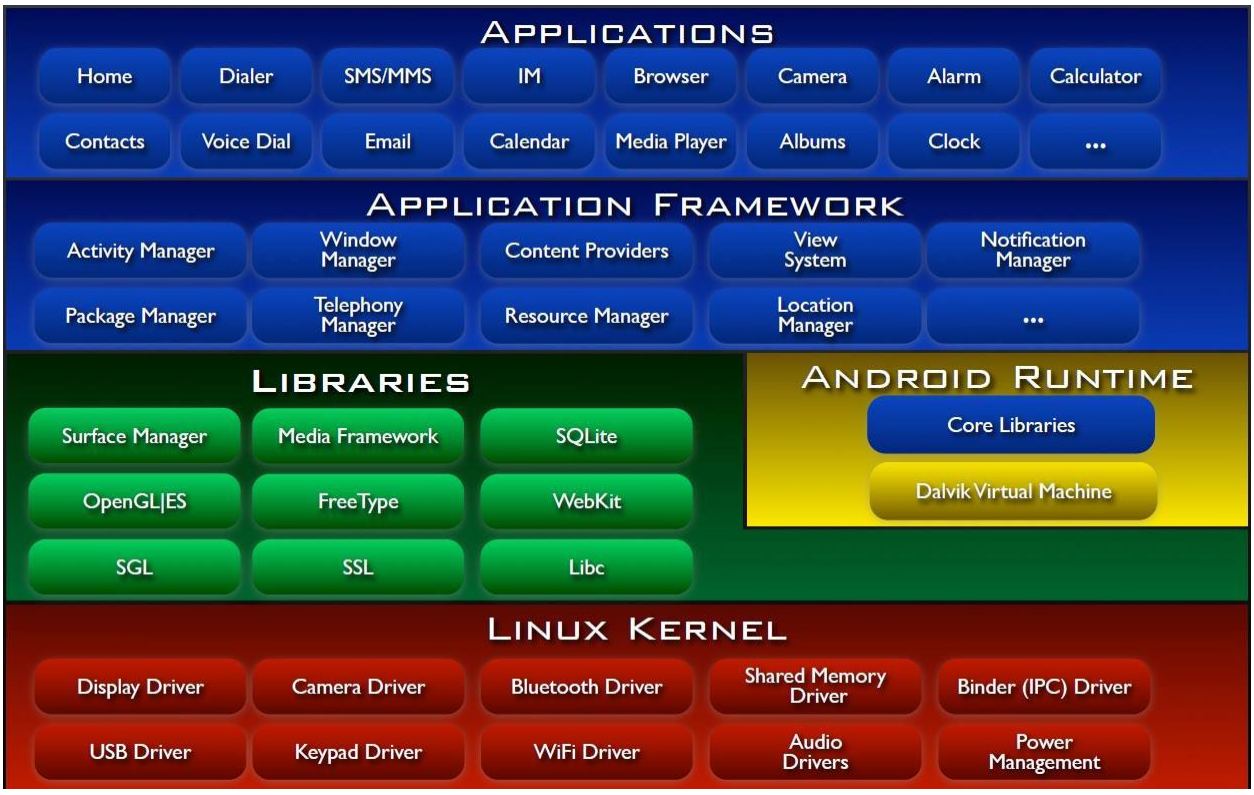
CS5/7319- Software Architecture and Design
Homework #3
(8 Points)

On Campus Due Date: 11:59pm on March 22, 2024
Off Campus Due Date: 11:59pm on March 23, 2024

Problem 1. Android Architecture (4 points)

In this assignment, you are going to get to know about Android and its design, as well as its application structure. You can see an overview of Android architecture in the diagram below. Our focus in this assignment is on the Applications and Application Framework layers. Application Framework in Android provides third-party developers with a set of tools, libraries, and APIs that allow them to design and implement their application in Android's ecosystem.



To ensure security of the system, Android maintains a mutually distrusting relation between applications in the systems. It runs each application in its own sandbox, an isolated virtual machine. However, this does not mean that applications cannot communicate with each other.

Android's application communication model further promotes the development of rich applications. Android developers can leverage existing data and services provided by other applications while still giving the impression of a single, seamless application. For example, a restaurant review application can ask other applications to display the restaurant's website, provide a map with the restaurant's location, and call the restaurant. This communication model reduces developer burden and promotes functionality reuse. Android achieves this by dividing applications into components and providing a message passing system so that

components can communicate within and across application boundaries.

Android provides a sophisticated message passing system, in which message objects called Intents are used to link applications. An Intent is a message that declares a recipient and optionally includes data. An Intent can be thought of as a self-contained object that specifies a remote procedure to invoke and includes the associated arguments. Applications use Intents for both inter-application communication and intra-application communication. Additionally, the operating system sends Intents to applications as event notifications. Some of these event notifications are system-wide events that can only be sent by the operating system. We call these messages system broadcast Intents.

Intents can be used for explicit or implicit communication. An explicit Intent specifies that it should be delivered to a particular application specified by the Intent, whereas an implicit Intent requests delivery to any application that supports a desired operation. In other words, an explicit Intent identifies the intended recipient by name, whereas an implicit Intent leaves it up to the Android platform to determine which application(s) should receive the Intent. For example, consider an application that stores contact information. When the user clicks on a contact's street address, the contacts application needs to ask another application to display a map of that location. To achieve this, the contacts application could send an explicit Intent directly to Google Maps. Alternatively, the contacts application could send an implicit Intent that would be delivered to any application that says it provides mapping functionality (e.g., Yahoo! Maps or Bing Maps). Using an explicit Intent guarantees that the Intent is delivered to the intended recipient, whereas implicit Intents allow for late runtime binding between different applications.

Since Intents are exchanged between application components, we call this type of communication Inter-Component Communication (ICC). To know more about components and activities, you can read Android documentation on components and activities. Alternatively, you can take a brief look at Android applications overview. This type of communication is the result of design decisions that Android system architects have taken. As we know, a design decision may have deep effects on various aspects of a software system. One of them is security. In this assignment, you are going to find some security issues caused by Android's communication methods. These security issues can be exploited by attacks that are called ICC attacks.

1. Which architecture style is applied to design for Android? And why? (1 point)

2. What potential vulnerabilities could implicit intent bring up? At least two vulnerabilities (1 point)

3. How to avoid the vulnerabilities you mentioned in Q2? (2 points)

Submit a PDF file named "**Android.pdf**" via Canvas Homework 3 submission link.

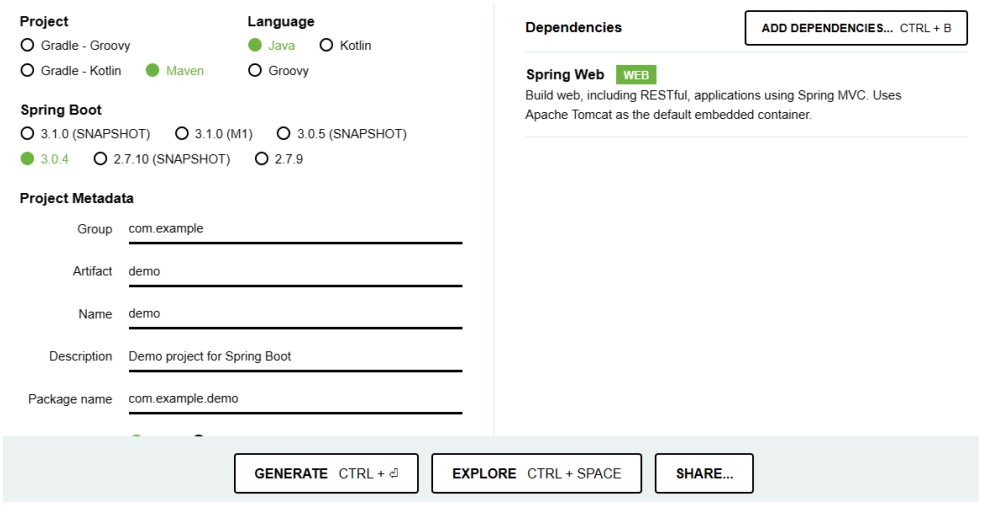
Problem 2. Container in REST API – Java Spring Boot (4 points)

Part 1 – REST API - Hello World: (Return 'Hello World!' and 'Welcome to CS 7319!') (0.5pt)

Description:

The purpose of this part is to get familiar with Java, IntelliJ, and REST API.

1. Install Java SDK 11 and IntelliJ in your computer. (Google the web/YouTube for these.)
2. Use Spring Initializer to create a Java project. (https://start.spring.io/)
   Select Maven



(https://www.youtube.com/watch?v=2gvd-UiAVfk)

3. Create a Java class called WelcomeController.

4. Add code to the above project to print 'Hello World!' and 'Welcome to CS 7319!' on the root ("/") and "/welcome" page respectively. (A little bit different from the above YouTube.)

5.
```java
@RestController
public class WelcomeController {
    @GetMapping("/")
    public String HelloWorld(){
        return "Hello World";
    }

    @GetMapping("/weloome")
    public String Welcome(){
        return "Welcome to CS7319";
    }
}
```

6. Take screenshots of the above pages and put them into a PDF named "HelloWorld.pdf".
7. Zip the source codes and name the file "HelloWorld.zip".
8. Submit the above two files to Canvas.

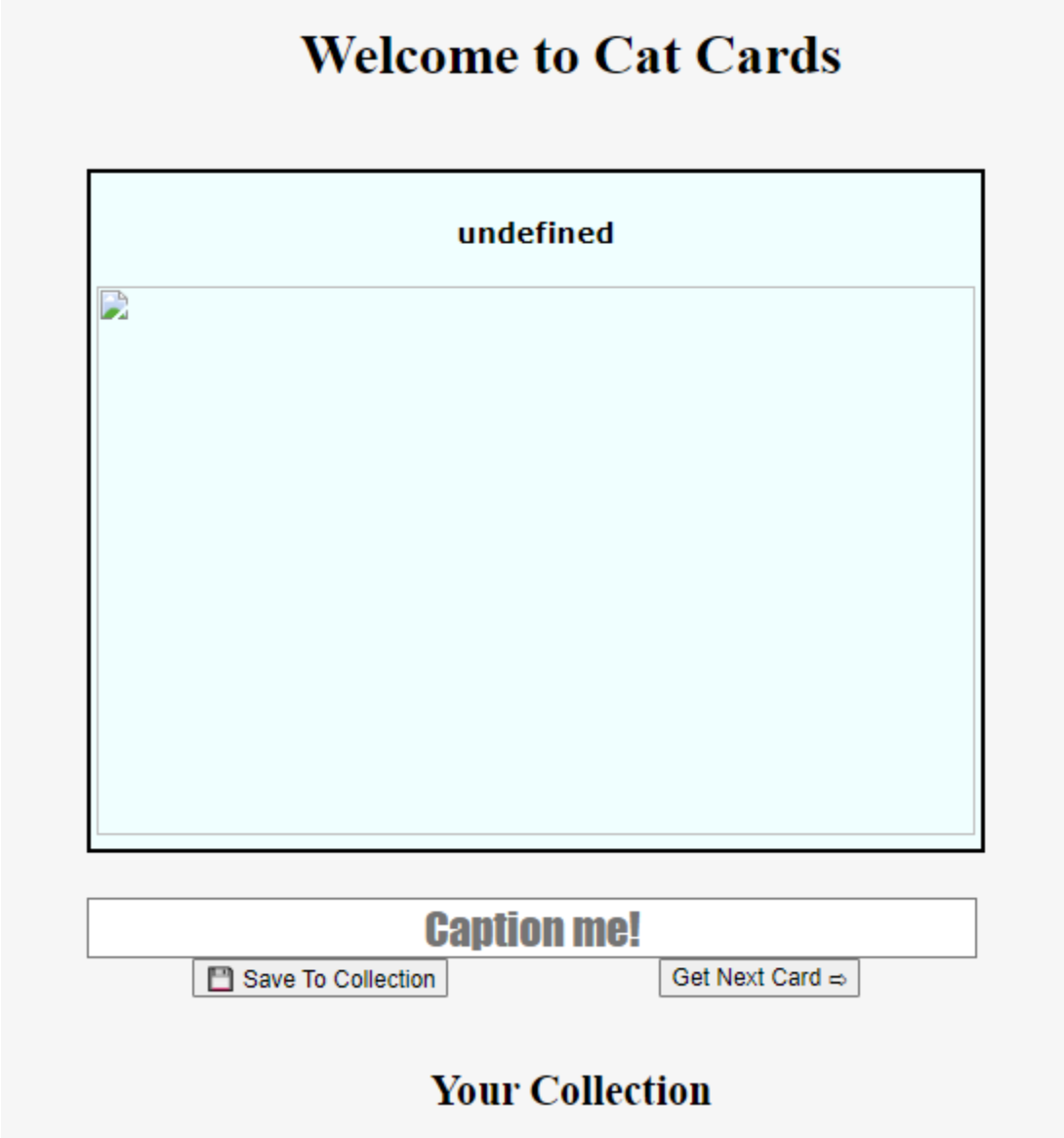
Part 2 – REST API - Cat Container: (Return Cat facts and pics.) (3.5pts)

Description:

The purpose of this part is to get familiar with the using a container in REST API – get/post/put/delete and database operations (Postgres).

1. Install Postgres (PgAdmin) in your computer. Use port 5432 and postgres1 as password for postgres – i.e. username/password – postgres/postgres1 - Postgres server 12 will be fine. (https://www.youtube.com/watch?v=0n41UTkOBb0)
2. Open the Java base code for Part2 in IntelliJ – (unzip catContainer.zip).
3. Open PgAdmin and create a database called 'catcards'.
4. Open the query tool under 'catcards' and run the catcards.sql to create the database tables for this project.
5. Read the Readme.md file to create a container and fix the missing codes.
   5a. You will create a CatFact container under 'model' and fix the missing codes in this app.
6. Take a screenshot of the catcards and put them in a PDF named "CatCards.pdf" after you fix the codes.
7. Zip the source codes and name the file "CatCards.zip".
8. Submit the above two files to Canvas.

The Base code will show this on the web page:



After you fix the missing codes, it should display something like the following: