ZOO MANAGEMENT SYSTEM

GROUP 7

LUKMAN AROGUNDADE 986040

STANLEY JUILEN 986057

INTRODUCTION

Zoological gardens provide an opportunity to open a whole new world of curiosity and interest, and sensitize visitors regarding the value and need for conservation of wildlife.

Zoos were initially started for the entertainment of people. Gradually over the years, they have come to play an important role in conservation. The goal of zoos is the conservation of animals in the wild.

PROBLEM STATEMENT

The main problem and motivation is basically lack of information about various animals.

This project is based on the various information related to the animal and the workers, which are present in the zoo.

Such as when we visit any zoo then we are not aware that which animal is placed at which place.

For knowing this information, we look for a graph present in the zoo, but with this information system we can easily search the zoo by filling the name of the animal, such as Lion, Tiger, etc. then by running this query one can easily find the block in which the animal is present.

The system sends an auto-generated order to suppliers when the stock of food and medicine reach the deadline.

FEATURES OF THE SYSTEM

Administration features

- Add/Edit/Update/view employee details
- Add/Edit/update/view animal detail
- Add/Edit/update/view animal food and medicine
- Add/Edit/update/view cell details
- Add/Edit/update/view supplier details
- Add/Edit/update/view Holiday/Promo packages
- Add/Edit/update/view Zoo card
- Stock management (*simple operation*)
- Generate Report: Animals, visitors, payments, employees

Visitor/Employee features

- Search for animal
- Visitor make payment
- Make appointment to the Zoo

System feature

- Send auto mail to supplier

SOFTWARE DESIGN PATTERN USED

Template method, Singleton

HARDWARE & SOFTWARE USED

Software used:

IntelliJ/Eclipse, Windows and Linux

Front End:

JSP, HTML, Java script, CSS

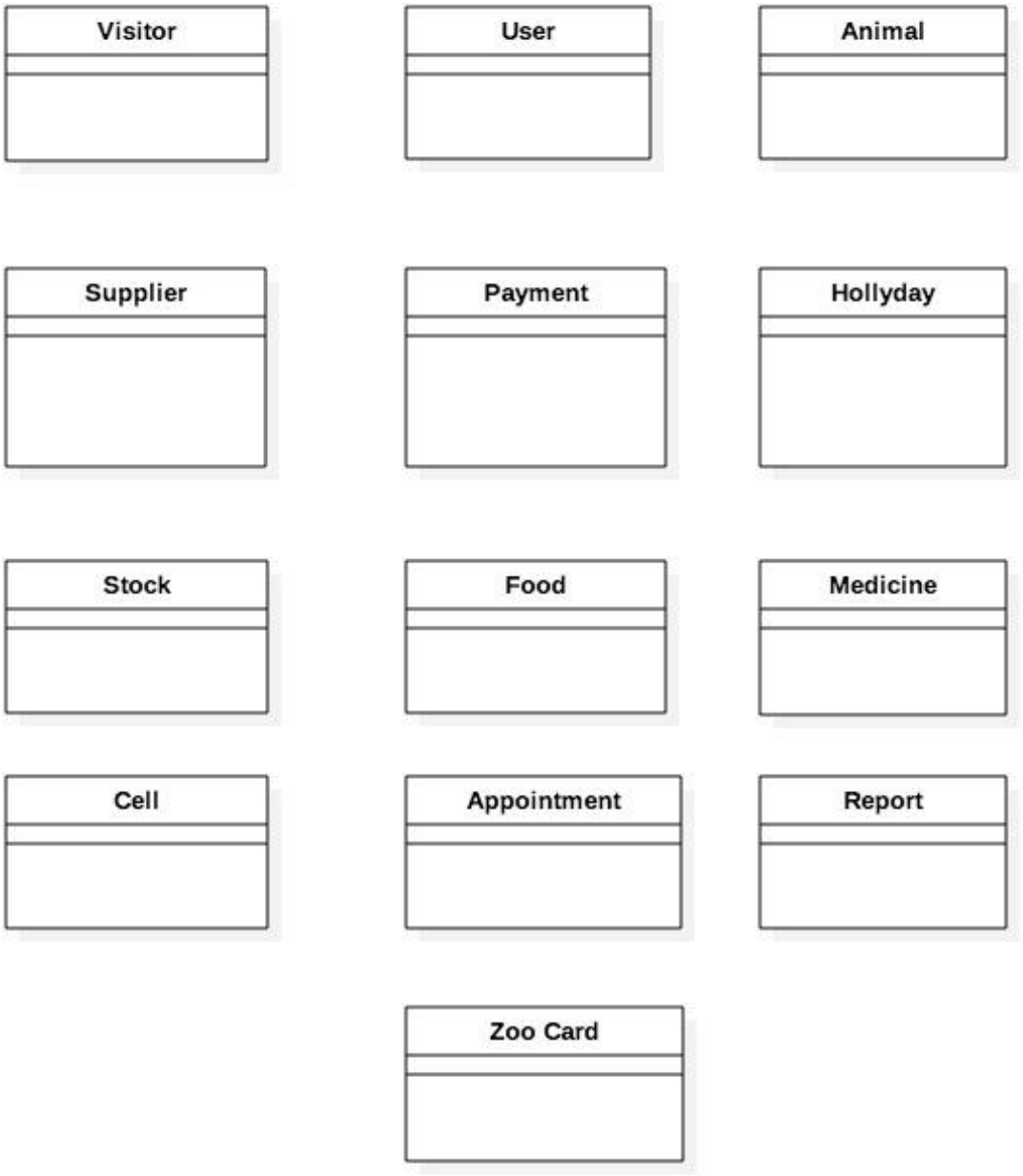
Back End:

Embedded Tomcat, Spring boot, Spring MVC, Spring security, Spring scheduler, RESTful microservice API, Data structure (Array List), JDBC template, JSON, Hibernate, GitHub

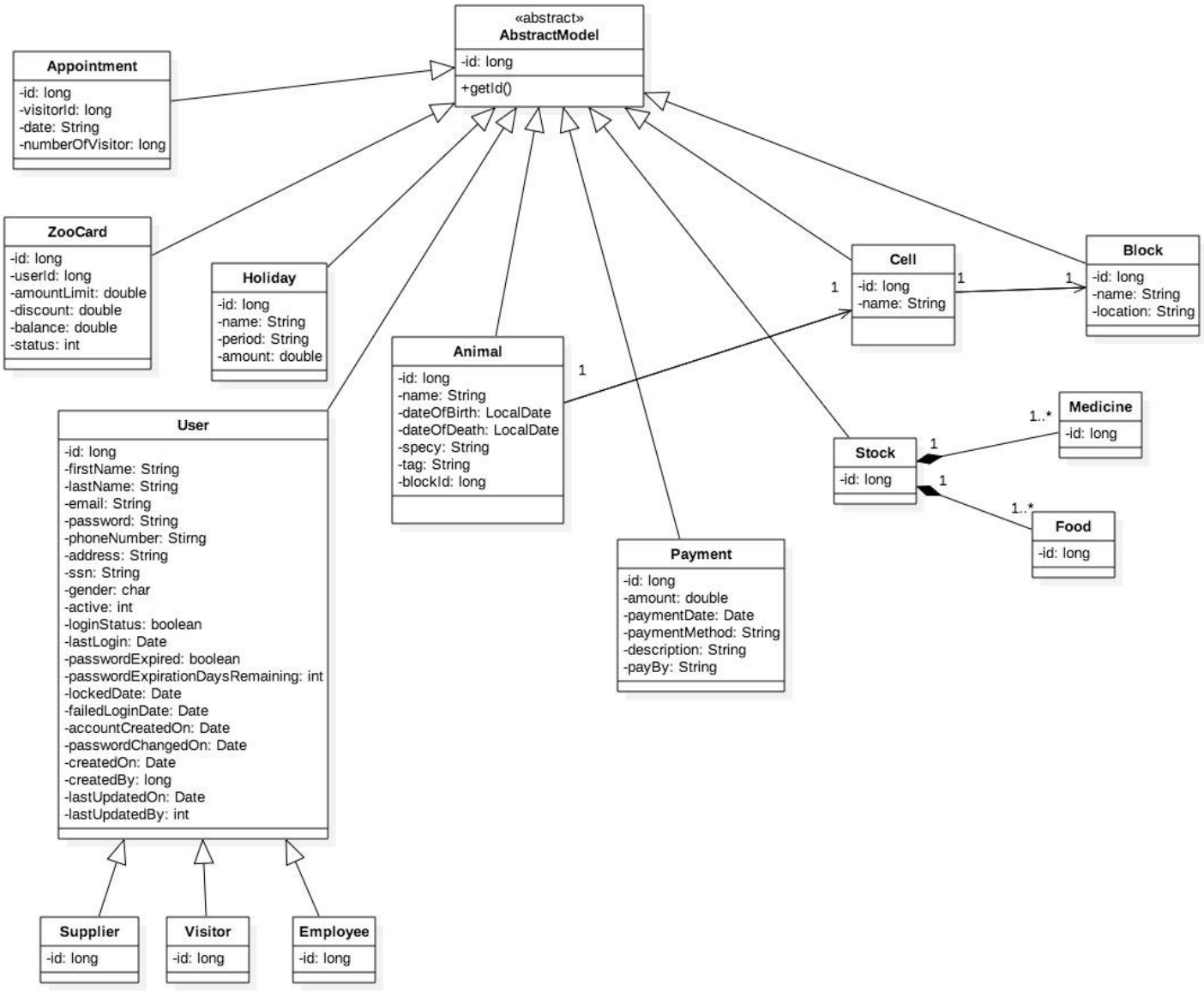
Database:

SQL Server

1- Identify Candidate Classes



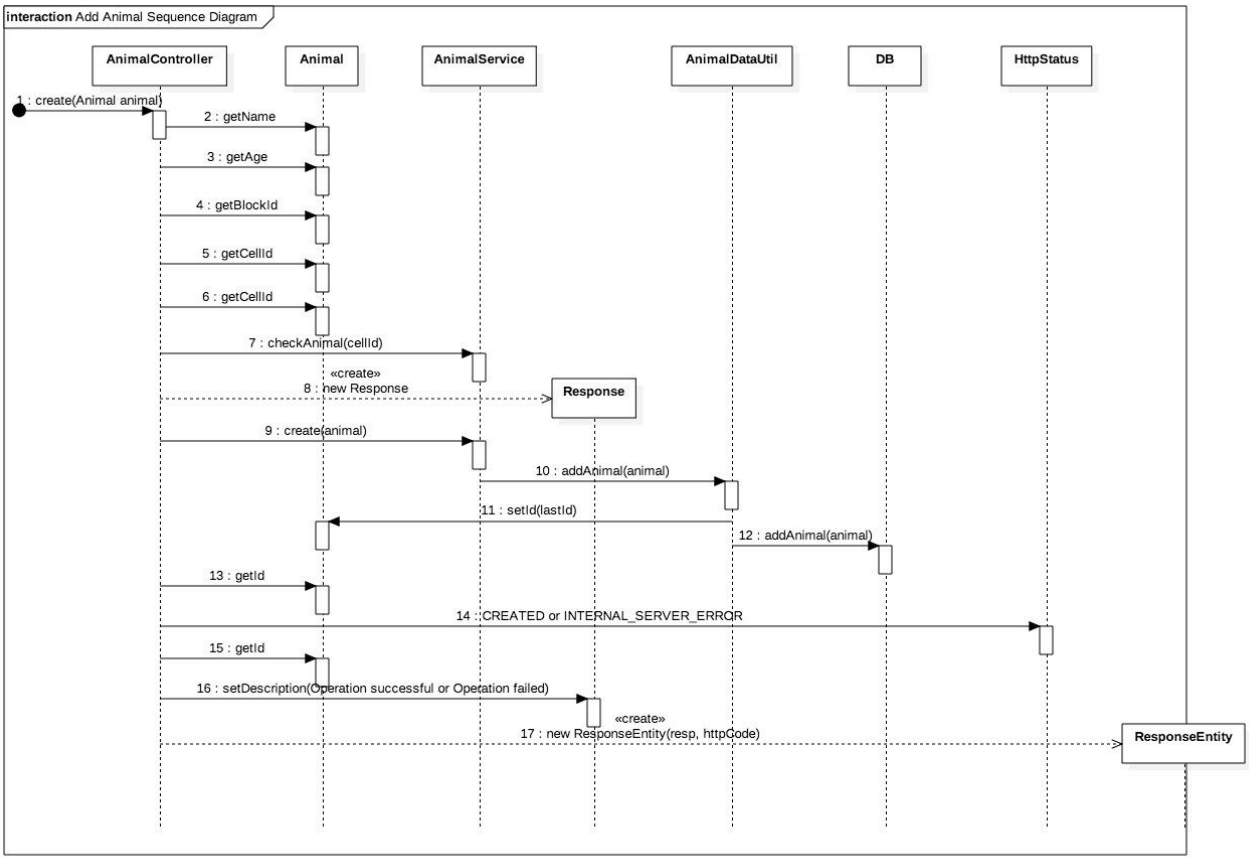
2- Create Class Diagram

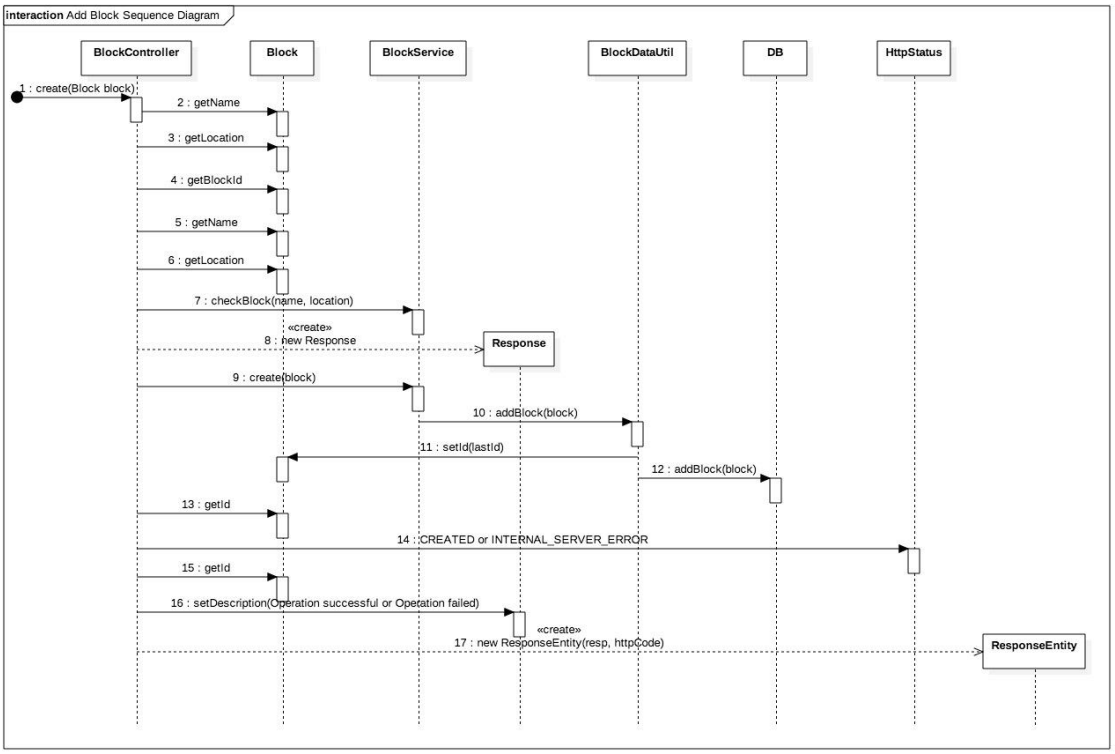


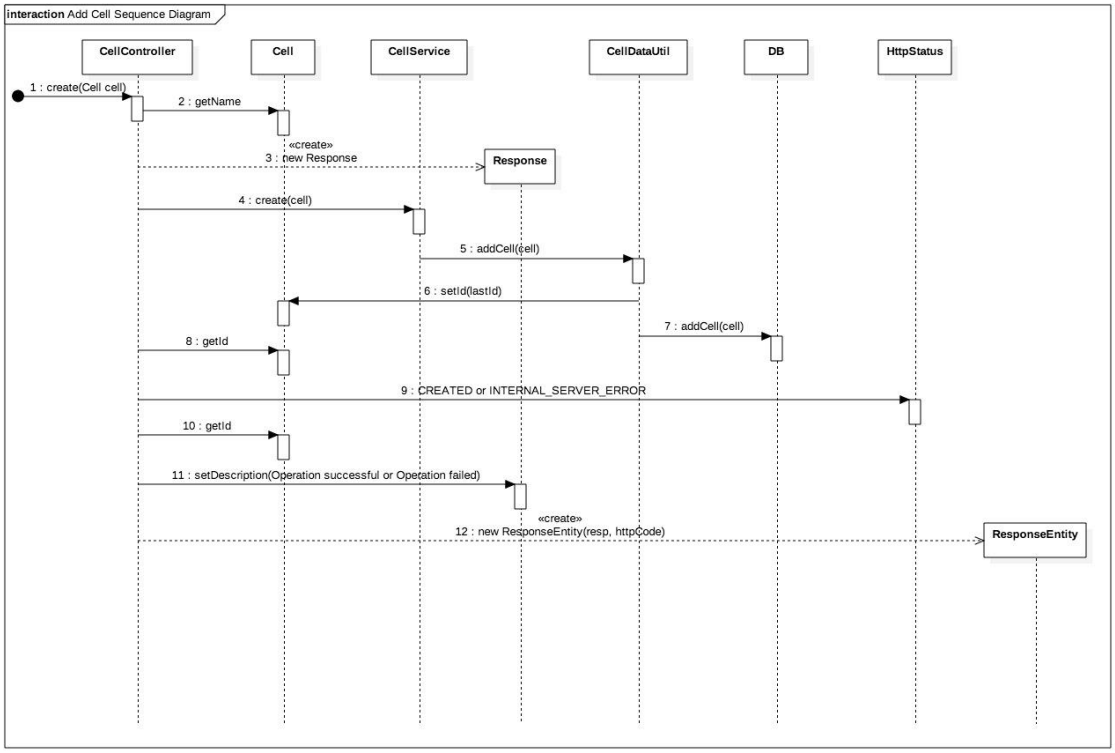
3- Data Dictionary

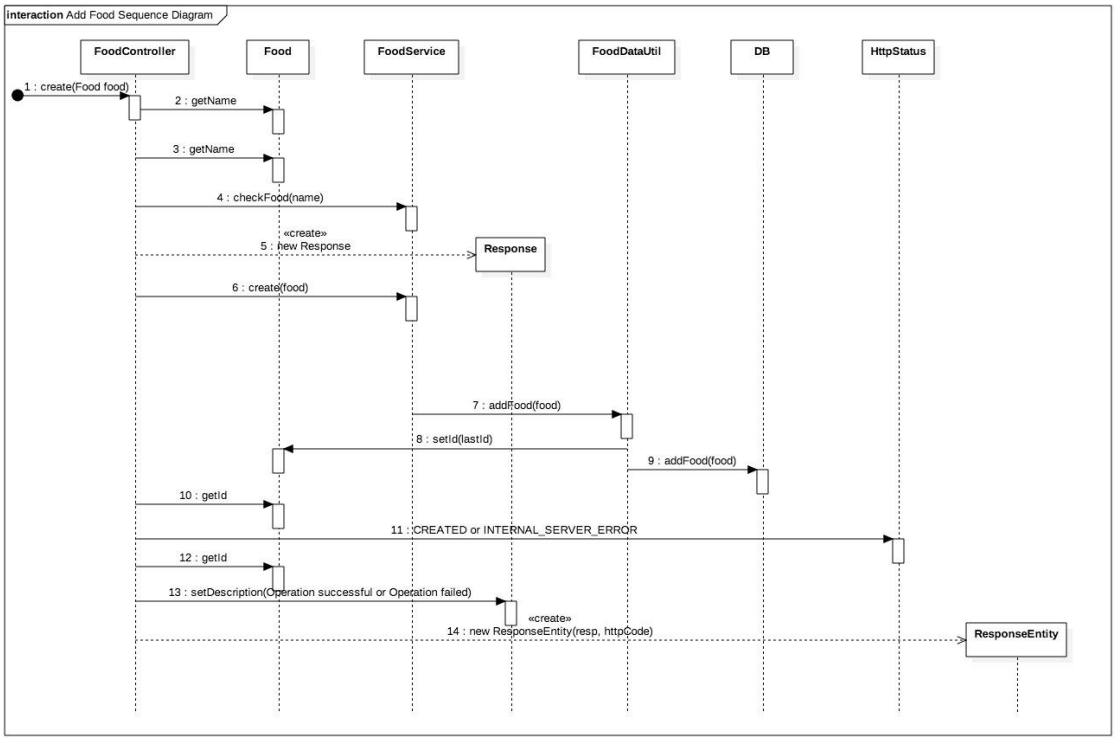
No	Class	Description
1	Animal	In the zoo we got different kind of animal. Ex: Lion, Elephant, Tiger
2	Cell	It is the room that we put the animal
3	Block	The block is the area that can content many cells. Ex: Block (East) content 20 cells
5	Stock	We by food and medicine by stock
6	Medicine	We use medicine to treat the animal or to prevent for some kind of disease
7	Food	We prepare food to nourish the animal
8	Appointment	A visitor can make appointment to come to the Zoo
9	User	Can be a Visitor, Supplier (who provide the stock) and Employee
10	Visitor	Person who come to visit the animal, the place
11	Supplier	The person who provide the Medicine and Food
12	Employee	It is the administrator of the system
13	Holiday	It is a promotion package that we offer to visitor, and have different kind of discount plan
14	Report	Report of the payment
15	Zoo Card	It is a gift card that we give to the visitor, so he can have a discount.

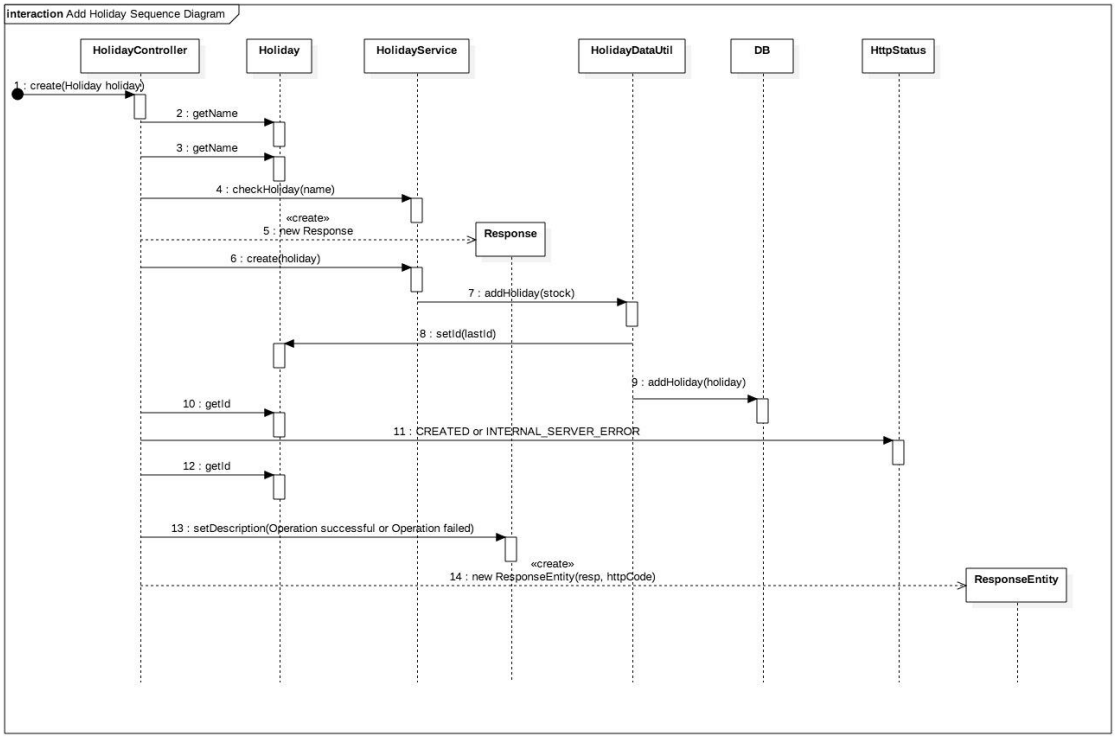
4- Create Sequence Diagram

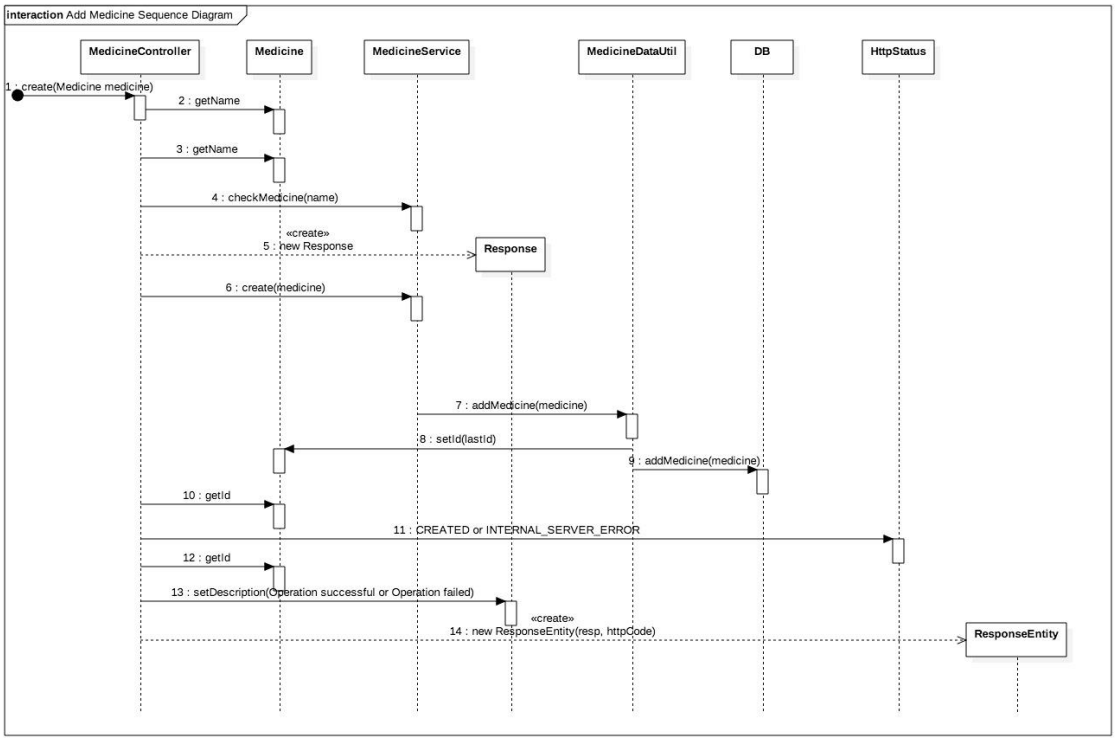


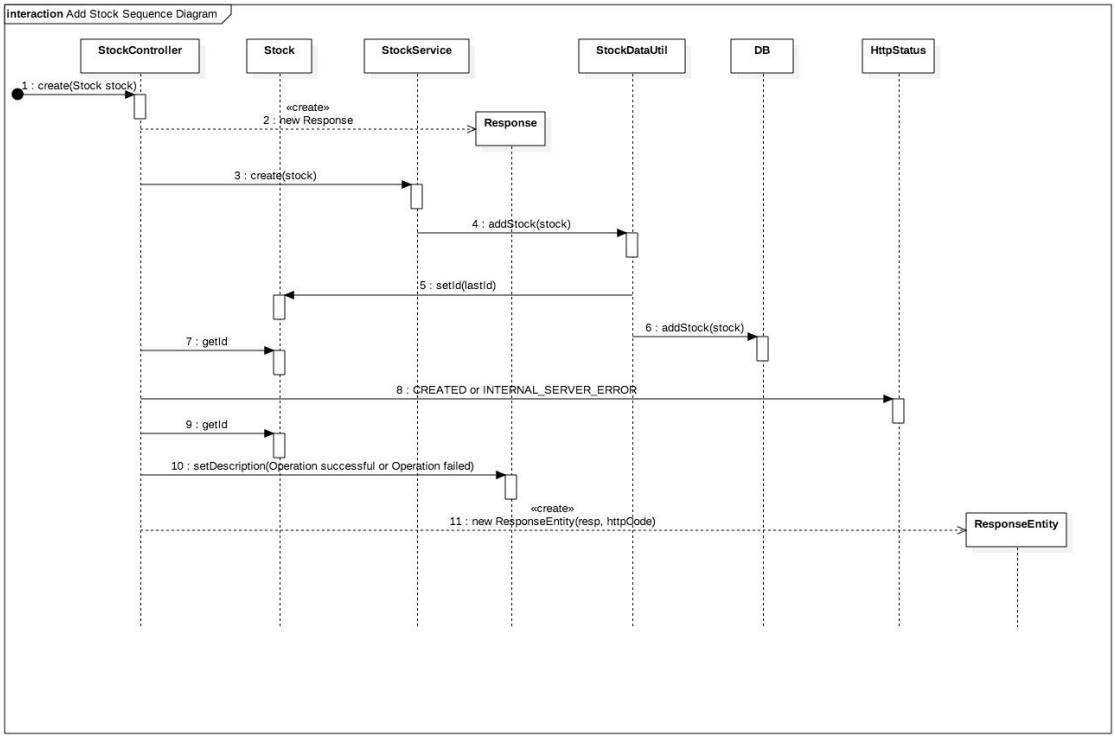



interaction Add Cell Sequence Diagram
CellController
Cell
CellService
CellDataUtil
DB
HttpStatus
1 : create(Cell cell)
2 : getName
«create»
3 : new Response
Response
4 : create(cell)
5 : addCell(cell)
6 : setId(lastId)
7 : addCell(cell)
8 : getId
9 : CREATED or INTERNAL_SERVER_ERROR
10 : getId
11 : setDescription(Operation successful or Operation failed)
«create»
12 : new ResponseEntity(resp, httpCode)
ResponseEntity


interaction Add Food Sequence Diagram
FoodController
Food
FoodService
FoodDataUtil
DB
HttpStatus
1 : create(Food food)
2 : getName
3 : getName
4 : checkFood(name)
«create»
5 : new Response
Response
6 : create(food)
7 : addFood(food)
8 : setId(lastId)
9 : addFood(food)
10 : getId
11 : CREATED or INTERNAL_SERVER_ERROR
12 : getId
13 : setDescription(Operation successful or Operation failed)
«create»
14 : new ResponseEntity(resp, httpCode)
ResponseEntity


interaction Add Holiday Sequence Diagram
HolidayController
Holiday
HolidayService
HolidayDataUtil
DB
HttpStatus
1 : create(Holiday holiday)
2 : getName
3 : getName
4 : checkHoliday(name)
«create»
5 : new Response
Response
6 : create(holiday)
7 : addHoliday(stock)
8 : setId(lastId)
9 : addHoliday(holiday)
10 : getId
11 : CREATED or INTERNAL_SERVER_ERROR
12 : getId
13 : setDescription(Operation successful or Operation failed)
«create»
14 : new ResponseEntity(resp, httpCode)
ResponseEntity


interaction Add Medicine Sequence Diagram
MedicineController
Medicine
MedicineService
MedicineDataUtil
DB
HttpStatus
1 : create(Medicine medicine)
2 : getName
3 : getName
4 : checkMedicine(name)
«create»
5 : new Response
Response
6 : create(medicine)
7 : addMedicine(medicine)
8 : setId(lastId)
9 : addMedicine(medicine)
10 : getId
11 : CREATED or INTERNAL_SERVER_ERROR
12 : getId
13 : setDescription(Operation successful or Operation failed)
«create»
14 : new ResponseEntity(resp, httpCode)
ResponseEntity


interaction Add Stock Sequence Diagram
StockController
Stock
StockService
StockDataUtil
DB
HttpStatus
1 : create(Stock stock)
«create»
2 : new Response
Response
3 : create(stock)
4 : addStock(stock)
5 : setId(lastId)
6 : addStock(stock)
7 : getId
8 : CREATED or INTERNAL_SERVER_ERROR
9 : getId
10 : setDescription(Operation successful or Operation failed)
«create»
11 : new ResponseEntity(resp, httpCode)
ResponseEntity

5- Database Diagram