js: 15](#_heading=h.txh1v85c4ktf)

[4.2 RescueService.js: 15](#_heading=h.15ewqt4tj6q9)

[4.3 EcommerceShop.js: 15](#_heading=h.qjft49aummfi)

[4.4 DoctorAppointment.js: 16](#_heading=h.x2bkbn84cqpw)

[4.5 SocialMediaBlogging.js: 16](#_heading=h.40752sdfqf14)

[Back-end Architecture 16](#_heading=h.ci1p5mj20rcd)

[5.1 Server.js: 16](#_heading=h.lf0c1h9c1wge)

[5.2 Routes: 16](#_heading=h.x4e9hnl46kfe)

[5.3 Controllers: 16](#_heading=h.gclxvjoq4ark)

[5.4 MongoDB Integration: 16](#_heading=h.vdqo49h0ydpx)

[Data Storage and Database Design 17](#_heading=h.pbyemqnayfn8)

[6.1 Users: 17](#_heading=h.num77q522klk)

[6.2 Pets: 17](#_heading=h.1e4xtl71yvj)

[6.3 Products: 17](#_heading=h.1gqc3he2vzpq)

[6.4 Appointments: 17](#_heading=h.db88ekezxo6v)

[6.5 BlogPosts: 17](#_heading=h.f3pioz8sn6xq)

[Authentication and Authorization 17](#_heading=h.56h51mt02omi)

[Deployment and Scalability 17](#_heading=h.dsry2wan1q5r)

[Testing and Quality Assurance 18](#_heading=h.430adqobm4v3)

###### Project Overview Statement

As the name suggests, the project revolves around animal care and services related to them. Since everyone is so engaged in their lives, no one finds enough time to research their pet, which doctor is better, and where to find the best quality pet products. In case, some animal is ever struck in a do-or-die situation, no one knows how to help the poor soul. Thus keeping all this in mind, PETx will provide all these services with ease so that pet owners can serve their pets with peace of mind.

###### High Level System Components

* + 1. Login / SignUp: Leads the user to Landing Page
    2. Rescue Service: No login required, Immediate contact to rescue team
    3. Appointment Section: Browse through doctors list, see reviews and choose from available appointment times.
    4. Blog page: Users can share their insights, findings, research, experiences, or other information with each other.
    5. Payment: Blockchain-based secure payment system

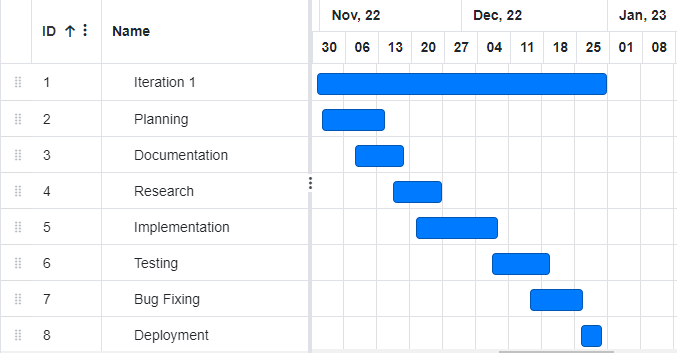
###### List of optional functional units

* Online store: Where users can buy all kinds of pet accessories, food, and medicines.
* Shelter Home: Where you can submit some stray / orphan animals, adopt a pet, or might abandon your pet as well.

###### Gantt chart

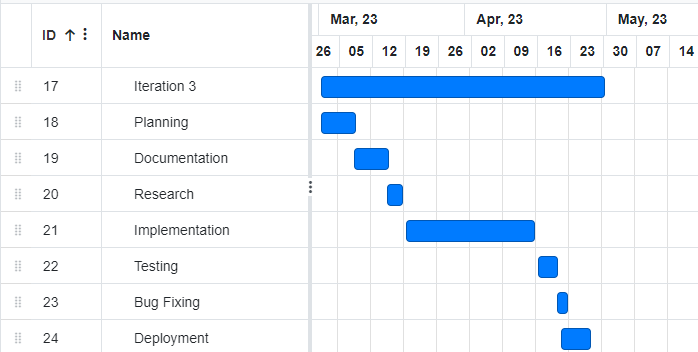


## Iteration One

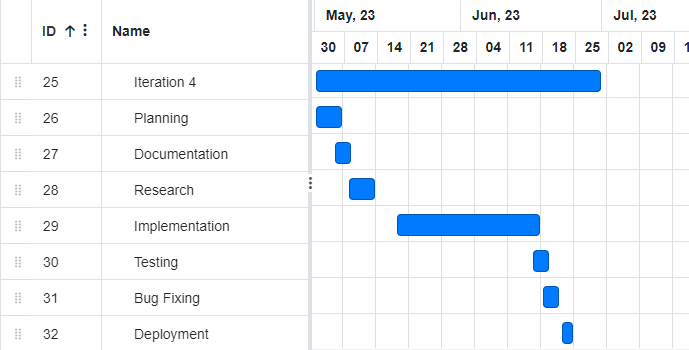


## Iteration Two

## Iteration Three



## Iteration Four



## Application Architecture

We will be using Three-Tier architecture for this project. These 3 layers will be as

* + 1. Frontend Tier
    2. Application Tier
    3. Database Tier

Frontend will be handled using HTML5, CSS3, ES6, and ReactJS. The application tier will be implemented using NodeJS and ExpressJS. While the last layer of the database will be implemented using MongoDB.

**PETx Architecture**

###### Introduction

The PETx application is a comprehensive platform built using the MERN stack (MongoDB, Express.js, React.js, Node.js). It consists of four main modules: Rescue Service, Ecommerce Shop, Doctor Appointment, and Social Media Blogging. This document aims to provide a concise overview of the architecture employed in PETx, highlighting the key components, their interactions, and the design choices made.

###### High-Level Architecture Overview

The PETx application follows a client-server architecture, where the client-side is built with React.js and the server-side is implemented using Node.js and Express.js. MongoDB is used as the database to store and retrieve application data. The architecture can be summarized as follows:

###### Component Breakdown

###### 3.1 Rescue Service

The Rescue Service module allows users to report and seek assistance for lost or injured pets. It includes features such as reporting a lost pet, contacting nearby animal shelters, and providing real-time updates on rescue operations.

###### 3.2 Ecommerce Shop

The Ecommerce Shop module enables users to purchase accessories, food, and other pet-related products. It provides functionalities like product listing, filtering, shopping cart management, and secure payment integration.

###### 3.3 Doctor Appointment

The Doctor Appointment module facilitates scheduling appointments with veterinarians. Users can view available time slots, book appointments, receive reminders, and maintain a history of their pet's medical visits.

### 

###### 3.4 Social Media Blogging

The Social Media Blogging module serves as a platform for pet owners to share experiences, write blogs, and connect with other pet enthusiasts. It includes features like creating blog posts, commenting, liking, and following other users.

### 

### Front-end Architecture

The front-end of PETx is developed using React.js, following a component-based architecture. Key components include:

###### **4.1 App.js**:

The root component that handles routing and navigation.

###### **4.2 RescueService.js:**

Implements the UI and logic for the Rescue Service module.

###### **4.3 EcommerceShop.js:**

Implements the UI and logic for the Ecommerce Shop module.

###### **4.4 DoctorAppointment.js:**

Implements the UI and logic for the Doctor Appointment module.

###### **4.5 SocialMediaBlogging.js:**

Implements the UI and logic for the Social Media Blogging module.

State management is handled using Redux, allowing efficient sharing of data between components and maintaining a centralized application state.

### Back-end Architecture

The back-end of PETx is built using Node.js and Express.js, providing a RESTful API to handle client requests. Key components and features include:

###### 5.1 Server.js:

The main entry point that sets up the Express server and establishes database connections.

###### 5.2 Routes:

Separate route files for each module (Rescue Service, Ecommerce Shop, Doctor Appointment, Social Media Blogging) that handle specific API endpoints and request handling.

###### 5.3 Controllers:

Corresponding controllers for each module that contain the logic for processing requests, interacting with the database, and returning appropriate responses.

###### 5.4 MongoDB Integration:

Utilizes the MongoDB Node.js driver to connect to the database, perform CRUD operations, and manage data persistence.

### Data Storage and Database Design

PETx utilizes MongoDB as the database to store and retrieve application data. The database design includes the following collections:

###### 6.1 Users:

Stores user information including authentication details and user roles.

###### 6.2 Pets:

Contains pet-related data such as pet details, owner information, and medical history.

###### 6.3 Products:

Stores details about products available in the Ecommerce Shop module.

###### 6.4 Appointments:

Records information about scheduled appointments with veterinarians.

###### 6.5 BlogPosts:

Stores blog post content, author details, and associated metadata.

### Authentication and Authorization

PETx implements user authentication and authorization using JSON Web Tokens (JWT). When users register or log in, a token is generated and included in subsequent API requests to verify the user's identity and permissions.

### Deployment and Scalability

The PETx application is deployed using a cloud-based hosting platform such as Heroku. The deployment process involves setting up the server environment, configuring the database connection, and deploying the client-side application. To ensure scalability, load balancing and caching techniques can be employed, and the application can be containerized using Docker for easy deployment.

### Testing and Quality Assurance

PETx follows a comprehensive testing approach, including unit tests, integration tests, and end-to-end tests. Testing frameworks such as Jest and React Testing Library are used for front-end testing, while tools like Postman or Supertest are employed for API testing. Continuous integration (CI) and code quality tools such as Travis CI and ESLint are utilized to ensure code reliability and maintainability.

Performance Optimization

To improve performance, PETx implements various strategies, including client-side caching of static assets, server-side optimizations (e.g., GZIP compression, HTTP/2), and the use of a Content Delivery Network (CDN) to serve static content globally.

Security Considerations

PETx prioritizes security and implements measures such as input validation, encryption of sensitive data, and protection against common web vulnerabilities (e.g., XSS, SQL injection). Communication between the client and server is secured using HTTPS.

Error Handling and Logging

PETx implements robust error handling mechanisms to provide meaningful error messages to users and log errors for debugging and monitoring purposes. Logging is performed using libraries like Winston or Morgan, which log events and errors to a centralized logging service.

This document provides a concise overview of the PETx application's architecture, covering its high-level structure, key components, data storage, authentication, deployment, testing, security measures, and more. It serves as a comprehensive reference for understanding the architecture and implementation of PETx in the MERN stack.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**