1. PROJECT NAME (5 points)

cocoa shelter

2. TEAM MEMBERS (5 points)

Xueyan Zhang

Wenjie Liu

Bo Pang

Sijing Zhang

3. PROBLEM TO BE SOLVED (about 3 sentences, 10 points)

Our goal is to develop an online platform for an animal shelter, aimed at making the adoption process better by enabling users to browse animals online. The platform will have two primary components:

Customer Page: A user-friendly interface that displays animals currently available for adoption.

Shelter Management Portals: A secure, functional dashboard for shelter employees to manage animals.

This will include CRUD capabilities, allowing staff to update profiles.The system will use MySQL for efficient data management, storing essential information on animals, employees.

4. PROJECT MOTIVATION (*why* you want to work on *this* problem, 10 points)

Our team, with experience adopting pets from shelters, is motivated to create a web-based system to improve the adoption process. The platform will simplify animal management for staff and make it easier for adopters to find and apply for pets. Our goal is to enhance shelter efficiency and help more animals find homes.

5. TECHNICAL APPROACH (1 paragraph, 15 points)

We are using Spring Boot to build the backend logic and create RESTful APIs for the shelter system. MySQL is used to store and manage data on animals, and users. Maven handles project dependencies.

6. TOOLS/FRAMEWORKS NEEDED (10 points)

SpringBoot

MyBatis

MySQL

JavaScript HTML CSS

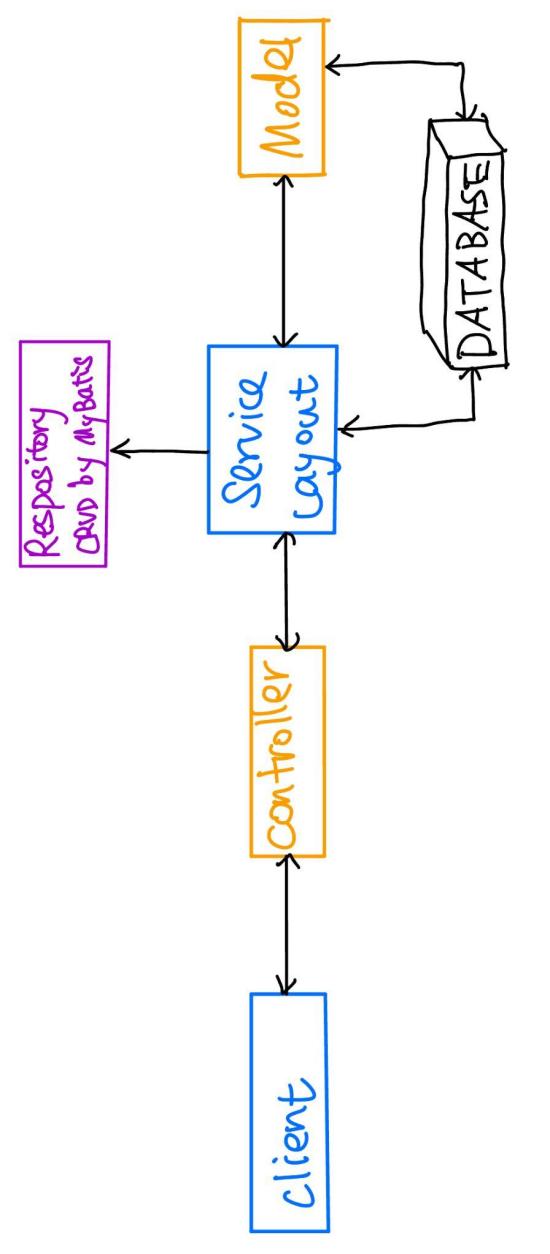
7. DELIVERY PLATFORM (Laptop text IO, Laptop GUI, Android Emulator, Web Page, Other; 10 points)

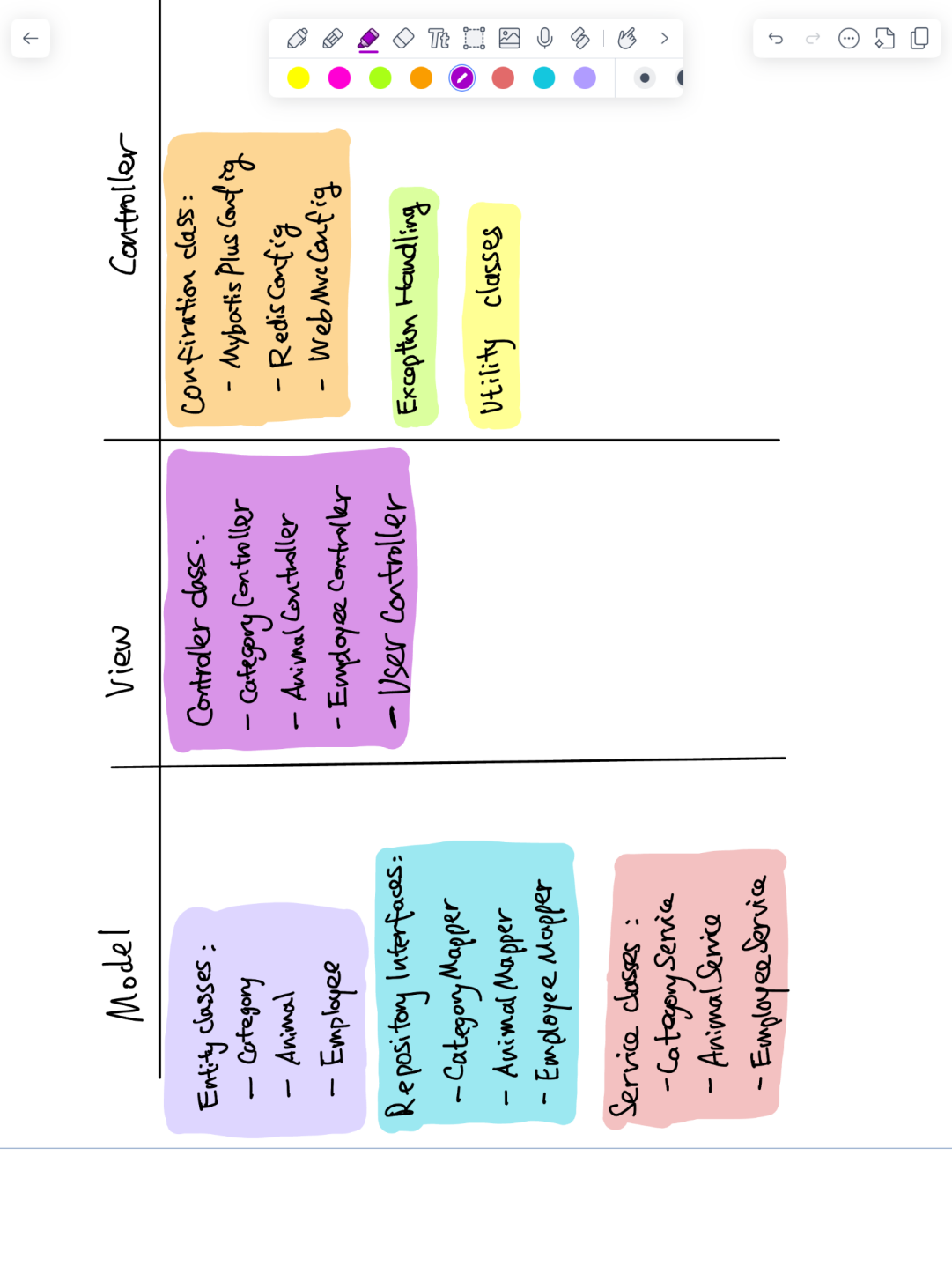
Web Page

8. RISK FACTORS (10 points) -- things you may need to learn along the way or uncertainties in design

A major risk factor is the need to learn and get familiar with the Spring Boot framework for backend development. Understanding its integration with MySQL and managing database connections may present challenges.

9. VERY INFORMAL ARCHITECTURE AND DESIGN DIAGRAMS (MVC or similar; Class Diagram(s), 15 points)  
 This can be very informal, phone picture of sheet of paper, hand-drawn on tablet, or similar.





10. FIVE EXAMPLES OF 5004 "BIG IDEAS" TO BE ILLUSTRATED (10 points)

1.Object-Oriented Programming: Controllers like CategoryController and AnimalController, and service classes. These classes represent different objects that manages specific tasks, they make the code easy to read and easy to maintain.

2.Encapsulation: Classes like CustomerException and GlobalExceptionHandler in the common package. These classes hide exception-handling details, so that external code can rely on them without understanding the inner logic.

3.Interfaces and Polymorphism: Mapper interfaces like CategoryMapper. Interfaces allow different classes to implement the same methods, we think this is a good practice of polymorphism.

4.Inheritance: Configuration classes like MybatisPlusConfig and RedisConfig. These classes reuse logic from their parent class, this reduces redundant code.

5.Unit Testing: We use Junit to write test code, so that we would know whether our code is wrong and where/ how they are wrong by throwing exceptions.

Two project ideas:

The first project is a shelter animal adoption system with two portals: one for customers and one for shelter staff. In the customer portal, users can browse animals, submit adoption applications, and check their status. The shelter portal allows staff to add new animals, manage listings, and approve or reject applications. This project is ideal for learning about web development in a realistic, real-world setting, as it involves creating user accounts, managing data, and performing CRUD operations. For a Java-based web version, Spring Boot would be useful for server-side logic, while MySQL can be used for database management. One challenge of this project is its complexity, as it requires effective database management and understanding frameworks and tools for building user-friendly interfaces.

The second project is a life choice game where players start with a set amount of health, money, and happiness. Players make choices through events such as investing in stocks, applying for jobs, buying lottery tickets, exercising, and seeking entertainment, each impacting their health, money, or happiness levels. At the end, the game evaluates how well players managed their resources. This project allows flexibility and creativity in designing events and outcomes and can vary in complexity depending on the number of events added. Since the game will be Java-based, it can be created as a web app with Java-based frameworks like Spring Boot for a web interface. However, setting up a fair scoring system and balancing game mechanics could take time and testing.