ICIN BANK-Online Banking

Contents

Contents:

1. Project Modules:

- Node js.
- Visual Studios.
- Angular 13.3.1
- Angular .json
- HTML
- Css

- Java Scriot
- karma configuration model.
- Sql Data Base
- PowerShell

2. Features of the application:

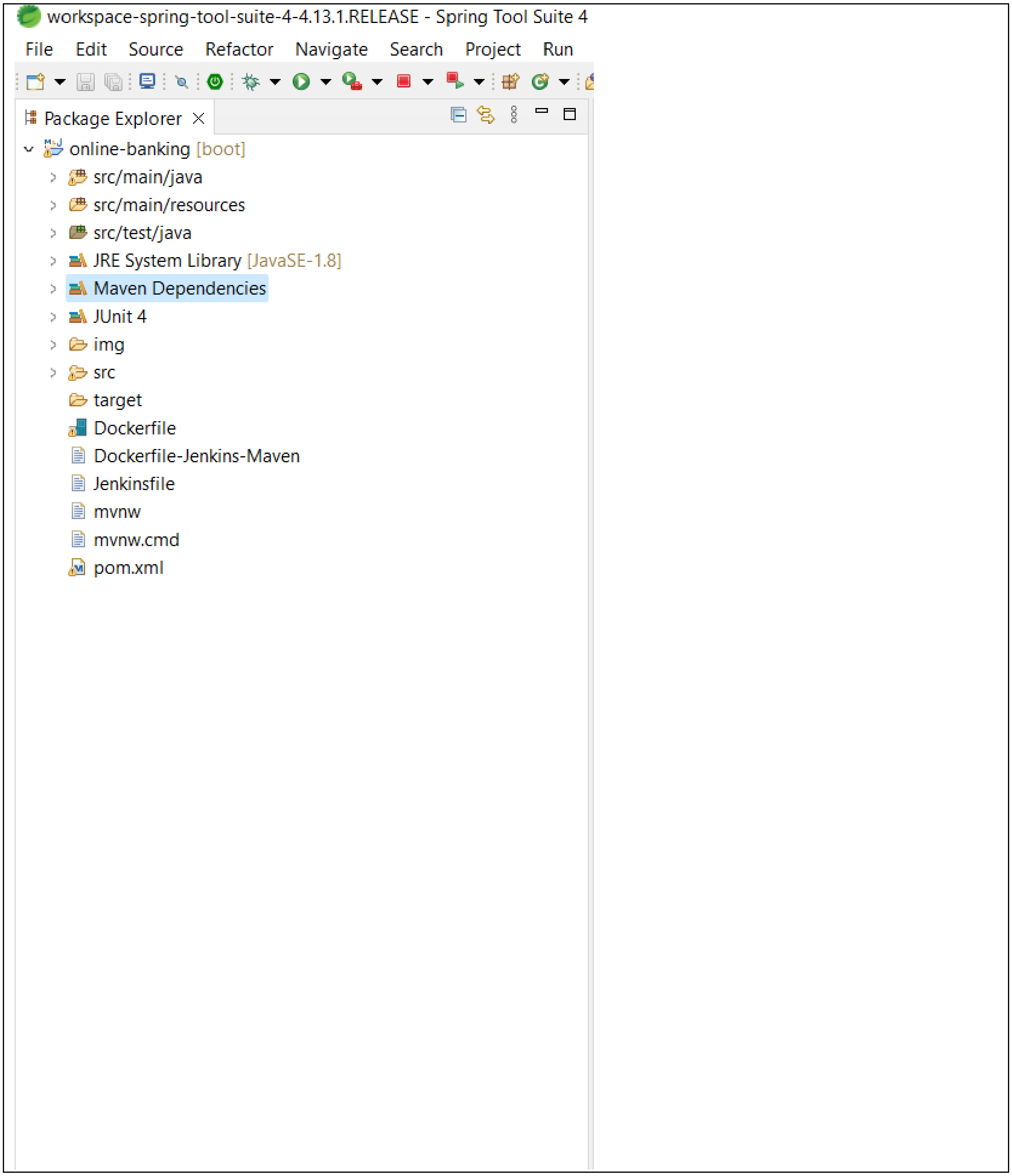
- Registration
- Login
- Account transactions
- Transfers
- Savings details
- Profile settings
- Requesting cheque books

3. GitHub Link.

https://github.com/Sai153793/capstone-project-ICIN-online-banking.git

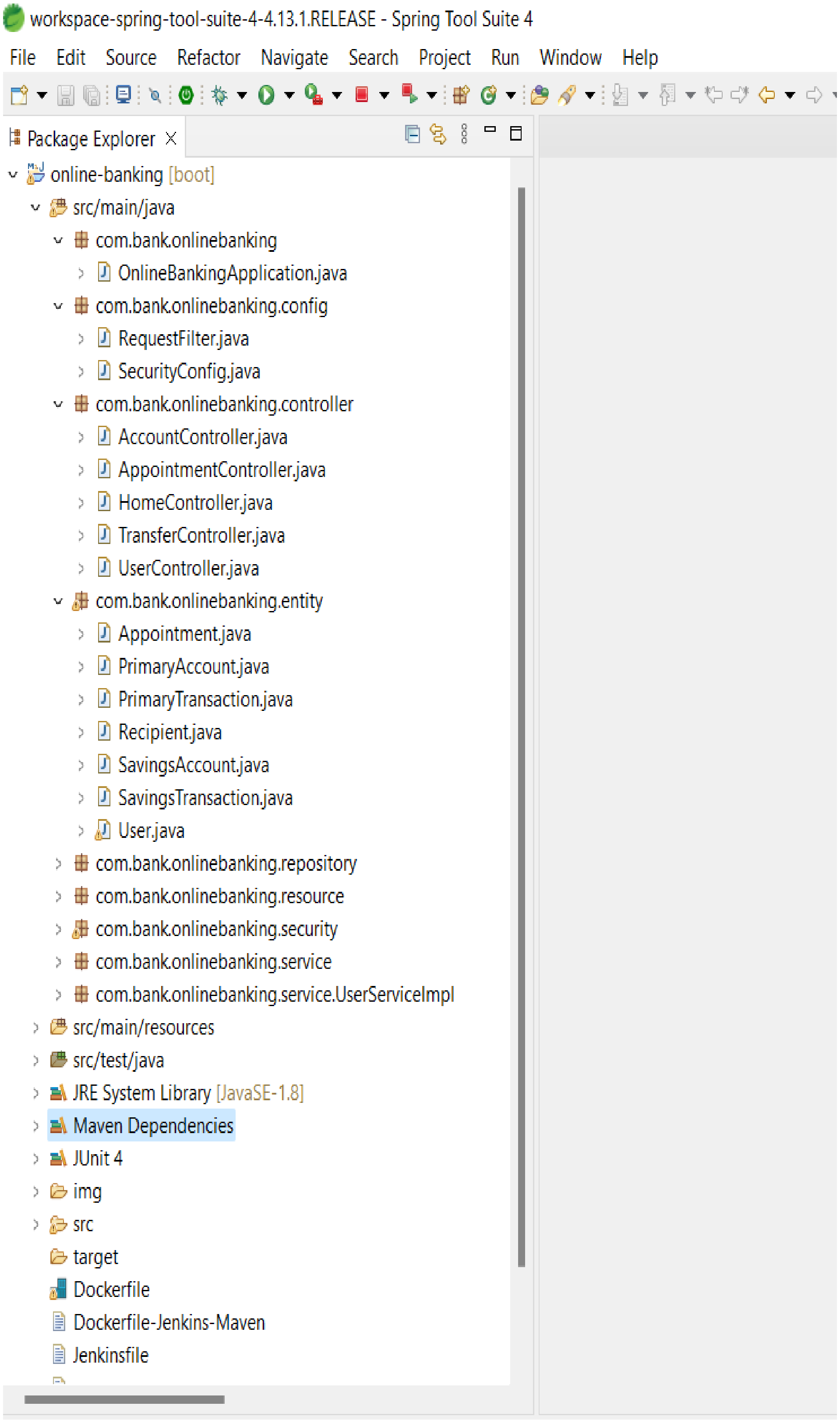
4. Folder Structure:

Folder Structure

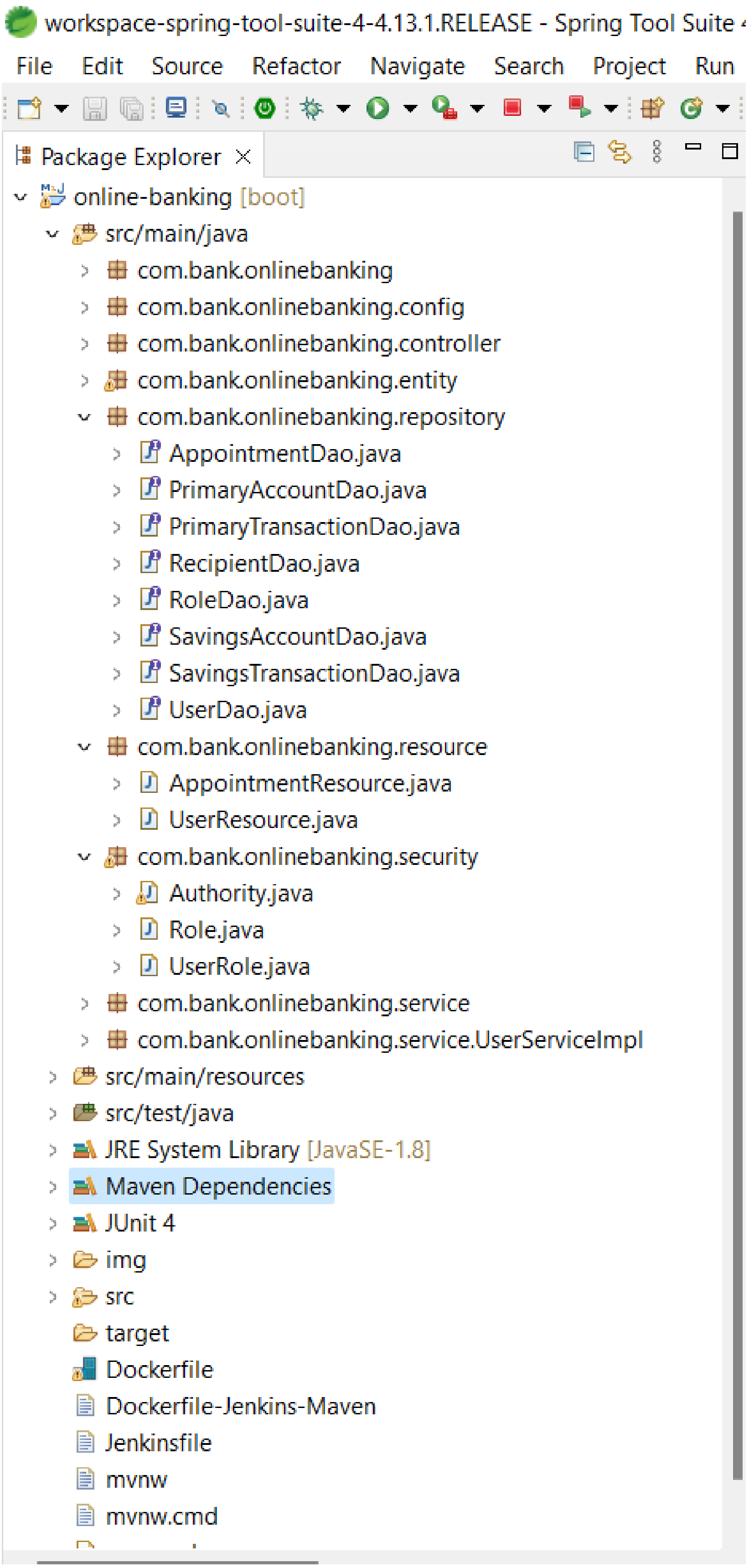


workspace-spring-tool-suite-4-4.13.1.RELEASE - Spring Tool Suite 4

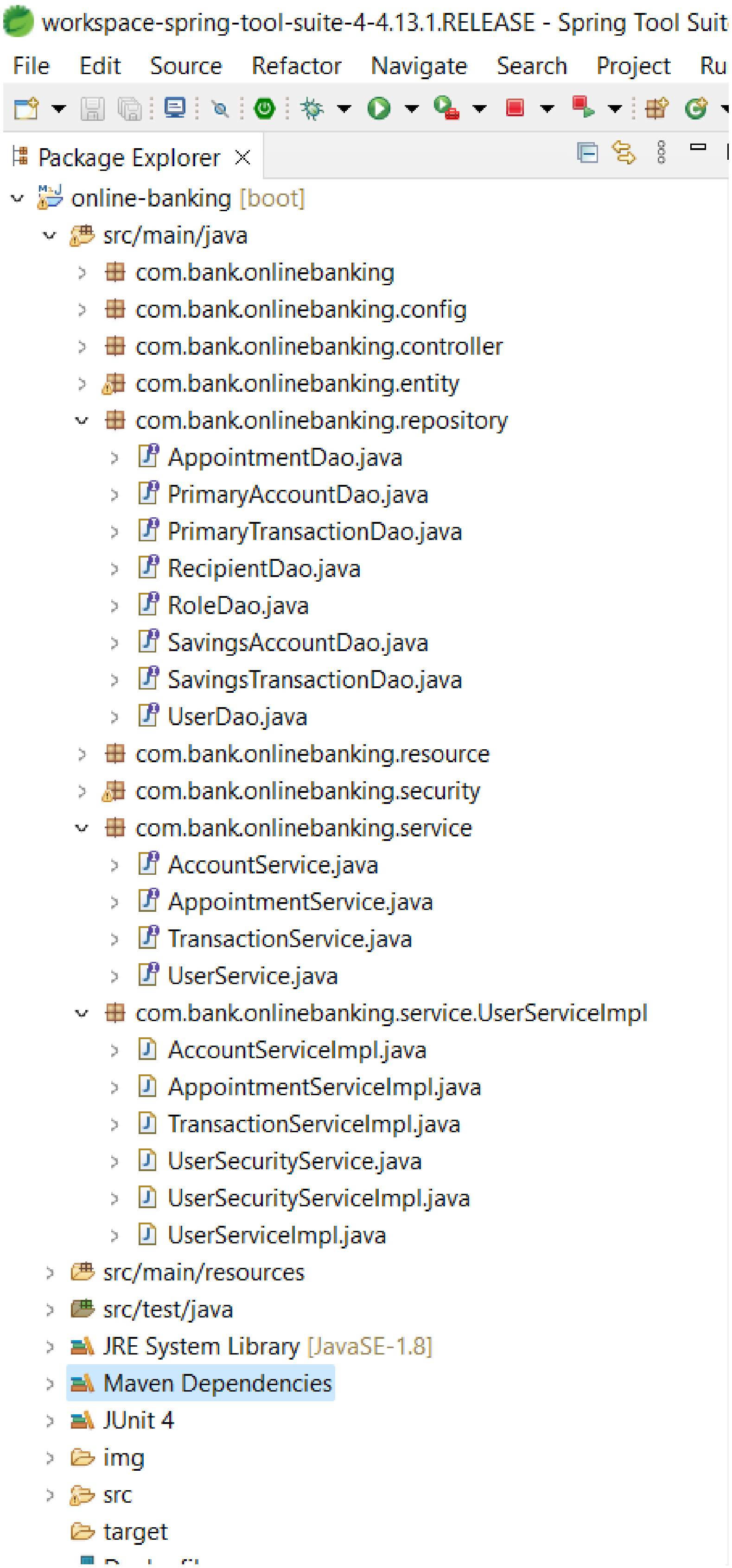
online-banking [boot]
- src/main/java
- src/main/resources
- src/test/java
- JRE System Library [JavaSE-1.8]
- Maven Dependencies
- JUnit 4
- img
- src
- target
- Dockerfile
- Dockerfile-Jenkins-Maven
- Jenkinsfile
- mvnw
- mvnw.cmd
- pom.xml



```
online-banking [boot]
  src/main/java
    com.bank.onlinebanking
      OnlineBankingApplication.java
    com.bank.onlinebanking.config
      RequestFilter.java
      SecurityConfig.java
    com.bank.onlinebanking.controller
      AccountController.java
      AppointmentController.java
      HomeController.java
      TransferController.java
      UserController.java
    com.bank.onlinebanking.entity
      Appointment.java
      PrimaryAccount.java
      PrimaryTransaction.java
      Recipient.java
      SavingsAccount.java
      SavingsTransaction.java
      User.java
    com.bank.onlinebanking.repository
    com.bank.onlinebanking.resource
    com.bank.onlinebanking.security
    com.bank.onlinebanking.service
    com.bank.onlinebanking.service.UserServiceImpl
  src/main/resources
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  JUnit 4
  img
  src
  target
  Dockerfile
  Dockerfile-Jenkins-Maven
  Jenkinsfile
```

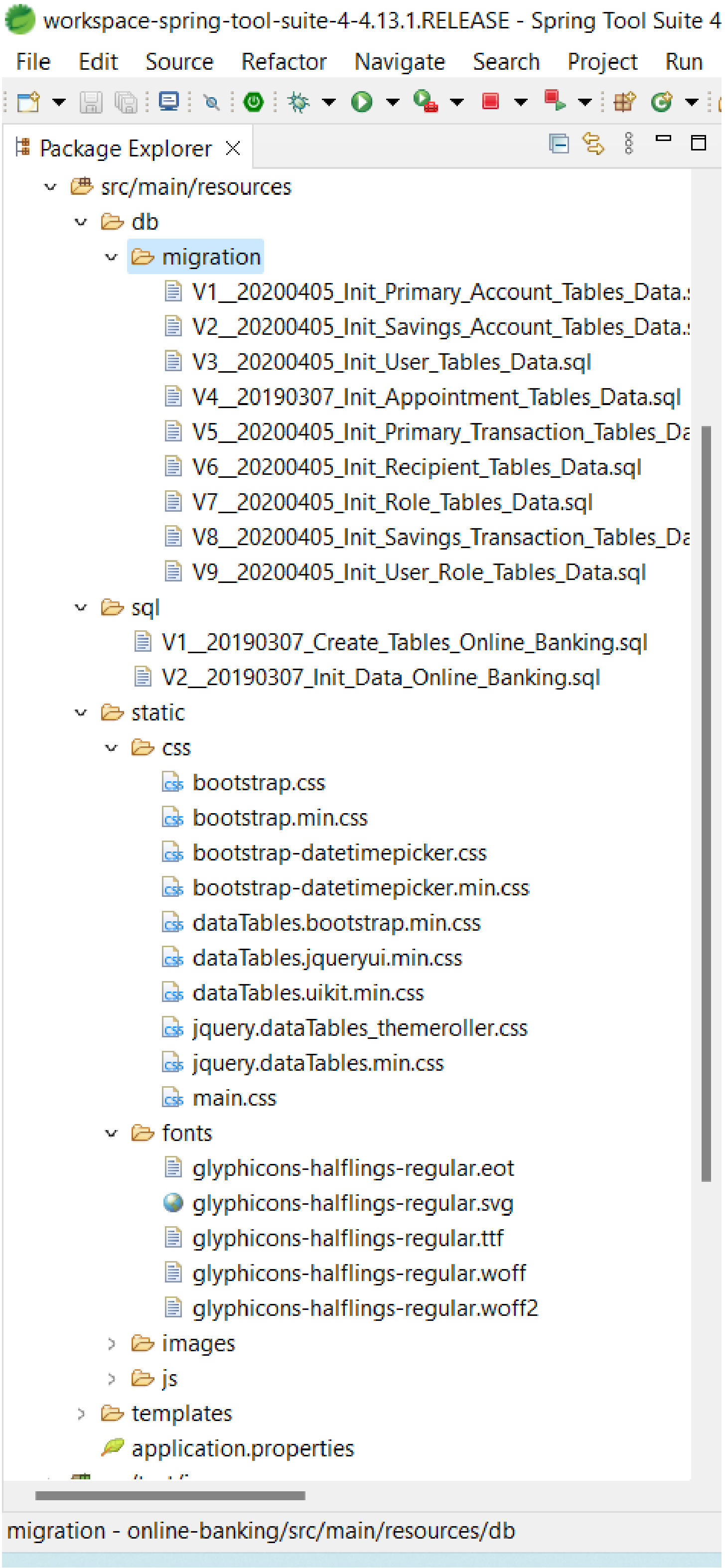

workspace-spring-tool-suite-4-4.13.1.RELEASE - Spring Tool Suite
File
Edit
Source
Refactor
Navigate
Search
Project
Run
Package Explorer

```
online-banking [boot]
  src/main/java
    com.bank.onlinebanking
    com.bank.onlinebanking.config
    com.bank.onlinebanking.controller
    com.bank.onlinebanking.entity
    com.bank.onlinebanking.repository
      AppointmentDao.java
      PrimaryAccountDao.java
      PrimaryTransactionDao.java
      RecipientDao.java
      RoleDao.java
      SavingsAccountDao.java
      SavingsTransactionDao.java
      UserDao.java
    com.bank.onlinebanking.resource
      AppointmentResource.java
      UserResource.java
    com.bank.onlinebanking.security
      Authority.java
      Role.java
      UserRole.java
    com.bank.onlinebanking.service
    com.bank.onlinebanking.service.UserServiceImpl
  src/main/resources
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  JUnit 4
  img
  src
  target
  Dockerfile
  Dockerfile-Jenkins-Maven
  Jenkinsfile
  mvnw
  mvnw.cmd
```



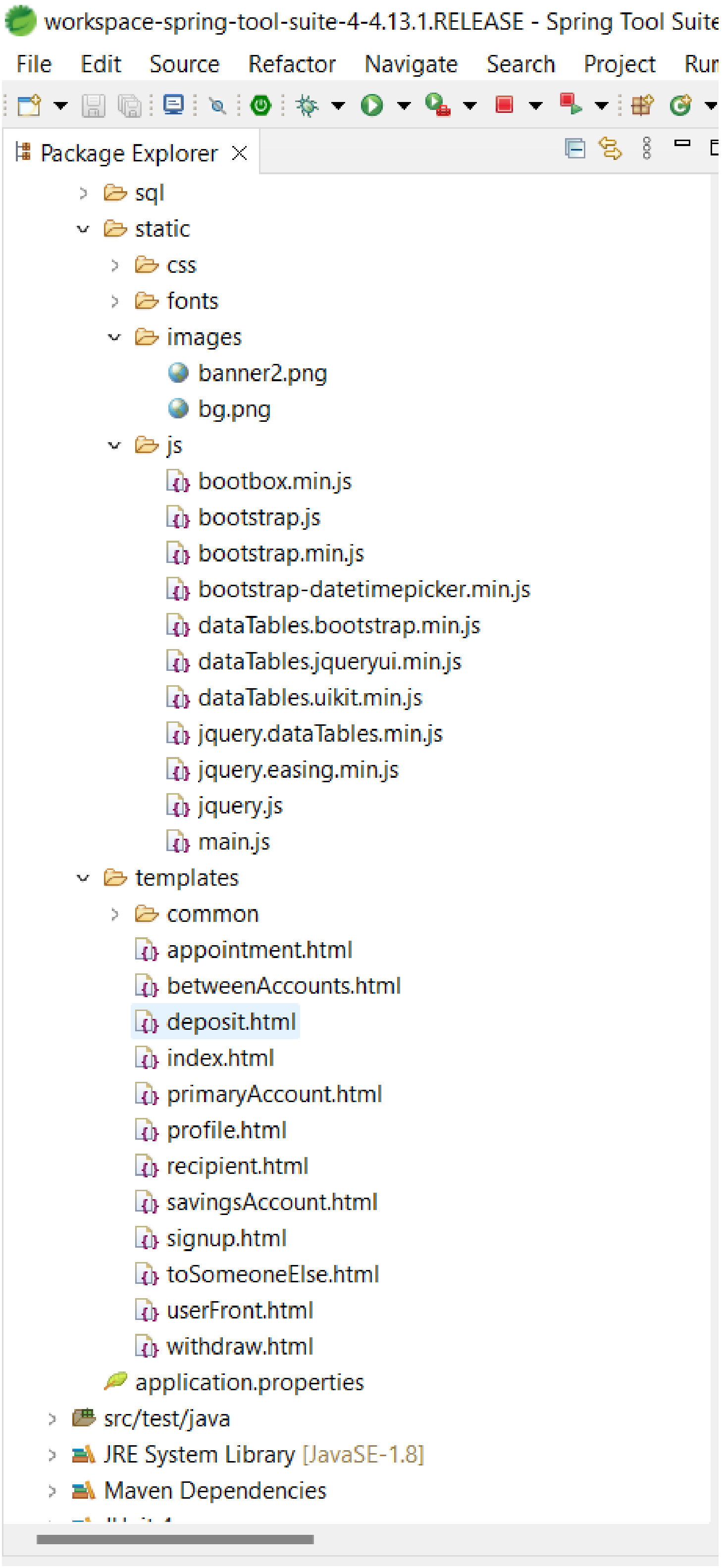
workspace-spring-tool-suite-4-4.13.1.RELEASE - Spring Tool Suit
File Edit Source Refactor Navigate Search Project Ru
Package Explorer

```
online-banking [boot]
  src/main/java
    com.bank.onlinebanking
    com.bank.onlinebanking.config
    com.bank.onlinebanking.controller
    com.bank.onlinebanking.entity
    com.bank.onlinebanking.repository
      AppointmentDao.java
      PrimaryAccountDao.java
      PrimaryTransactionDao.java
      RecipientDao.java
      RoleDao.java
      SavingsAccountDao.java
      SavingsTransactionDao.java
      UserDao.java
    com.bank.onlinebanking.resource
    com.bank.onlinebanking.security
    com.bank.onlinebanking.service
      AccountService.java
      AppointmentService.java
      TransactionService.java
      UserService.java
    com.bank.onlinebanking.service.UserServiceImpl
      AccountServiceImpl.java
      AppointmentServiceImpl.java
      TransactionServiceImpl.java
      UserSecurityService.java
      UserSecurityServiceImpl.java
      UserServiceImpl.java
  src/main/resources
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  JUnit 4
  img
  src
  target
```

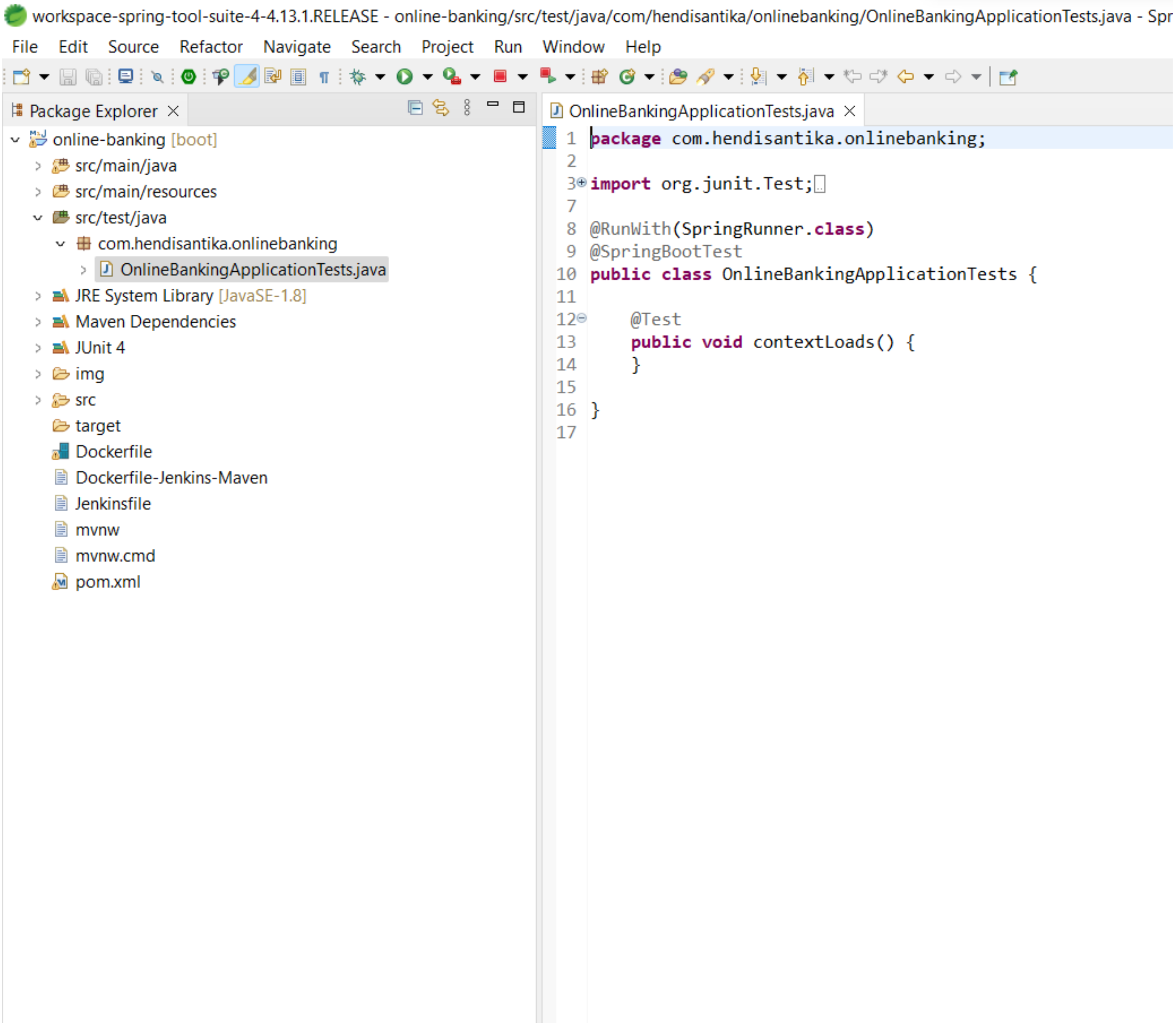

workspace-spring-tool-suite-4-4.13.1.RELEASE - Spring Tool Suite 4
File
Edit
Source
Refactor
Navigate
Search
Project
Run
Package Explorer

```
src/main/resources
  db
    migration
      V1__20200405_Init_Primary_Account_Tables_Data.
      V2__20200405_Init_Savings_Account_Tables_Data.
      V3__20200405_Init_User_Tables_Data.sql
      V4__20190307_Init_Appointment_Tables_Data.sql
      V5__20200405_Init_Primary_Transaction_Tables_Da
      V6__20200405_Init_Recipient_Tables_Data.sql
      V7__20200405_Init_Role_Tables_Data.sql
      V8__20200405_Init_Savings_Transaction_Tables_Da
      V9__20200405_Init_User_Role_Tables_Data.sql
  sql
    V1__20190307_Create_Tables_Online_Banking.sql
    V2__20190307_Init_Data_Online_Banking.sql
  static
    css
      bootstrap.css
      bootstrap.min.css
      bootstrap-datetimepicker.css
      bootstrap-datetimepicker.min.css
      dataTables.bootstrap.min.css
      dataTables.jqueryui.min.css
      dataTables.uikit.min.css
      jquery.dataTables_themeroller.css
      jquery.dataTables.min.css
      main.css
    fonts
      glyphicons-halflings-regular.eot
      glyphicons-halflings-regular.svg
      glyphicons-halflings-regular.ttf
      glyphicons-halflings-regular.woff
      glyphicons-halflings-regular.woff2
    images
    js
  templates
  application.properties
```

migration - online-banking/src/main/resources/db

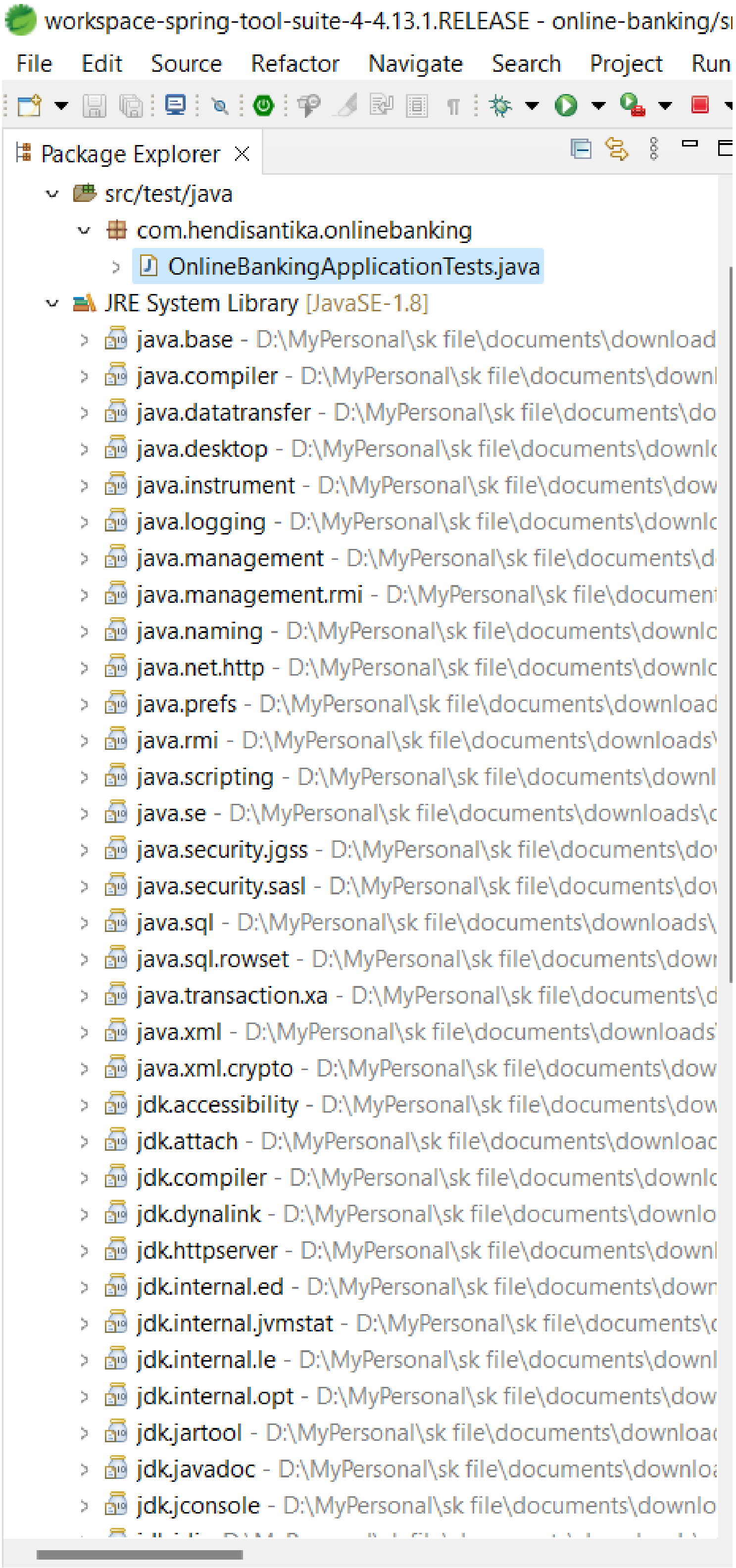

workspace-spring-tool-suite-4-4.13.1.RELEASE - Spring Tool Suite
File
Edit
Source
Refactor
Navigate
Search
Project
Run
Package Explorer

```
    sql
    static
        css
        fonts
        images
            banner2.png
            bg.png
        js
            bootbox.min.js
            bootstrap.js
            bootstrap.min.js
            bootstrap-datetimepicker.min.js
            dataTables.bootstrap.min.js
            dataTables.jqueryui.min.js
            dataTables.uikit.min.js
            jquery.dataTables.min.js
            jquery.easing.min.js
            jquery.js
            main.js
    templates
        common
        appointment.html
        betweenAccounts.html
        deposit.html
        index.html
        primaryAccount.html
        profile.html
        recipient.html
        savingsAccount.html
        signup.html
        toSomeoneElse.html
        userFront.html
        withdraw.html
    application.properties
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
```

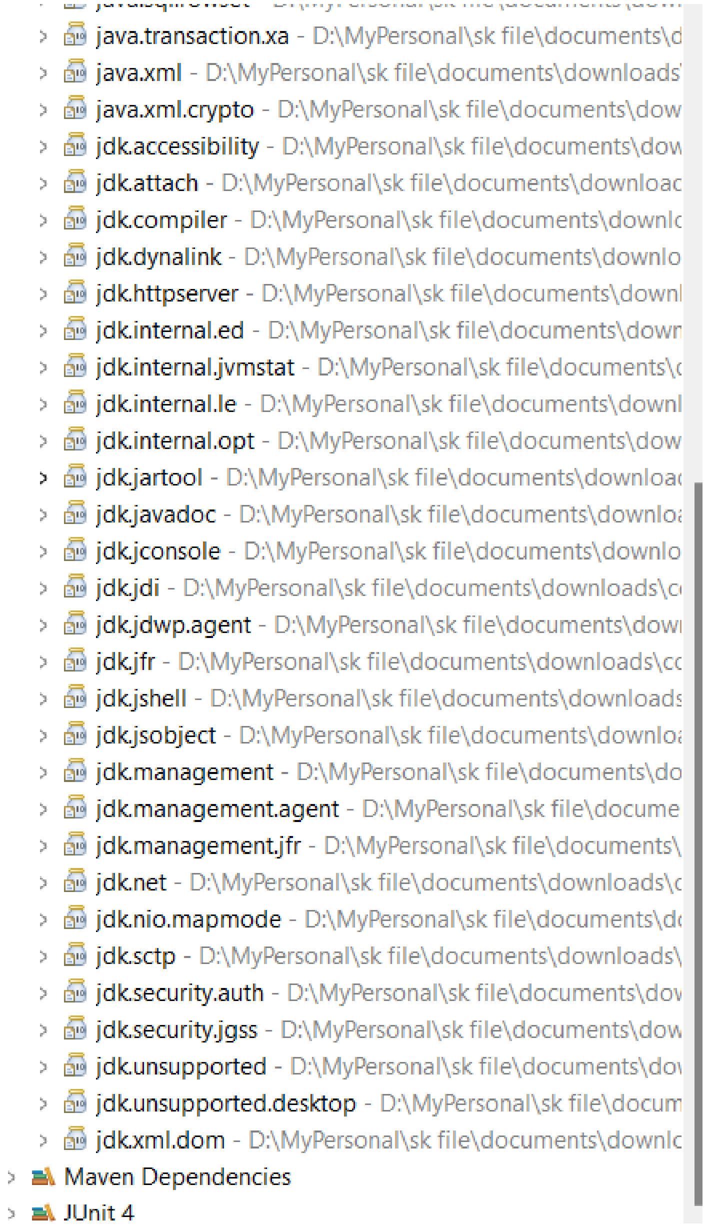


workspace-spring-tool-suite-4-4.13.1.RELEASE - online-banking/src/test/java/com/hendisantika/onlinebanking/OnlineBankingApplicationTests.java - Spr
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
online-banking [boot]
src/main/java
src/main/resources
src/test/java
com.hendisantika.onlinebanking
OnlineBankingApplicationTests.java
JRE System Library [JavaSE-1.8]
Maven Dependencies
JUnit 4
img
src
target
Dockerfile
Dockerfile-Jenkins-Maven
Jenkinsfile
mvnw
mvnw.cmd
pom.xml
OnlineBankingApplicationTests.java

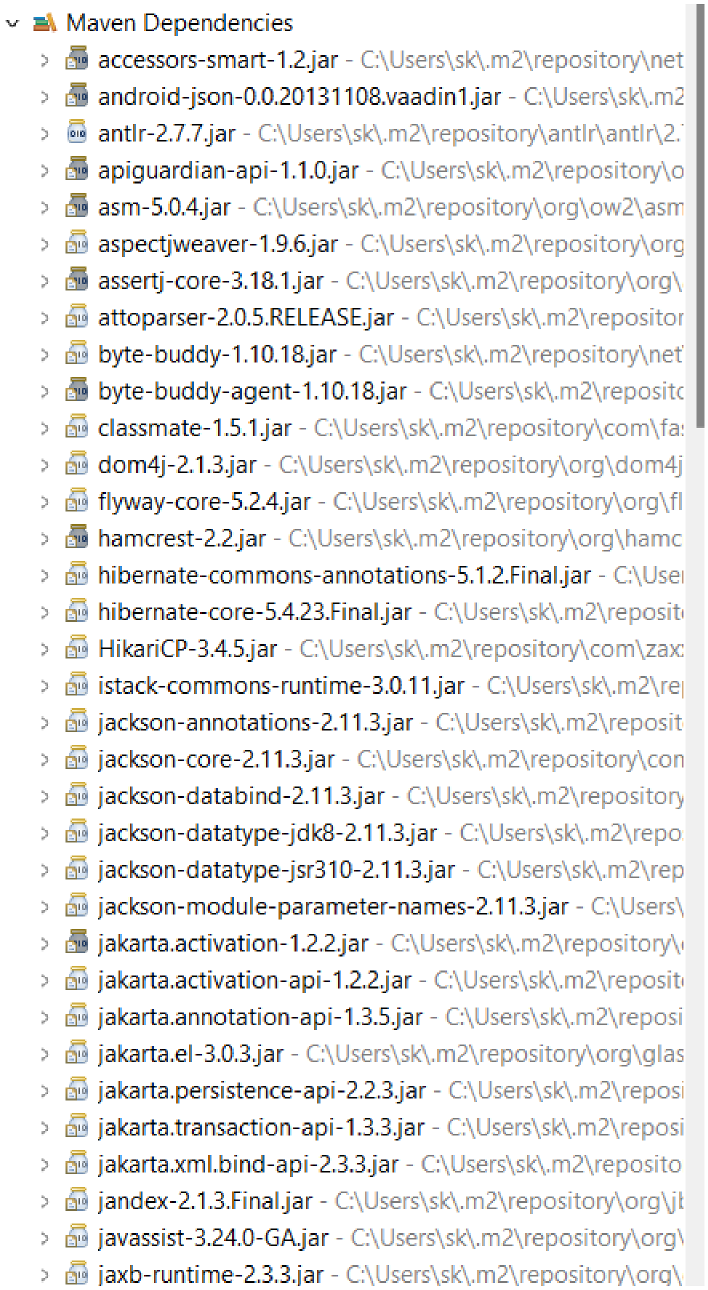
```java
package com.hendisantika.onlinebanking;

import org.junit.Test;

@RunWith(SpringRunner.class)
@SpringBootTest
public class OnlineBankingApplicationTests {

    @Test
    public void contextLoads() {
    }

}
```


workspace-spring-tool-suite-4-4.13.1.RELEASE - online-banking/s
File
Edit
Source
Refactor
Navigate
Search
Project
Run
Package Explorer

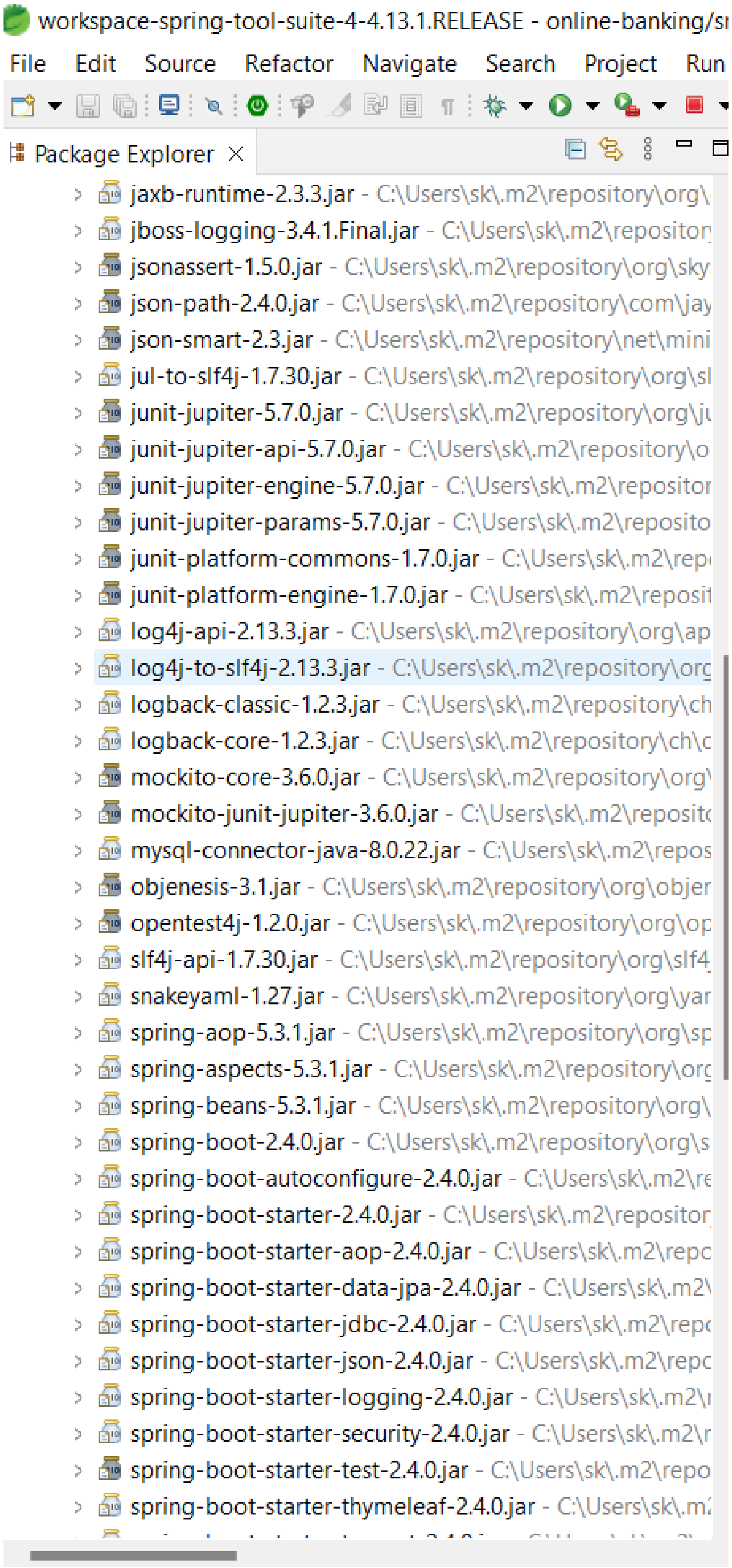
```
src/test/java
  com.hendisantika.onlinebanking
    OnlineBankingApplicationTests.java
JRE System Library [JavaSE-1.8]
  java.base - D:\MyPersonal\sk file\documents\download
  java.compiler - D:\MyPersonal\sk file\documents\downl
  java.datatransfer - D:\MyPersonal\sk file\documents\do
  java.desktop - D:\MyPersonal\sk file\documents\downl
  java.instrument - D:\MyPersonal\sk file\documents\dow
  java.logging - D:\MyPersonal\sk file\documents\downl
  java.management - D:\MyPersonal\sk file\documents\d
  java.management.rmi - D:\MyPersonal\sk file\documen
  java.naming - D:\MyPersonal\sk file\documents\downl
  java.net.http - D:\MyPersonal\sk file\documents\downl
  java.prefs - D:\MyPersonal\sk file\documents\download
  java.rmi - D:\MyPersonal\sk file\documents\downloads\
  java.scripting - D:\MyPersonal\sk file\documents\downl
  java.se - D:\MyPersonal\sk file\documents\downloads\
  java.security.jgss - D:\MyPersonal\sk file\documents\do
  java.security.sasl - D:\MyPersonal\sk file\documents\do
  java.sql - D:\MyPersonal\sk file\documents\downloads\
  java.sql.rowset - D:\MyPersonal\sk file\documents\dow
  java.transaction.xa - D:\MyPersonal\sk file\documents\d
  java.xml - D:\MyPersonal\sk file\documents\downloads
  java.xml.crypto - D:\MyPersonal\sk file\documents\dow
  jdk.accessibility - D:\MyPersonal\sk file\documents\dow
  jdk.attach - D:\MyPersonal\sk file\documents\downloa
  jdk.compiler - D:\MyPersonal\sk file\documents\downl
  jdk.dynalink - D:\MyPersonal\sk file\documents\downlo
  jdk.httpserver - D:\MyPersonal\sk file\documents\downl
  jdk.internal.ed - D:\MyPersonal\sk file\documents\dowr
  jdk.internal.jvmstat - D:\MyPersonal\sk file\documents\
  jdk.internal.le - D:\MyPersonal\sk file\documents\downl
  jdk.internal.opt - D:\MyPersonal\sk file\documents\dow
  jdk.jartool - D:\MyPersonal\sk file\documents\downloa
  jdk.javadoc - D:\MyPersonal\sk file\documents\downlo
  jdk.jconsole - D:\MyPersonal\sk file\documents\downlo
```



```
  > java.sql.rowset - D:\MyPersonal\sk file\documents\dow
  > java.transaction.xa - D:\MyPersonal\sk file\documents\d
  > java.xml - D:\MyPersonal\sk file\documents\downloads\
  > java.xml.crypto - D:\MyPersonal\sk file\documents\dow
  > jdk.accessibility - D:\MyPersonal\sk file\documents\dow
  > jdk.attach - D:\MyPersonal\sk file\documents\downloac
  > jdk.compiler - D:\MyPersonal\sk file\documents\downlc
  > jdk.dynalink - D:\MyPersonal\sk file\documents\downlo
  > jdk.httpserver - D:\MyPersonal\sk file\documents\downl
  > jdk.internal.ed - D:\MyPersonal\sk file\documents\dowr
  > jdk.internal.jvmstat - D:\MyPersonal\sk file\documents\c
  > jdk.internal.le - D:\MyPersonal\sk file\documents\downl
  > jdk.internal.opt - D:\MyPersonal\sk file\documents\dow
  > jdk.jartool - D:\MyPersonal\sk file\documents\downloac
  > jdk.javadoc - D:\MyPersonal\sk file\documents\downloa
  > jdk.jconsole - D:\MyPersonal\sk file\documents\downlo
  > jdk.jdi - D:\MyPersonal\sk file\documents\downloads\c
  > jdk.jdwp.agent - D:\MyPersonal\sk file\documents\dowr
  > jdk.jfr - D:\MyPersonal\sk file\documents\downloads\cc
  > jdk.jshell - D:\MyPersonal\sk file\documents\downloads
  > jdk.jsobject - D:\MyPersonal\sk file\documents\downloa
  > jdk.management - D:\MyPersonal\sk file\documents\do
  > jdk.management.agent - D:\MyPersonal\sk file\docume
  > jdk.management.jfr - D:\MyPersonal\sk file\documents\
  > jdk.net - D:\MyPersonal\sk file\documents\downloads\c
  > jdk.nio.mapmode - D:\MyPersonal\sk file\documents\dc
  > jdk.sctp - D:\MyPersonal\sk file\documents\downloads\
  > jdk.security.auth - D:\MyPersonal\sk file\documents\dov
  > jdk.security.jgss - D:\MyPersonal\sk file\documents\dow
  > jdk.unsupported - D:\MyPersonal\sk file\documents\dov
  > jdk.unsupported.desktop - D:\MyPersonal\sk file\docum
  > jdk.xml.dom - D:\MyPersonal\sk file\documents\downlc
> Maven Dependencies
> JUnit 4
```

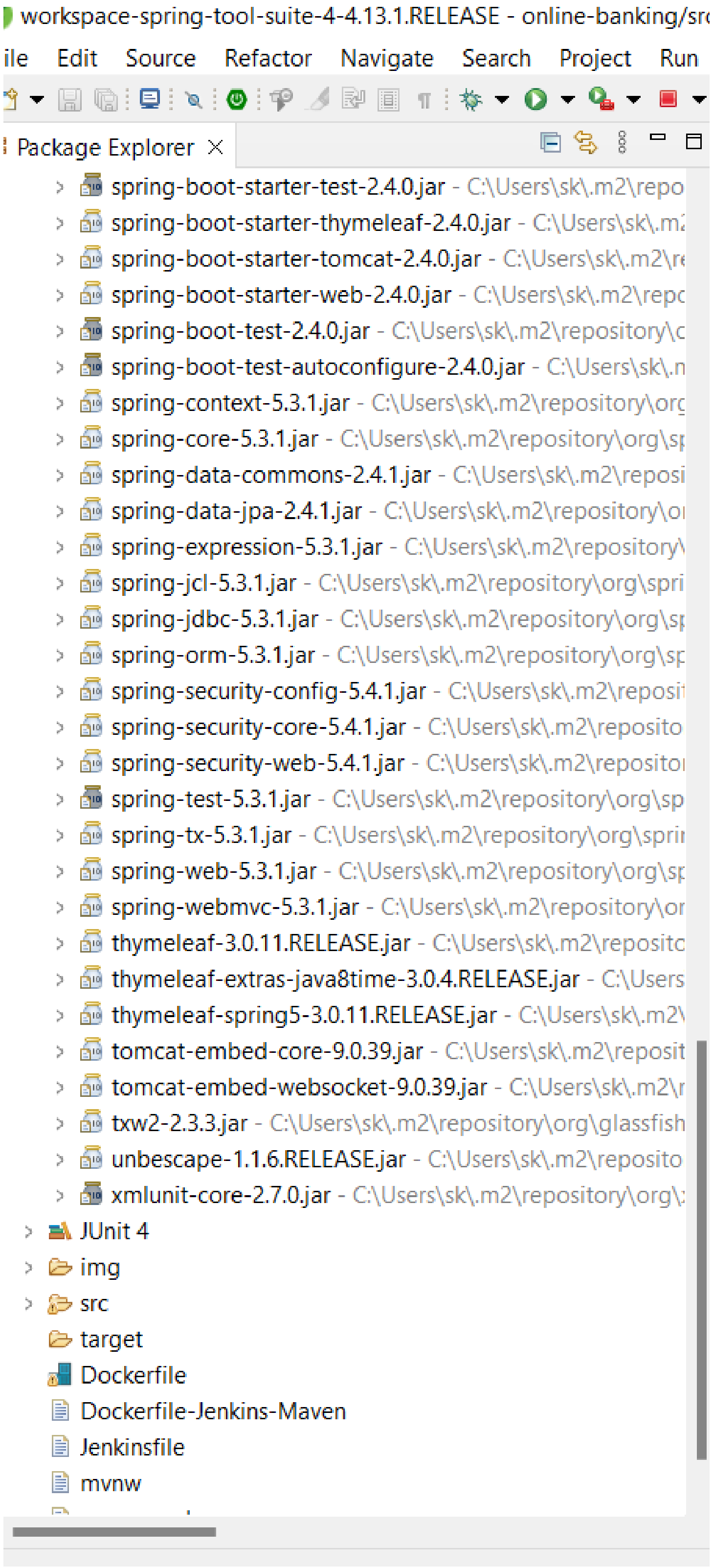


```
Maven Dependencies
  accessors-smart-1.2.jar - C:\Users\sk\.m2\repository\net
  android-json-0.0.20131108.vaadin1.jar - C:\Users\sk\.m2
  antlr-2.7.7.jar - C:\Users\sk\.m2\repository\antlr\antlr\2.
  apiguardian-api-1.1.0.jar - C:\Users\sk\.m2\repository\o
  asm-5.0.4.jar - C:\Users\sk\.m2\repository\org\ow2\asm
  aspectjweaver-1.9.6.jar - C:\Users\sk\.m2\repository\org
  assertj-core-3.18.1.jar - C:\Users\sk\.m2\repository\org\
  attoparser-2.0.5.RELEASE.jar - C:\Users\sk\.m2\repositor
  byte-buddy-1.10.18.jar - C:\Users\sk\.m2\repository\net
  byte-buddy-agent-1.10.18.jar - C:\Users\sk\.m2\reposito
  classmate-1.5.1.jar - C:\Users\sk\.m2\repository\com\fa
  dom4j-2.1.3.jar - C:\Users\sk\.m2\repository\org\dom4j
  flyway-core-5.2.4.jar - C:\Users\sk\.m2\repository\org\fl
  hamcrest-2.2.jar - C:\Users\sk\.m2\repository\org\hamc
  hibernate-commons-annotations-5.1.2.Final.jar - C:\User
  hibernate-core-5.4.23.Final.jar - C:\Users\sk\.m2\reposit
  HikariCP-3.4.5.jar - C:\Users\sk\.m2\repository\com\zax
  istack-commons-runtime-3.0.11.jar - C:\Users\sk\.m2\re
  jackson-annotations-2.11.3.jar - C:\Users\sk\.m2\reposit
  jackson-core-2.11.3.jar - C:\Users\sk\.m2\repository\con
  jackson-databind-2.11.3.jar - C:\Users\sk\.m2\repository
  jackson-datatype-jdk8-2.11.3.jar - C:\Users\sk\.m2\repo
  jackson-datatype-jsr310-2.11.3.jar - C:\Users\sk\.m2\rep
  jackson-module-parameter-names-2.11.3.jar - C:\Users\
  jakarta.activation-1.2.2.jar - C:\Users\sk\.m2\repository\
  jakarta.activation-api-1.2.2.jar - C:\Users\sk\.m2\reposit
  jakarta.annotation-api-1.3.5.jar - C:\Users\sk\.m2\reposi
  jakarta.el-3.0.3.jar - C:\Users\sk\.m2\repository\org\glas
  jakarta.persistence-api-2.2.3.jar - C:\Users\sk\.m2\reposi
  jakarta.transaction-api-1.3.3.jar - C:\Users\sk\.m2\reposi
  jakarta.xml.bind-api-2.3.3.jar - C:\Users\sk\.m2\reposito
  jandex-2.1.3.Final.jar - C:\Users\sk\.m2\repository\org\j
  javassist-3.24.0-GA.jar - C:\Users\sk\.m2\repository\org\
  jaxb-runtime-2.3.3.jar - C:\Users\sk\.m2\repository\org\
```



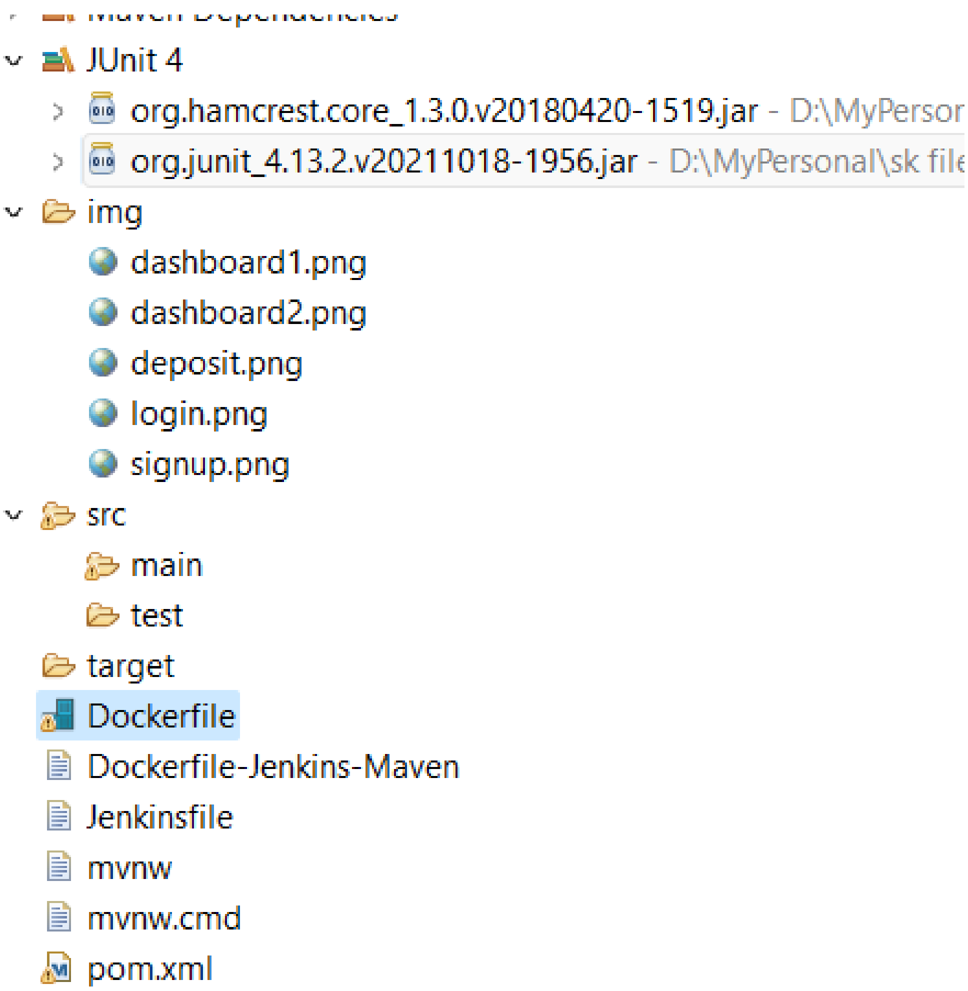
workspace-spring-tool-suite-4-4.13.1.RELEASE - online-banking/s
File
Edit
Source
Refactor
Navigate
Search
Project
Run
Package Explorer

```
jaxb-runtime-2.3.3.jar - C:\Users\sk\.m2\repository\org\
jboss-logging-3.4.1.Final.jar - C:\Users\sk\.m2\repositor
jsonassert-1.5.0.jar - C:\Users\sk\.m2\repository\org\sky
json-path-2.4.0.jar - C:\Users\sk\.m2\repository\com\jay
json-smart-2.3.jar - C:\Users\sk\.m2\repository\net\mini
jul-to-slf4j-1.7.30.jar - C:\Users\sk\.m2\repository\org\sl
junit-jupiter-5.7.0.jar - C:\Users\sk\.m2\repository\org\ju
junit-jupiter-api-5.7.0.jar - C:\Users\sk\.m2\repository\o
junit-jupiter-engine-5.7.0.jar - C:\Users\sk\.m2\repositor
junit-jupiter-params-5.7.0.jar - C:\Users\sk\.m2\reposito
junit-platform-commons-1.7.0.jar - C:\Users\sk\.m2\rep
junit-platform-engine-1.7.0.jar - C:\Users\sk\.m2\reposit
log4j-api-2.13.3.jar - C:\Users\sk\.m2\repository\org\ap
log4j-to-slf4j-2.13.3.jar - C:\Users\sk\.m2\repository\org
logback-classic-1.2.3.jar - C:\Users\sk\.m2\repository\ch
logback-core-1.2.3.jar - C:\Users\sk\.m2\repository\ch\c
mockito-core-3.6.0.jar - C:\Users\sk\.m2\repository\org\
mockito-junit-jupiter-3.6.0.jar - C:\Users\sk\.m2\reposito
mysql-connector-java-8.0.22.jar - C:\Users\sk\.m2\repos
objenesis-3.1.jar - C:\Users\sk\.m2\repository\org\objer
opentest4j-1.2.0.jar - C:\Users\sk\.m2\repository\org\op
slf4j-api-1.7.30.jar - C:\Users\sk\.m2\repository\org\slf4
snakeyaml-1.27.jar - C:\Users\sk\.m2\repository\org\yar
spring-aop-5.3.1.jar - C:\Users\sk\.m2\repository\org\sp
spring-aspects-5.3.1.jar - C:\Users\sk\.m2\repository\org
spring-beans-5.3.1.jar - C:\Users\sk\.m2\repository\org\
spring-boot-2.4.0.jar - C:\Users\sk\.m2\repository\org\s
spring-boot-autoconfigure-2.4.0.jar - C:\Users\sk\.m2\re
spring-boot-starter-2.4.0.jar - C:\Users\sk\.m2\repositor
spring-boot-starter-aop-2.4.0.jar - C:\Users\sk\.m2\repo
spring-boot-starter-data-jpa-2.4.0.jar - C:\Users\sk\.m2\
spring-boot-starter-jdbc-2.4.0.jar - C:\Users\sk\.m2\repo
spring-boot-starter-json-2.4.0.jar - C:\Users\sk\.m2\repo
spring-boot-starter-logging-2.4.0.jar - C:\Users\sk\.m2\r
spring-boot-starter-security-2.4.0.jar - C:\Users\sk\.m2\r
spring-boot-starter-test-2.4.0.jar - C:\Users\sk\.m2\repo
spring-boot-starter-thymeleaf-2.4.0.jar - C:\Users\sk\.m2
```



workspace-spring-tool-suite-4-4.13.1.RELEASE - online-banking/sr

```
    spring-boot-starter-test-2.4.0.jar - C:\Users\sk\.m2\repo
    spring-boot-starter-thymeleaf-2.4.0.jar - C:\Users\sk\.m2
    spring-boot-starter-tomcat-2.4.0.jar - C:\Users\sk\.m2\re
    spring-boot-starter-web-2.4.0.jar - C:\Users\sk\.m2\repo
    spring-boot-test-2.4.0.jar - C:\Users\sk\.m2\repository\c
    spring-boot-test-autoconfigure-2.4.0.jar - C:\Users\sk\.n
    spring-context-5.3.1.jar - C:\Users\sk\.m2\repository\org
    spring-core-5.3.1.jar - C:\Users\sk\.m2\repository\org\sp
    spring-data-commons-2.4.1.jar - C:\Users\sk\.m2\reposi
    spring-data-jpa-2.4.1.jar - C:\Users\sk\.m2\repository\or
    spring-expression-5.3.1.jar - C:\Users\sk\.m2\repository\
    spring-jcl-5.3.1.jar - C:\Users\sk\.m2\repository\org\spri
    spring-jdbc-5.3.1.jar - C:\Users\sk\.m2\repository\org\sp
    spring-orm-5.3.1.jar - C:\Users\sk\.m2\repository\org\sp
    spring-security-config-5.4.1.jar - C:\Users\sk\.m2\reposi
    spring-security-core-5.4.1.jar - C:\Users\sk\.m2\reposito
    spring-security-web-5.4.1.jar - C:\Users\sk\.m2\reposito
    spring-test-5.3.1.jar - C:\Users\sk\.m2\repository\org\sp
    spring-tx-5.3.1.jar - C:\Users\sk\.m2\repository\org\spri
    spring-web-5.3.1.jar - C:\Users\sk\.m2\repository\org\sp
    spring-webmvc-5.3.1.jar - C:\Users\sk\.m2\repository\or
    thymeleaf-3.0.11.RELEASE.jar - C:\Users\sk\.m2\reposito
    thymeleaf-extras-java8time-3.0.4.RELEASE.jar - C:\Users
    thymeleaf-spring5-3.0.11.RELEASE.jar - C:\Users\sk\.m2\
    tomcat-embed-core-9.0.39.jar - C:\Users\sk\.m2\reposit
    tomcat-embed-websocket-9.0.39.jar - C:\Users\sk\.m2\r
    txw2-2.3.3.jar - C:\Users\sk\.m2\repository\org\glassfish
    unbescape-1.1.6.RELEASE.jar - C:\Users\sk\.m2\reposito
    xmlunit-core-2.7.0.jar - C:\Users\sk\.m2\repository\org\
  JUnit 4
  img
  src
  target
  Dockerfile
  Dockerfile-Jenkins-Maven
  Jenkinsfile
  mvnw
```



```
>  Maven Dependencies
v  JUnit 4
     >  org.hamcrest.core_1.3.0.v20180420-1519.jar - D:\MyPersor
     >  org.junit_4.13.2.v20211018-1956.jar - D:\MyPersonal\sk file
v  img
       dashboard1.png
       dashboard2.png
       deposit.png
       login.png
       signup.png
v  src
       main
       test
   target
   Dockerfile
   Dockerfile-Jenkins-Maven
   Jenkinsfile
   mvnw
   mvnw.cmd
   pom.xml
```

5. Project Code:

1. Pom.Xml

```
<?xmlversion="1.0"encoding="UTF-8"?>
<projectxmlns="http://maven.apache.org/POM/4.0.0"xmlns:xsi="http://www.w3.org/2001/X
MLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<groupId>com.hendisantika</groupId>
<artifactId>online-banking</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>jar</packaging>

<name>online-banking</name>
<description>Demo project for Spring Boot</description>

<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>2.4.0</version>
</parent>
<properties>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
<project.reporting.outputEncoding>UTF-
8</project.reporting.outputEncoding>
<java.version>1.8</java.version>
</properties>

<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>

<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>

<dependency>
<groupId>org.springframework.boot</groupId>
```

```
    <artifactId>spring-boot-starter-jdbc</artifactId>
</dependency>

<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>

<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
</dependency>

<dependency
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-security</artifactId>
</dependency>
<!-- https://mvnrepository.com/artifact/org.flywaydb/flyway-core -->
<dependency>
<groupId>org.flywaydb</groupId>
<artifactId>flyway-core</artifactId>
<version>5.2.4</version>
</dependency>

<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-maven-plugin</artifactId></plugin>
</plugins>
    </build>

</project>
```

2. OnlineBankingApplication:

```
package com.bank.onlinebanking;



import org.springframework.boot.SpringApplication;
```

```java
import org.springframework.boot.autoconfigure.SpringBootApplication;


@SpringBootApplication
public class OnlineBankingApplication {


  public static void main(String[] args) {
    SpringApplication.run(OnlineBankingApplication.class, args);
  }
```

3. RequestFilter.java:

```java
package com.bank.onlinebanking.config;


import org.springframework.core.Ordered;
import org.springframework.core.annotation.Order;
import org.springframework.stereotype.Component;


import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


/**
 * Created by IntelliJ IDEA.
 * Project : online-banking




 * Time: 06.27
 * To change this template use File | Settings | File Templates.
 */
@Component
@Order(Ordered.HIGHEST_PRECEDENCE)
public class RequestFilter implements Filter {

    public void doFilter(ServletRequest req, ServletResponse res, FilterChain chain) {
        HttpServletResponse response = (HttpServletResponse) res;
        HttpServletRequest request = (HttpServletRequest) req;

        response.setHeader("Access-Control-Allow-Origin", "http://localhost:4200");
        response.setHeader("Access-Control-Allow-Methods", "POST, PUT, GET, OPTIONS, DELETE");
        response.setHeader("Access-Control-Allow-Headers", "x-requested-with");
        response.setHeader("Access-Control-Max-Age", "3600");
        response.setHeader("Access-Control-Allow-Credentials", "true");

        if (!(request.getMethod().equalsIgnoreCase("OPTIONS"))) {
            try {
                chain.doFilter(req, res);
            } catch (Exception e) {
                e.printStackTrace();
            }
        } else {
            System.out.println("Pre-flight");
            response.setHeader("Access-Control-Allow-Methods", "POST,GET,DELETE");
            response.setHeader("Access-Control-Max-Age", "3600");
            response.setHeader("Access-Control-Allow-Headers", "authorization, content-type," +
                    "access-control-request-headers,access-control-request-
method,accept,origin,authorization,x-requested-with");
            response.setStatus(HttpServletResponse.SC_OK);
        }

    }

    public void init(FilterConfig filterConfig) {
    }

    public void destroy() {
    }

}
```

4. SecurityConfig.java:

```
package com.bank.onlinebanking.config;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
```

```
import
org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder
;
import
org.springframework.security.config.annotation.method.configuration.EnableGlobalMethodSecurity;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import
org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.web.util.matcher.AntPathRequestMatcher;


import com.bank.onlinebanking.service.UserServiceImpl.UserSecurityService;


import java.security.SecureRandom;


/**
 * Created by IntelliJ IDEA.
 * Project : online-banking




 * Time: 06.29
 * To change this template use File | Settings | File Templates.
 */
@Configuration
@EnableWebSecurity
@EnableGlobalMethodSecurity(prePostEnabled = true)
public class SecurityConfig extends WebSecurityConfigurerAdapter {
```

```
//    @Autowired
//    private Environment env;

    private static final String SALT = "salt"; // Salt should be protected carefully
    private static final String[] PUBLIC_MATCHERS = {
            "/webjars/**",
            "/css/**",
            "/js/**",
            "/images/**",
            "/",
            "/about/**",
            "/contact/**",
            "/error/**/*",
            "/console/**",
            "/signup"
    };
    @Autowired
    private UserSecurityService userSecurityService;

    @Bean
    public BCryptPasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder(12, new SecureRandom(SALT.getBytes()));
    }

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http
```

```
                .authorizeRequests().
//                antMatchers("/**").
        antMatchers(PUBLIC_MATCHERS).
                permitAll().anyRequest().authenticated();


        http
                .csrf().disable().cors().disable()

.formLogin().failureUrl("/index?error").defaultSuccessUrl("/userFront").loginPage("/index").permitAll()
                .and()
                .logout().logoutRequestMatcher(new AntPathRequestMatcher("/logout")).logoutSuccessUrl("/index?logout").deleteCookies("remember-me").permitAll()
                .and()
                .rememberMe();
    }



    @Autowired
    public void configureGlobal(AuthenticationManagerBuilder auth) throws Exception {
//            auth.inMemoryAuthentication().withUser("user").password("password").roles("USER"); //This is in-memory authentication
        auth.userDetailsService(userSecurityService).passwordEncoder(passwordEncoder());
    }



}
```

5. AccountController.java:

```
package com.bank.onlinebanking.controller;


import com.bank.onlinebanking.entity.PrimaryAccount;
import com.bank.onlinebanking.entity.PrimaryTransaction;
import com.bank.onlinebanking.entity.SavingsAccount;
import com.bank.onlinebanking.entity.SavingsTransaction;
import com.bank.onlinebanking.entity.User;
import com.bank.onlinebanking.service.AccountService;
import com.bank.onlinebanking.service.TransactionService;
import com.bank.onlinebanking.service.UserService;


import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;


import java.security.Principal;
import java.util.List;


/**
 * Created by IntelliJ IDEA.
 * Project : online-banking
 * To change this template use File | Settings | File Templates.
 */
@Controller
@RequestMapping("/account")
public class AccountController {

    @Autowired
    private UserService userService;

    @Autowired
    private AccountService accountService;

    @Autowired
    private TransactionService transactionService;

    @RequestMapping("/primaryAccount")
    public String primaryAccount(Model model, Principal principal) {
        List<PrimaryTransaction> primaryTransactionList =
transactionService.findPrimaryTransactionList(principal.getName());

        User user = userService.findByUsername(principal.getName());
        PrimaryAccount primaryAccount = user.getPrimaryAccount();

        model.addAttribute("primaryAccount", primaryAccount);
        model.addAttribute("primaryTransactionList", primaryTransactionList);

        return "primaryAccount";
    }

    @RequestMapping("/savingsAccount")
    public String savingsAccount(Model model, Principal principal) {
        List<SavingsTransaction> savingsTransactionList =
transactionService.findSavingsTransactionList(principal.getName());
        User user = userService.findByUsername(principal.getName());
        SavingsAccount savingsAccount = user.getSavingsAccount();

        model.addAttribute("savingsAccount", savingsAccount);
        model.addAttribute("savingsTransactionList", savingsTransactionList);

        return "savingsAccount";
    }

    @RequestMapping(value = "/deposit", method = RequestMethod.GET)
    public String deposit(Model model) {
        model.addAttribute("accountType", "");
        model.addAttribute("amount", "");

        return "deposit";
    }

    @RequestMapping(value = "/deposit", method = RequestMethod.POST)
    public String depositPOST(@ModelAttribute("amount") String amount,
@ModelAttribute("accountType") String accountType, Principal principal) {
        accountService.deposit(accountType, Double.parseDouble(amount), principal);

        return "redirect:/userFront";
    }


    @RequestMapping(value = "/withdraw", method = RequestMethod.GET)
    public String withdraw(Model model) {
        model.addAttribute("accountType", "");
        model.addAttribute("amount", "");

        return "withdraw";
    }


    @RequestMapping(value = "/withdraw", method = RequestMethod.POST)
    public String withdrawPOST(@ModelAttribute("amount") String amount,
@ModelAttribute("accountType") String accountType, Principal principal) {
        accountService.withdraw(accountType, Double.parseDouble(amount), principal);

        return "redirect:/userFront";
    }
}
```

6. AppointmentController.java:

```
package com.bank.onlinebanking.controller;

import com.bank.onlinebanking.entity.Appointment;
import com.bank.onlinebanking.entity.User;
import com.bank.onlinebanking.service.AppointmentService;
import com.bank.onlinebanking.service.UserService;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import java.security.Principal;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

/**
 * Created by IntelliJ IDEA.
 * Project : online-banking




 * Time: 06.35
 * To change this template use File | Settings | File Templates.
 */
@Controller
@RequestMapping("/appointment")
public class AppointmentController {

    @Autowired
    private AppointmentService appointmentService;

    @Autowired
    private UserService userService;

    @RequestMapping(value = "/create", method = RequestMethod.GET)
    public String createAppointment(Model model) {
        Appointment appointment = new Appointment();
        model.addAttribute("appointment", appointment);
        model.addAttribute("dateString", "");

        return "appointment";
    }

    @RequestMapping(value = "/create", method = RequestMethod.POST)
    public String createAppointmentPost(@ModelAttribute("appointment") Appointment appointment,
@ModelAttribute("dateString") String date, Model model, Principal principal) throws ParseException {

        SimpleDateFormat format1 = new SimpleDateFormat("yyyy-MM-dd hh:mm");
        Date d1 = format1.parse(date);
        appointment.setDate(d1);

        User user = userService.findByUsername(principal.getName());
        appointment.setUser(user);
        appointmentService.createAppointment(appointment);

        return "redirect:/userFront";
    }


}
```

7. HomeController.java:

```
package com.bank.onlinebanking.controller;

import com.bank.onlinebanking.entity.PrimaryAccount;
import com.bank.onlinebanking.entity.SavingsAccount;
import com.bank.onlinebanking.entity.User;
import com.bank.onlinebanking.repository.RoleDao;
import com.bank.onlinebanking.security.UserRole;
import com.bank.onlinebanking.service.UserService;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;

import java.security.Principal;
import java.util.HashSet;
import java.util.Set;

/**
 * Created by IntelliJ IDEA.
 * Project : online-banking




 * Time: 06.34
 * To change this template use File | Settings | File Templates.
 */
@Controller
public class HomeController {
```

```
@Autowired
private UserService userService;

@Autowired
private RoleDao roleDao;

@RequestMapping("/")
public String home() {
    return "redirect:/index";
}

@RequestMapping("/index")
public String index() {
    return "index";
}

@GetMapping("/signup")
public String signup(Model model) {
    User user = new User();

    model.addAttribute("user", user);

    return "signup";
}

@PostMapping("/signup")
public String signupPost(@ModelAttribute("user") User user, Model model) {

    if (userService.checkUserExists(user.getUsername(), user.getEmail())) {

        if (userService.checkEmailExists(user.getEmail())) {
            model.addAttribute("emailExists", true);
        }

        if (userService.checkUsernameExists(user.getUsername())) {
            model.addAttribute("usernameExists", true);
        }

        return "signup";
    } else {
        Set<UserRole> userRoles = new HashSet<>();
        userRoles.add(new UserRole(user, roleDao.findByName("ROLE_USER")));

        userService.createUser(user, userRoles);

        return "redirect:/";
    }
    }

    @GetMapping("/userFront")
    public String userFront(Principal principal, Model model) {
        User user = userService.findByUsername(principal.getName());
        PrimaryAccount primaryAccount = user.getPrimaryAccount();
        SavingsAccount savingsAccount = user.getSavingsAccount();

        model.addAttribute("primaryAccount", primaryAccount);
        model.addAttribute("savingsAccount", savingsAccount);

        return "userFront";
    }
}
```

8. TransferController.java

```
package com.bank.onlinebanking.controller;

import com.bank.onlinebanking.entity.PrimaryAccount;
import com.bank.onlinebanking.entity.Recipient;
import com.bank.onlinebanking.entity.SavingsAccount;
import com.bank.onlinebanking.entity.User;
import com.bank.onlinebanking.service.TransactionService;
import com.bank.onlinebanking.service.UserService;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.transaction.annotation.Transactional;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;

import java.security.Principal;
import java.util.List;

/**
 * Created by IntelliJ IDEA.
 * Project : online-banking
 * Time: 06.37
 * To change this template use File | Settings | File Templates.
 */
@Controller
@RequestMapping("/transfer")
public class TransferController {

  @Autowired
  private TransactionService transactionService;

  @Autowired
  private UserService userService;

  @RequestMapping(value = "/betweenAccounts", method = RequestMethod.GET)
  public String betweenAccounts(Model model) {
    model.addAttribute("transferFrom", "");
    model.addAttribute("transferTo", "");
    model.addAttribute("amount", "");

    return "betweenAccounts";
  }

  @RequestMapping(value = "/betweenAccounts", method = RequestMethod.POST)
  public String betweenAccountsPost(
      @ModelAttribute("transferFrom") String transferFrom,
      @ModelAttribute("transferTo") String transferTo,
      @ModelAttribute("amount") String amount,
      Principal principal
  ) throws Exception {
    User user = userService.findByUsername(principal.getName());
    PrimaryAccount primaryAccount = user.getPrimaryAccount();
    SavingsAccount savingsAccount = user.getSavingsAccount();
    transactionService.betweenAccountsTransfer(transferFrom, transferTo, amount, primaryAccount,
savingsAccount);

    return "redirect:/userFront";
  }

  @RequestMapping(value = "/recipient", method = RequestMethod.GET)
  public String recipient(Model model, Principal principal) {
    List<Recipient> recipientList = transactionService.findRecipientList(principal);

    Recipient recipient = new Recipient();

    model.addAttribute("recipientList", recipientList);
    model.addAttribute("recipient", recipient);
```

```
        return "recipient";
    }

    @RequestMapping(value = "/recipient/save", method = RequestMethod.POST)
    public String recipientPost(@ModelAttribute("recipient") Recipient recipient, Principal principal) {

        User user = userService.findByUsername(principal.getName());
        recipient.setUser(user);
        transactionService.saveRecipient(recipient);

        return "redirect:/transfer/recipient";
    }

    @RequestMapping(value = "/recipient/edit", method = RequestMethod.GET)
    public String recipientEdit(@RequestParam(value = "recipientName") String recipientName, Model
model, Principal principal) {

        Recipient recipient = transactionService.findRecipientByName(recipientName);
        List<Recipient> recipientList = transactionService.findRecipientList(principal);

        model.addAttribute("recipientList", recipientList);
        model.addAttribute("recipient", recipient);

        return "recipient";
    }

    @RequestMapping(value = "/recipient/delete", method = RequestMethod.GET)
    @Transactional
    public String recipientDelete(@RequestParam(value = "recipientName") String recipientName, Model
model, Principal principal) {

        transactionService.deleteRecipientByName(recipientName);

        List<Recipient> recipientList = transactionService.findRecipientList(principal);

        Recipient recipient = new Recipient();
        model.addAttribute("recipient", recipient);
        model.addAttribute("recipientList", recipientList);


        return "recipient";
    }

    @RequestMapping(value = "/toSomeoneElse", method = RequestMethod.GET)
    public String toSomeoneElse(Model model, Principal principal) {
        List<Recipient> recipientList = transactionService.findRecipientList(principal);
        model.addAttribute("recipientList", recipientList);
        model.addAttribute("accountType", "");

        return "toSomeoneElse";
    }

    @RequestMapping(value = "/toSomeoneElse", method = RequestMethod.POST)
    public String toSomeoneElsePost(@ModelAttribute("recipientName") String recipientName,
@ModelAttribute("accountType") String accountType, @ModelAttribute("amount") String amount,
Principal principal) {
        User user = userService.findByUsername(principal.getName());
        Recipient recipient = transactionService.findRecipientByName(recipientName);
        transactionService.toSomeoneElseTransfer(recipient, accountType, amount,
user.getPrimaryAccount(), user.getSavingsAccount());

        return "redirect:/userFront";
    }
}
```

9. UserController.java

```
package com.bank.onlinebanking.controller;

import com.bank.onlinebanking.entity.User;
import com.bank.onlinebanking.service.UserService;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;

import java.security.Principal;

/**
 * Created by IntelliJ IDEA.
 * Project : online-banking




 * Time: 06.38
 * To change this template use File | Settings | File Templates.
 */
@Controller
@RequestMapping("/user")
public class UserController {

    @Autowired
    private UserService userService;

    @GetMapping("/profile")
    public String profile(Principal principal, Model model) {
        User user = userService.findByUsername(principal.getName());

        model.addAttribute("user", user);

        return "profile";
    }

    @PostMapping("/profile")
    public String profilePost(@ModelAttribute("user") User newUser, Model model) {
        User user = userService.findByUsername(newUser.getUsername());
        user.setUsername(newUser.getUsername());
        user.setFirstName(newUser.getFirstName());
        user.setLastName(newUser.getLastName());
        user.setEmail(newUser.getEmail());
        user.setPhone(newUser.getPhone());

        model.addAttribute("user", user);

        userService.saveUser(user);

        return "profile";
    }


}
```

10.Appointment.java:

```
package com.bank.onlinebanking.entity;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import java.util.Date;
```

```
/**
 * Created by IntelliJ IDEA.
 * Project : online-banking




 * Time: 06.46
 * To change this template use File | Settings | File Templates.
 */
@Entity
public class Appointment {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private Date date;
    private String location;
    private String description;
    private boolean confirmed;

    @ManyToOne
    @JoinColumn(name = "user_id")
    private User user;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public Date getDate() {
        return date;
    }

    public void setDate(Date date) {
        this.date = date;
    }

    public String getLocation() {
        return location;
    }

    public void setLocation(String location) {
        this.location = location;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public User getUser() {
        return user;
    }

    public void setUser(User user) {
        this.user = user;
    }

    public boolean isConfirmed() {
        return confirmed;
    }

    public void setConfirmed(boolean confirmed) {
        this.confirmed = confirmed;
    }

    @Override
    public String toString() {
        return "Appointment{" +
                "id=" + id +
                ", date=" + date +
                ", location='" + location + '\'' +
                ", description='" + description + '\'' +
                ", user=" + user +
                '}';
    }
}
```

11.PrimaryAccount.java:

```
package com.bank.onlinebanking.entity;

import com.fasterxml.jackson.annotation.JsonIgnore;

import javax.persistence.CascadeType;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import java.math.BigDecimal;
import java.util.List;

/**
 * Created by IntelliJ IDEA.
 * Project : online-banking




 * Time: 06.49
 * To change this template use File | Settings | File Templates.
 */
@Entity
public class PrimaryAccount {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private int accountNumber;
    private BigDecimal accountBalance;

    @OneToMany(mappedBy = "primaryAccount", cascade = CascadeType.ALL, fetch = FetchType.LAZY)
    @JsonIgnore
    private List<PrimaryTransaction> primaryTransactionList;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public int getAccountNumber() {
        return accountNumber;
    }

    public void setAccountNumber(int accountNumber) {
        this.accountNumber = accountNumber;
    }
    public BigDecimal getAccountBalance() {
        return accountBalance;
    }

    public void setAccountBalance(BigDecimal accountBalance) {
        this.accountBalance = accountBalance;
    }

    public List<PrimaryTransaction> getPrimaryTransactionList() {
        return primaryTransactionList;
    }

    public void setPrimaryTransactionList(List<PrimaryTransaction> primaryTransactionList) {
        this.primaryTransactionList = primaryTransactionList;
    }


}
```

12.PrimaryTransaction.java:

```
package com.bank.onlinebanking.entity;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import java.math.BigDecimal;
import java.util.Date;

/**
 * Created by IntelliJ IDEA.
 * Project : online-banking




 * Time: 06.49
 * To change this template use File | Settings | File Templates.
 */
@Entity
public class PrimaryTransaction {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private Date date;
    private String description;
    private String type;
    private String status;
    private double amount;
    private BigDecimal availableBalance;
    @ManyToOne
    @JoinColumn(name = "primary_account_id")
    private PrimaryAccount primaryAccount;


    public PrimaryTransaction() {
    }

    public PrimaryTransaction(Date date, String description, String type, String status, double amount,
BigDecimal availableBalance, PrimaryAccount primaryAccount) {
        this.date = date;
        this.description = description;
        this.type = type;
        this.status = status;
        this.amount = amount;
        this.availableBalance = availableBalance;
        this.primaryAccount = primaryAccount;
    }

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public Date getDate() {
        return date;
    }

    public void setDate(Date date) {
        this.date = date;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public String getType() {
        return type;
    }

    public void setType(String type) {
        this.type = type;
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

    public double getAmount() {
        return amount;
    }

    public void setAmount(double amount) {
        this.amount = amount;
    }

    public BigDecimal getAvailableBalance() {
        return availableBalance;
    }

    public void setAvailableBalance(BigDecimal availableBalance) {
        this.availableBalance = availableBalance;
    }

    public PrimaryAccount getPrimaryAccount() {
        return primaryAccount;
    }

    public void setPrimaryAccount(PrimaryAccount primaryAccount) {
        this.primaryAccount = primaryAccount;
    }

}
```

13. Recipient.java:

```
package com.bank.onlinebanking.entity;

import com.fasterxml.jackson.annotation.JsonIgnore;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;

/**
 * Created by IntelliJ IDEA.
 * Project : online-banking




 * Time: 06.54
 * To change this template use File | Settings | File Templates.
 */
@Entity
public class Recipient {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;
    private String email;
    private String phone;
    private String accountNumber;
    private String description;

    @ManyToOne
    @JoinColumn(name = "user_id")
    @JsonIgnore
    private User user;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
```

```
    this.id = id;
  }

  public String getName() {
    return name;
  }

  public void setName(String name) {
    this.name = name;
  }

  public String getEmail() {
    return email;
  }

  public void setEmail(String email) {
    this.email = email;
  }

  public String getPhone() {
    return phone;
  }

  public void setPhone(String phone) {
    this.phone = phone;
  }

  public String getAccountNumber() {
    return accountNumber;
  }

  public void setAccountNumber(String accountNumber) {
    this.accountNumber = accountNumber;
  }

  public User getUser() {
    return user;
  }

  public void setUser(User user) {
    this.user = user;
  }

  public String getDescription() {
    return description;
  }
```

```
    public void setDescription(String description) {
        this.description = description;
    }
}
```

14.SavingsAccount.java:

```
package com.bank.onlinebanking.entity;

import com.fasterxml.jackson.annotation.JsonIgnore;

import javax.persistence.CascadeType;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import java.math.BigDecimal;
import java.util.List;

/**
 * Created by IntelliJ IDEA.
 * Project : online-banking




 * Time: 06.51
 * To change this template use File | Settings | File Templates.
 */
@Entity
public class SavingsAccount {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private int accountNumber;
    private BigDecimal accountBalance;

    @OneToMany(mappedBy = "savingsAccount", cascade = CascadeType.ALL, fetch = FetchType.LAZY)
    @JsonIgnore
    private List<SavingsTransaction> savingsTransactionList;
    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public int getAccountNumber() {
        return accountNumber;
    }

    public void setAccountNumber(int accountNumber) {
        this.accountNumber = accountNumber;
    }

    public BigDecimal getAccountBalance() {
        return accountBalance;
    }

    public void setAccountBalance(BigDecimal accountBalance) {
        this.accountBalance = accountBalance;
    }

    public List<SavingsTransaction> getSavingsTransactionList() {
        return savingsTransactionList;
    }

    public void setSavingsTransactionList(List<SavingsTransaction> savingsTransactionList) {
        this.savingsTransactionList = savingsTransactionList;
    }


}
```

15.SavingsTransaction.java:

```
package com.bank.onlinebanking.entity;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import java.math.BigDecimal;
import java.util.Date;

/**
 * Created by IntelliJ IDEA.
 * Project : online-banking




 * Time: 06.52
 * To change this template use File | Settings | File Templates.
 */
@Entity
public class SavingsTransaction {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private Date date;
    private String description;
    private String type;
    private String status;
    private double amount;
    private BigDecimal availableBalance;

    @ManyToOne
    @JoinColumn(name = "savings_account_id")
    private SavingsAccount savingsAccount;

    public SavingsTransaction() {
    }

    public SavingsTransaction(Date date, String description, String type, String status, double amount,
BigDecimal availableBalance, SavingsAccount savingsAccount) {
        this.date = date;
        this.description = description;
        this.type = type;
        this.status = status;
        this.amount = amount;
        this.availableBalance = availableBalance;
        this.savingsAccount = savingsAccount;
    }

    public Long getId() {
        return id;
    }
```

```
public void setId(Long id) {
   this.id = id;
}

public Date getDate() {
   return date;
}

public void setDate(Date date) {
   this.date = date;
}

public String getDescription() {
   return description;
}

public void setDescription(String description) {
   this.description = description;
}

public String getType() {
   return type;
}

public void setType(String type) {
   this.type = type;
}

public String getStatus() {
   return status;
}

public void setStatus(String status) {
   this.status = status;
}

public double getAmount() {
   return amount;
}

public void setAmount(double amount) {
   this.amount = amount;
}

public BigDecimal getAvailableBalance() {
   return availableBalance;
```

```java
    }

    public void setAvailableBalance(BigDecimal availableBalance) {
        this.availableBalance = availableBalance;
    }

    public SavingsAccount getSavingsAccount() {
        return savingsAccount;
    }

    public void setSavingsAccount(SavingsAccount savingsAccount) {
        this.savingsAccount = savingsAccount;
    }
}
```

16. User.java

```java
package com.bank.onlinebanking.entity;

import com.bank.onlinebanking.security.Authority;
import com.bank.onlinebanking.security.UserRole;
import com.fasterxml.jackson.annotation.JsonIgnore;

import org.springframework.security.core.GrantedAuthority;
import org.springframework.security.core.userdetails.UserDetails;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.OneToOne;
import java.util.Collection;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

/**
 * Created by IntelliJ IDEA.
 * Project : online-banking
 * Time: 06.47
 * To change this template use File | Settings | File Templates.
 */
@Entity
public class User implements UserDetails {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "userId", nullable = false, updatable = false)
    private Long userId;
    private String username;
    private String password;
    private String firstName;
    private String lastName;

    @Column(name = "email", nullable = false, unique = true)
    private String email;
    private String phone;

    private boolean enabled = true;

    @OneToOne
    private PrimaryAccount primaryAccount;

    @OneToOne
    private SavingsAccount savingsAccount;

    @OneToMany(mappedBy = "user", cascade = CascadeType.ALL, fetch = FetchType.LAZY)
    @JsonIgnore
    private List<Appointment> appointmentList;

    @OneToMany(mappedBy = "user", cascade = CascadeType.ALL, fetch = FetchType.LAZY)
    private List<Recipient> recipientList;

    @OneToMany(mappedBy = "user", cascade = CascadeType.ALL, fetch = FetchType.EAGER)
    @JsonIgnore
    private Set<UserRole> userRoles = new HashSet<>();

    public Set<UserRole> getUserRoles() {
        return userRoles;
    }

    public void setUserRoles(Set<UserRole> userRoles) {
        this.userRoles = userRoles;
    }

    public Long getUserId() {
```

```
    return userId;
}

public void setUserId(Long userId) {
    this.userId = userId;
}

public String getUsername() {
    return username;
}

public void setUsername(String username) {
    this.username = username;
}

public String getFirstName() {
    return firstName;
}

public void setFirstName(String firstName) {
    this.firstName = firstName;
}

public String getLastName() {
    return lastName;
}

public void setLastName(String lastName) {
    this.lastName = lastName;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}

public String getPhone() {
    return phone;
}

public void setPhone(String phone) {
    this.phone = phone;
}
```

```
    public List<Appointment> getAppointmentList() {
        return appointmentList;
    }

    public void setAppointmentList(List<Appointment> appointmentList) {
        this.appointmentList = appointmentList;
    }

    public List<Recipient> getRecipientList() {
        return recipientList;
    }

    public void setRecipientList(List<Recipient> recipientList) {
        this.recipientList = recipientList;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public PrimaryAccount getPrimaryAccount() {
        return primaryAccount;
    }

    public void setPrimaryAccount(PrimaryAccount primaryAccount) {
        this.primaryAccount = primaryAccount;
    }

    public SavingsAccount getSavingsAccount() {
        return savingsAccount;
    }

    public void setSavingsAccount(SavingsAccount savingsAccount) {
        this.savingsAccount = savingsAccount;
    }

    @Override
    public String toString() {
        return "User{" +
                "userId=" + userId +
                ", username='" + username + '\'' +
                ", password='" + password + '\'' +
                ", firstName='" + firstName + '\'' +
                ", lastName='" + lastName + '\'' +
                ", email='" + email + '\'' +
                ", phone='" + phone + '\'' +
                ", appointmentList=" + appointmentList +
                ", recipientList=" + recipientList +
                ", userRoles=" + userRoles +
                '}';
    }

    @Override
    public Collection<? extends GrantedAuthority> getAuthorities() {
        Set<GrantedAuthority> authorities = new HashSet<>();
        userRoles.forEach(ur -> authorities.add(new Authority(ur.getRole().getName())));
        return authorities;
    }

    @Override
    public boolean isAccountNonExpired() {
        // TODO Auto-generated method stub
        return true;
    }

    @Override
    public boolean isAccountNonLocked() {
        // TODO Auto-generated method stub
        return true;
    }

    @Override
    public boolean isCredentialsNonExpired() {
        // TODO Auto-generated method stub
        return true;
    }

    @Override
    public boolean isEnabled() {
        return enabled;
    }

    public void setEnabled(boolean enabled) {
        this.enabled = enabled;
    }


}
```

17.AppointmentResource.java:

```
package com.example.datamodel.repository;

import com.example.datamodel.entity.User;
import com.example.datamodel.entity.UserOrder;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;


import javax.persistence.EntityManager;
import javax.persistence.Query;
import javax.persistence.TypedQuery;
import javax.transaction.Transactional;
import java.util.List;
import java.util.Queue;

@Repository
@Transactional
public class UserRepo {

    @Autowired
    EntityManager em;


    private Logger log = LoggerFactory.getLogger(this.getClass());

    /*
     * find user by user id
     * using entity manager's find method
     * return type is user object*/
    public User findUserById(Long userId) {
        User user = em.find(User.class, userId);
        log.info("user object returned -> {}", user);
        return user;

    }

    public User userUpdate(Long userId, String oldPassword, String newPassword) {
        User user =  em.find(User.class, userId);
        if (user.getPassword() == oldPassword) {
            user.setPassword(newPassword);
            if (user.getPassword() == newPassword) {
                return user;
            } else {
                return null;
            }
        } else {
            return null;
        }
    }

    public User addNewUser(User user) {
        em.persist(user);
        return user;
    }

    public List<User> getUsers() {
        TypedQuery<User> query = em.createNamedQuery("get_all_users",User.class);
        List<User> result =  query.getResultList();
        return result;
    }

    public User verifyUser(String email, String password) {
        User user = null;

        List<User> users = getUsers();
        User newuser = null;
        for (int i = 0; i < users.size(); i++) {
            user = users.get(i);
            if (user.getEmail().equals(email)) {
                if (user.getPassword().equals(password)) {
                    newuser = user;
                    break;
                }

            }
        }
        log.info("user -> {}", user);
        return newuser;
    }

    public List<User> searchUserByName(String name) {
        Query query = em.createNativeQuery("Select * from user where name = :name",User.class);
        query.setParameter("name",name);
        List result = query.getResultList();
```

```
        return result;
    }

    public void addUserOrderToUser(Long id, UserOrder userOrder) {
        User user = em.find(User.class,id);
        user.setUserOrder(userOrder);
        em.merge(user);

    }

    public UserOrder verifyUserToOrder(Long userId) {
        User user = em.find(User.class,userId);
            return user.getUserOrder();

    }




}
```

18. UserResource.css:

```
package com.bank.onlinebanking.resource;

import com.bank.onlinebanking.entity.PrimaryTransaction;
import com.bank.onlinebanking.entity.SavingsTransaction;
import com.bank.onlinebanking.entity.User;
import com.bank.onlinebanking.service.TransactionService;
import com.bank.onlinebanking.service.UserService;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.access.prepost.PreAuthorize;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;
```

```
/**
 * Created by IntelliJ IDEA.
 * Project : online-banking




 * Time: 06.39
 * To change this template use File | Settings | File Templates.
 */
@RestController
@RequestMapping("/api")
@PreAuthorize("hasRole('ADMIN')")
public class UserResource {

   @Autowired
   private UserService userService;

   @Autowired
   private TransactionService transactionService;

   @RequestMapping(value = "/user/all", method = RequestMethod.GET)
   public List<User> userList() {
      return userService.findUserList();
   }

   @RequestMapping(value = "/user/primary/transaction", method = RequestMethod.GET)
   public List<PrimaryTransaction> getPrimaryTransactionList(@RequestParam("username") String
username) {
      return transactionService.findPrimaryTransactionList(username);
   }

   @RequestMapping(value = "/user/savings/transaction", method = RequestMethod.GET)
   public List<SavingsTransaction> getSavingsTransactionList(@RequestParam("username") String
username) {
      return transactionService.findSavingsTransactionList(username);
   }

   @RequestMapping("/user/{username}/enable")
   public void enableUser(@PathVariable("username") String username) {
      userService.enableUser(username);
   }

   @RequestMapping("/user/{username}/disable")
   public void diableUser(@PathVariable("username") String username) {
      userService.disableUser(username);
   }
}
```

19.role.java

```
package com.bank.onlinebanking.security;

import javax.persistence.CascadeType;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import java.util.HashSet;
import java.util.Set;

/**
 * Created by IntelliJ IDEA.
 * Project : online-banking



 * Date: 06/08/18
 * Time: 20.45
 * To change this template use File | Settings | File Templates.
 */
@Entity
public class Role {

    @Id
    private int roleId;

    private String name;

    @OneToMany(mappedBy = "role", cascade = CascadeType.ALL, fetch = FetchType.LAZY)
    private Set<UserRole> userRoles = new HashSet<>();

    public Role() {

    }

    public int getRoleId() {
        return roleId;
    }

    public void setRoleId(int roleId) {
        this.roleId = roleId;
    }
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public Set<UserRole> getUserRoles() {
        return userRoles;
    }

    public void setUserRoles(Set<UserRole> userRoles) {
        this.userRoles = userRoles;
    }


}
```

20.userrole.java

```
package com.bank.onlinebanking.security;


import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.Table;

import com.bank.onlinebanking.entity.User;

/**
 * Created by IntelliJ IDEA.
 * Project : online-banking



 * Date: 06/08/18
 * Time: 20.44
 * To change this template use File | Settings | File Templates.
 */
@Entity
@Table(name = "user_role")
public class UserRole {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private long userRoleId;

    @ManyToOne(fetch = FetchType.EAGER)
    @JoinColumn(name = "user_id")
    private User user;

    @ManyToOne(fetch = FetchType.EAGER)
    @JoinColumn(name = "role_id")
    private Role role;


    public UserRole(User user, Role role) {
        this.user = user;
        this.role = role;
    }

    public UserRole() {
    }

    public long getUserRoleId() {
        return userRoleId;
    }

    public void setUserRoleId(long userRoleId) {
        this.userRoleId = userRoleId;
    }

    public User getUser() {
        return user;
    }

    public void setUser(User user) {
        this.user = user;
    }

    public Role getRole() {
        return role;
    }
    public void setRole(Role role) {
        this.role = role;
    }


}
```

21.AccountServiceImpl.java

```
package com.bank.onlinebanking.service.UserServiceImpl;

import com.bank.onlinebanking.entity.PrimaryAccount;
import com.bank.onlinebanking.entity.PrimaryTransaction;
import com.bank.onlinebanking.entity.SavingsAccount;
import com.bank.onlinebanking.entity.SavingsTransaction;
import com.bank.onlinebanking.entity.User;
import com.bank.onlinebanking.repository.PrimaryAccountDao;
import com.bank.onlinebanking.repository.SavingsAccountDao;
import com.bank.onlinebanking.service.AccountService;
import com.bank.onlinebanking.service.TransactionService;
import com.bank.onlinebanking.service.UserService;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.math.BigDecimal;
import java.security.Principal;
import java.util.Date;

/**
 * Created by IntelliJ IDEA.
 * Project : online-banking



 * Date: 09/08/18
 * Time: 04.36
 * To change this template use File | Settings | File Templates.
 */
@Service
public class AccountServiceImpl implements AccountService {

    private static int nextAccountNumber = 11223101;

    @Autowired
    private PrimaryAccountDao primaryAccountDao;
```

```
  @Autowired
  private SavingsAccountDao savingsAccountDao;

  @Autowired
  private UserService userService;

  @Autowired
  private TransactionService transactionService;

  public PrimaryAccount createPrimaryAccount() {
    PrimaryAccount primaryAccount = new PrimaryAccount();
    primaryAccount.setAccountBalance(new BigDecimal(0.0));
    primaryAccount.setAccountNumber(accountGen());

    primaryAccountDao.save(primaryAccount);

    return primaryAccountDao.findByAccountNumber(primaryAccount.getAccountNumber());
  }

  public SavingsAccount createSavingsAccount() {
    SavingsAccount savingsAccount = new SavingsAccount();
    savingsAccount.setAccountBalance(new BigDecimal(0.0));
    savingsAccount.setAccountNumber(accountGen());

    savingsAccountDao.save(savingsAccount);

    return savingsAccountDao.findByAccountNumber(savingsAccount.getAccountNumber());
  }

  public void deposit(String accountType, double amount, Principal principal) {
    User user = userService.findByUsername(principal.getName());

    if (accountType.equalsIgnoreCase("Primary")) {
      PrimaryAccount primaryAccount = user.getPrimaryAccount();
      primaryAccount.setAccountBalance(primaryAccount.getAccountBalance().add(new
BigDecimal(amount)));
      primaryAccountDao.save(primaryAccount);

      Date date = new Date();

      PrimaryTransaction primaryTransaction = new PrimaryTransaction(date, "Deposit to Primary
Account", "Account", "Finished", amount, primaryAccount.getAccountBalance(), primaryAccount);
      transactionService.savePrimaryDepositTransaction(primaryTransaction);

    } else if (accountType.equalsIgnoreCase("Savings")) {
      SavingsAccount savingsAccount = user.getSavingsAccount();
      savingsAccount.setAccountBalance(savingsAccount.getAccountBalance().add(new
BigDecimal(amount)));
      savingsAccountDao.save(savingsAccount);

      Date date = new Date();
      SavingsTransaction savingsTransaction = new SavingsTransaction(date, "Deposit to savings
Account", "Account", "Finished", amount, savingsAccount.getAccountBalance(), savingsAccount);
      transactionService.saveSavingsDepositTransaction(savingsTransaction);
    }
  }

  public void withdraw(String accountType, double amount, Principal principal) {
    User user = userService.findByUsername(principal.getName());

    if (accountType.equalsIgnoreCase("Primary")) {
      PrimaryAccount primaryAccount = user.getPrimaryAccount();
      primaryAccount.setAccountBalance(primaryAccount.getAccountBalance().subtract(new
BigDecimal(amount)));
      primaryAccountDao.save(primaryAccount);

      Date date = new Date();

      PrimaryTransaction primaryTransaction = new PrimaryTransaction(date, "Withdraw from
Primary Account", "Account", "Finished", amount, primaryAccount.getAccountBalance(),
primaryAccount);
      transactionService.savePrimaryWithdrawTransaction(primaryTransaction);
    } else if (accountType.equalsIgnoreCase("Savings")) {
      SavingsAccount savingsAccount = user.getSavingsAccount();
      savingsAccount.setAccountBalance(savingsAccount.getAccountBalance().subtract(new
BigDecimal(amount)));
      savingsAccountDao.save(savingsAccount);

      Date date = new Date();
      SavingsTransaction savingsTransaction = new SavingsTransaction(date, "Withdraw from savings
Account", "Account", "Finished", amount, savingsAccount.getAccountBalance(), savingsAccount);
      transactionService.saveSavingsWithdrawTransaction(savingsTransaction);
    }
  }

  private int accountGen() {
    return ++nextAccountNumber;
  }


}
```

22.AppointmentServiceImpl.java:

```
package com.bank.onlinebanking.service.UserServiceImpl;

import com.bank.onlinebanking.entity.Appointment;
import com.bank.onlinebanking.repository.AppointmentDao;
import com.bank.onlinebanking.service.AppointmentService;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

/**
 * Created by IntelliJ IDEA.
 * Project : online-banking



 * Date: 09/08/18
 * Time: 04.38
 * To change this template use File | Settings | File Templates.
 */
@Service
public class AppointmentServiceImpl implements AppointmentService {

   @Autowired
   private AppointmentDao appointmentDao;

   public Appointment createAppointment(Appointment appointment) {
      return appointmentDao.save(appointment);
   }

   public List<Appointment> findAll() {
      return appointmentDao.findAll();
   }

   public Appointment findAppointment(Long id) {
      return appointmentDao.findById(id).get();
   }

   public void confirmAppointment(Long id) {
      Appointment appointment = findAppointment(id);
      appointment.setConfirmed(true);
      appointmentDao.save(appointment);
   }
}
```

23. TransactionServiceImpl.java:

```
package com.bank.onlinebanking.service.UserServiceImpl;

import com.bank.onlinebanking.entity.PrimaryAccount;
import com.bank.onlinebanking.entity.PrimaryTransaction;
import com.bank.onlinebanking.entity.Recipient;
import com.bank.onlinebanking.entity.SavingsAccount;
import com.bank.onlinebanking.entity.SavingsTransaction;
import com.bank.onlinebanking.entity.User;
import com.bank.onlinebanking.repository.PrimaryAccountDao;
import com.bank.onlinebanking.repository.PrimaryTransactionDao;
import com.bank.onlinebanking.repository.RecipientDao;
import com.bank.onlinebanking.repository.SavingsAccountDao;
import com.bank.onlinebanking.repository.SavingsTransactionDao;
import com.bank.onlinebanking.service.TransactionService;
import com.bank.onlinebanking.service.UserService;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.math.BigDecimal;
import java.security.Principal;
import java.util.Date;
import java.util.List;
import java.util.stream.Collectors;

/**
 * Created by IntelliJ IDEA.
 * Project : online-banking



 * Date: 10/08/18
 * Time: 06.21
 * To change this template use File | Settings | File Templates.
 */
@Service
public class TransactionServiceImpl implements TransactionService {

    @Autowired
    private UserService userService;

    @Autowired
    private PrimaryTransactionDao primaryTransactionDao;
    @Autowired
    private SavingsTransactionDao savingsTransactionDao;

    @Autowired
    private PrimaryAccountDao primaryAccountDao;

    @Autowired
    private SavingsAccountDao savingsAccountDao;

    @Autowired
    private RecipientDao recipientDao;


    public List<PrimaryTransaction> findPrimaryTransactionList(String username) {
        User user = userService.findByUsername(username);
        List<PrimaryTransaction> primaryTransactionList =
user.getPrimaryAccount().getPrimaryTransactionList();

        return primaryTransactionList;
    }

    public List<SavingsTransaction> findSavingsTransactionList(String username) {
        User user = userService.findByUsername(username);
        List<SavingsTransaction> savingsTransactionList =
user.getSavingsAccount().getSavingsTransactionList();

        return savingsTransactionList;
    }

    public void savePrimaryDepositTransaction(PrimaryTransaction primaryTransaction) {
        primaryTransactionDao.save(primaryTransaction);
    }

    public void saveSavingsDepositTransaction(SavingsTransaction savingsTransaction) {
        savingsTransactionDao.save(savingsTransaction);
    }

    public void savePrimaryWithdrawTransaction(PrimaryTransaction primaryTransaction) {
        primaryTransactionDao.save(primaryTransaction);
    }

    public void saveSavingsWithdrawTransaction(SavingsTransaction savingsTransaction) {
        savingsTransactionDao.save(savingsTransaction);
    }

    public void betweenAccountsTransfer(String transferFrom, String transferTo, String amount,
PrimaryAccount primaryAccount, SavingsAccount savingsAccount) throws Exception {
        if (transferFrom.equalsIgnoreCase("Primary") && transferTo.equalsIgnoreCase("Savings")) {
            primaryAccount.setAccountBalance(primaryAccount.getAccountBalance().subtract(new
BigDecimal(amount)));
            savingsAccount.setAccountBalance(savingsAccount.getAccountBalance().add(new
BigDecimal(amount)));
            primaryAccountDao.save(primaryAccount);
            savingsAccountDao.save(savingsAccount);

            Date date = new Date();

            PrimaryTransaction primaryTransaction = new PrimaryTransaction(date, "Between account
transfer from " + transferFrom + " to " + transferTo, "Account", "Finished",
Double.parseDouble(amount), primaryAccount.getAccountBalance(), primaryAccount);
            primaryTransactionDao.save(primaryTransaction);
        } else if (transferFrom.equalsIgnoreCase("Savings") && transferTo.equalsIgnoreCase("Primary")) {
            primaryAccount.setAccountBalance(primaryAccount.getAccountBalance().add(new
BigDecimal(amount)));
            savingsAccount.setAccountBalance(savingsAccount.getAccountBalance().subtract(new
BigDecimal(amount)));
            primaryAccountDao.save(primaryAccount);
            savingsAccountDao.save(savingsAccount);

            Date date = new Date();

            SavingsTransaction savingsTransaction = new SavingsTransaction(date, "Between account
transfer from " + transferFrom + " to " + transferTo, "Transfer", "Finished",
Double.parseDouble(amount), savingsAccount.getAccountBalance(), savingsAccount);
            savingsTransactionDao.save(savingsTransaction);
        } else {
            throw new Exception("Invalid Transfer");
        }
    }

    public List<Recipient> findRecipientList(Principal principal) {
        String username = principal.getName();
        List<Recipient> recipientList = recipientDao.findAll().stream()
                .filter(recipient -> username.equals(recipient.getUser().getUsername()))
                .collect(Collectors.toList());

        return recipientList;
    }

    public Recipient saveRecipient(Recipient recipient) {
        return recipientDao.save(recipient);
    }

    public Recipient findRecipientByName(String recipientName) {
        return recipientDao.findByName(recipientName);
    }

    public void deleteRecipientByName(String recipientName) {
        recipientDao.deleteByName(recipientName);
    }

    public void toSomeoneElseTransfer(Recipient recipient, String accountType, String amount,
PrimaryAccount primaryAccount, SavingsAccount savingsAccount) {
        if (accountType.equalsIgnoreCase("Primary")) {
            primaryAccount.setAccountBalance(primaryAccount.getAccountBalance().subtract(new
BigDecimal(amount)));
            primaryAccountDao.save(primaryAccount);

            Date date = new Date();

            PrimaryTransaction primaryTransaction = new PrimaryTransaction(date, "Transfer to recipient "
+ recipient.getName(), "Transfer", "Finished", Double.parseDouble(amount),
primaryAccount.getAccountBalance(), primaryAccount);
            primaryTransactionDao.save(primaryTransaction);
        } else if (accountType.equalsIgnoreCase("Savings")) {
            savingsAccount.setAccountBalance(savingsAccount.getAccountBalance().subtract(new
BigDecimal(amount)));
            savingsAccountDao.save(savingsAccount);

            Date date = new Date();

            SavingsTransaction savingsTransaction = new SavingsTransaction(date, "Transfer to recipient " +
recipient.getName(), "Transfer", "Finished", Double.parseDouble(amount),
savingsAccount.getAccountBalance(), savingsAccount);
            savingsTransactionDao.save(savingsTransaction);
        }
    }
}
```

24. UserSecurityService.java:

```
package com.bank.onlinebanking.service.UserServiceImpl;

import com.bank.onlinebanking.entity.User;
import com.bank.onlinebanking.repository.UserDao;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.core.userdetails.UsernameNotFoundException;
import org.springframework.stereotype.Service;

/**
 * Created by IntelliJ IDEA.
 * Project : online-banking




 * Time: 06.30
 * To change this template use File | Settings | File Templates.
 */
@Service
public class UserSecurityService implements UserDetailsService {

    private static final Logger LOG = LoggerFactory.getLogger(UserSecurityService.class);

    @Autowired
    private UserDao userDao;

    @Override
    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
        User user = userDao.findByUsername(username);
        if (null == user) {
            LOG.warn("Username {} not found", username);
            throw new UsernameNotFoundException("Username " + username + " not found");
        }
        return user;
    }
}
```

25.UserSecurityServiceImpl.java:

```
package com.bank.onlinebanking.service.UserServiceImpl;



import com.bank.onlinebanking.entity.User;

import com.bank.onlinebanking.repository.UserDao;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.core.userdetails.UsernameNotFoundException;
import org.springframework.stereotype.Service;

/**
 * Created by IntelliJ IDEA.
 * Project : online-banking



 * Date: 10/08/18
 * Time: 06.22
 * To change this template use File | Settings | File Templates.
 */
@Service
public class UserSecurityServiceImpl implements UserDetailsService {

    private static final Logger LOG = LoggerFactory.getLogger(UserSecurityServiceImpl.class);

    @Autowired
    private UserDao userDao;
    @Override

    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {

        User user = userDao.findByUsername(username);

        if (null == user) {

            LOG.warn("Username {} not found", username);

            throw new UsernameNotFoundException("Username " + username + " not found");

        }

        return user;

    }

}
```

26. UserServiceImpl.java

```
package com.bank.onlinebanking.service.UserServiceImpl;

import com.bank.onlinebanking.entity.User;
import com.bank.onlinebanking.repository.RoleDao;
import com.bank.onlinebanking.repository.UserDao;
import com.bank.onlinebanking.security.UserRole;
import com.bank.onlinebanking.service.AccountService;
import com.bank.onlinebanking.service.UserService;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;
import java.util.Set;

/**
 * Created by IntelliJ IDEA.
 * Project : online-banking
 * Date: 10/08/18
 * Time: 06.23
 * To change this template use File | Settings | File Templates.
 */
@Service
@Transactional
public class UserServiceImpl implements UserService {

    private static final Logger LOG = LoggerFactory.getLogger(UserService.class);

    @Autowired
    private UserDao userDao;

    @Autowired
    private RoleDao roleDao;

    @Autowired
    private BCryptPasswordEncoder passwordEncoder;

    @Autowired
    private AccountService accountService;

    public void save(User user) {
        userDao.save(user);
    }

    public User findByUsername(String username) {
        return userDao.findByUsername(username);
    }

    public User findByEmail(String email) {
        return userDao.findByEmail(email);
    }


    public User createUser(User user, Set<UserRole> userRoles) {
        User localUser = userDao.findByUsername(user.getUsername());

        if (localUser != null) {
            LOG.info("User with username {} already exist. Nothing will be done. ", user.getUsername());
        } else {
            String encryptedPassword = passwordEncoder.encode(user.getPassword());
            user.setPassword(encryptedPassword);

            for (UserRole ur : userRoles) {
                roleDao.save(ur.getRole());
            }

            user.getUserRoles().addAll(userRoles);

            user.setPrimaryAccount(accountService.createPrimaryAccount());
            user.setSavingsAccount(accountService.createSavingsAccount());

            localUser = userDao.save(user);
        }

        return localUser;
    }

    public boolean checkUserExists(String username, String email) {
        return checkUsernameExists(username) || checkEmailExists(username);
    }

    public boolean checkUsernameExists(String username) {
        return null != findByUsername(username);

    }

    public boolean checkEmailExists(String email) {
        return null != findByEmail(email);

    }

    public User saveUser(User user) {
        return userDao.save(user);
    }

    public List<User> findUserList() {
        return userDao.findAll();
    }

    public void enableUser(String username) {
        User user = findByUsername(username);
        user.setEnabled(true);
        userDao.save(user);
    }

    public void disableUser(String username) {
        User user = findByUsername(username);
        user.setEnabled(false);
        LOG.info("user status : {}", user.isEnabled());
        userDao.save(user);
        LOG.info(username, "{}, is disabled.");
    }
}
```

27.bootstrap.css

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com"crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Lato:wght@300&display=swap"
rel="stylesheet">
<link href="/css/SignUp.css" rel="stylesheet" type="text/css">
</head>
<body>

<div class="loginBox">
<div class="logo"></div>
<div class="title">SignUp</div>
<form action="UserSignUp" method="post">
<div class="fields">
<div class="username">
<svg class="svg-icon"viewBox="0 0 20 20">
<pathd="M12.075,10.812c1.358-0.853,2.242-2.507,2.242-4.037c0-2.181-1.795-4.618-4.198-
4.618S5.921,4.594,5.921,6.775c0,1.53,0.884,3.185,2.242,4.037c-3.222,0.865-5.6,3.807-
5.6,7.298c0,0.23,0.189,0.42,0.42,0.42h14.273c0.23,0,0.42-0.189,0.42-
0.42C17.676,14.619,15.297,11.677,12.075,10.812 M6.761,6.775c0-2.162,1.773-3.778,3.358-
3.778s3.359,1.616,3.359,3.778c0,2.162-1.774,3.778-3.359,3.778S6.761,8.937,6.761,6.775
M3.415,17.69c0.218-3.51,3.142-6.297,6.704-6.297c3.562,0,6.486,2.787,6.705,6.297H3.415z"></path>
</svg>
<input class="user-input" type="text" placeholder="Choose Your UserName" name="userName"
required>
</div>
<div class="username">
<svg class="svg-icon"viewBox="0 0 20 20">
<pathd="M12.075,10.812c1.358-0.853,2.242-2.507,2.242-4.037c0-2.181-1.795-4.618-4.198-
4.618S5.921,4.594,5.921,6.775c0,1.53,0.884,3.185,2.242,4.037c-3.222,0.865-5.6,3.807-
5.6,7.298c0,0.23,0.189,0.42,0.42,0.42h14.273c0.23,0,0.42-0.189,0.42-
0.42C17.676,14.619,15.297,11.677,12.075,10.812 M6.761,6.775c0-2.162,1.773-3.778,3.358-
3.778s3.359,1.616,3.359,3.778c0,2.162-1.774,3.778-3.359,3.778S6.761,8.937,6.761,6.775
M3.415,17.69c0.218-3.51,3.142-6.297,6.704-6.297c3.562,0,6.486,2.787,6.705,6.297H3.415z"></path>
```

```
</svg>
<input class="user-input" type="email" placeholder="Enter Your email" name="email" required>
</div>
<div class="password">
<svg class="svg-icon"viewBox="0 0 20 20">
<pathd="M17.308,7.564h-1.993c0-2.929-2.385-5.314-5.314-5.314S4.686,4.635,4.686,7.564H2.693c-
0.244,0-0.443,0.2-0.443,0.443v9.3c0,0.243,0.199,0.442,0.443,0.442h14.615c0.243,0,0.442-0.199,0.442-
0.442v-9.3C17.75,7.764,17.551,7.564,17.308,7.564
M10,3.136c2.442,0,4.43,1.986,4.43,4.428H5.571C5.571,5.122,7.558,3.136,10,3.136
M16.865,16.864H3.136V8.45h13.729V16.864z M10,10.664c-0.854,0-1.55,0.696-
1.55,1.551c0,0.699,0.467,1.292,1.107,1.485v0.95c0,0.243,0.2,0.442,0.443,0.442s0.443-0.199,0.443-
0.442V13.7c0.64-0.193,1.106-0.786,1.106-1.485C11.55,11.36,10.854,10.664,10,10.664 M10,12.878c-
0.366,0-0.664-0.298-0.664-0.663c0-0.366,0.298-0.665,0.664-
0.665c0.365,0,0.664,0.299,0.664,0.665C10.664,12.58,10.365,12.878,10,12.878"></path>
</svg>
<input class="user-input" type="text" placeholder="Enter Your name" name="name" required>
</div>
<div class="password">
<svg class="svg-icon"viewBox="0 0 20 20">
<pathd="M17.308,7.564h-1.993c0-2.929-2.385-5.314-5.314-5.314S4.686,4.635,4.686,7.564H2.693c-
0.244,0-0.443,0.2-0.443,0.443v9.3c0,0.243,0.199,0.442,0.443,0.442h14.615c0.243,0,0.442-0.199,0.442-
0.442v-9.3C17.75,7.764,17.551,7.564,17.308,7.564
M10,3.136c2.442,0,4.43,1.986,4.43,4.428H5.571C5.571,5.122,7.558,3.136,10,3.136
M16.865,16.864H3.136V8.45h13.729V16.864z M10,10.664c-0.854,0-1.55,0.696-
1.55,1.551c0,0.699,0.467,1.292,1.107,1.485v0.95c0,0.243,0.2,0.442,0.443,0.442s0.443-0.199,0.443-
0.442V13.7c0.64-0.193,1.106-0.786,1.106-1.485C11.55,11.36,10.854,10.664,10,10.664 M10,12.878c-
0.366,0-0.664-0.298-0.664-0.663c0-0.366,0.298-0.665,0.664-
0.665c0.365,0,0.664,0.299,0.664,0.665C10.664,12.58,10.365,12.878,10,12.878"></path>
</svg>
<input class="user-input" type="password" placeholder="Enter Your password" name="password"
required>
</div>
<div class="signin-button">
<input class="user-input" type="submit" value="Sign Up">
</div>
</div>
</form>
<a href="">Login</a>

</div>
</body>
</html>
```

28.datepicker.css

```
/*!
 * Datetimepicker for Bootstrap
 *
 * Copyright 2012 Stefan Petre
 * Improvements by Andrew Rowls
 * Licensed under the Apache License v2.0
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 */
.datetimepicker {
padding: 4px;
margin-top: 1px;
-webkit-border-radius: 4px;
-moz-border-radius: 4px;
border-radius: 4px;
direction: ltr;
}

.datetimepicker-inline {
width: 220px;
}

.datetimepicker.datetimepicker-rtl {
direction: rtl;
}

.datetimepicker.datetimepicker-rtltabletrtdspan {
float: right;
}

.datetimepicker-dropdown, .datetimepicker-dropdown-left {
top: 0;
left: 0;
}

[class*=" datetimepicker-dropdown"]:before {
content: '';
display: inline-block;
border-left: 7px solid transparent;
border-right: 7px solid transparent;
border-bottom: 7px solid #cccccc;
border-bottom-color: rgba(0, 0, 0, 0.2);
position: absolute;
}

[class*=" datetimepicker-dropdown"]:after {
content: '';
display: inline-block;
border-left: 6px solid transparent;
border-right: 6px solid transparent;
border-bottom: 6px solid #ffffff;
position: absolute;
}

[class*=" datetimepicker-dropdown-top"]:before {
content: '';
display: inline-block;
border-left: 7px solid transparent;
border-right: 7px solid transparent;
border-top: 7px solid #cccccc;
border-top-color: rgba(0, 0, 0, 0.2);
border-bottom: 0;
}

[class*=" datetimepicker-dropdown-top"]:after {
content: '';
display: inline-block;
border-left: 6px solid transparent;
border-right: 6px solid transparent;
border-top: 6px solid #ffffff;
border-bottom: 0;
}

.datetimepicker-dropdown-bottom-left:before {
top: -7px;
right: 6px;
}

.datetimepicker-dropdown-bottom-left:after {
top: -6px;
right: 7px;
}

.datetimepicker-dropdown-bottom-right:before {
top: -7px;
left: 6px;
}

.datetimepicker-dropdown-bottom-right:after {
top: -6px;
left: 7px;
}

.datetimepicker-dropdown-top-left:before {
bottom: -7px;
right: 6px;
}

.datetimepicker-dropdown-top-left:after {
bottom: -6px;
right: 7px;
}
```

```css
.datetimepicker-dropdown-top-right:before {
bottom: -7px;
left: 6px;
}

.datetimepicker-dropdown-top-right:after {
bottom: -6px;
left: 7px;
}

.datetimepicker>div {
display: none;
}

.datetimepicker.minutesdiv.datetimepicker-minutes {
display: block;
}

.datetimepicker.hoursdiv.datetimepicker-hours {
display: block;
}

.datetimepicker.daysdiv.datetimepicker-days {
display: block;
}

.datetimepicker.monthsdiv.datetimepicker-months {
display: block;
}

.datetimepicker.yearsdiv.datetimepicker-years {
display: block;
}

.datetimepickertable {
margin: 0;
}

.datetimepickertd,
.datetimepickerth {
text-align: center;
width: 20px;
height: 20px;
-webkit-border-radius: 4px;
-moz-border-radius: 4px;
border-radius: 4px;
border: none;
}

.table-striped.datetimepickertabletrtd,
.table-striped.datetimepickertabletrth {
background-color: transparent;
}
```

```
.datetimepickertabletrtd.minute:hover {
background: #eeeeee;
cursor: pointer;
}

.datetimepickertabletrtd.hour:hover {
background: #eeeeee;
cursor: pointer;
}

.datetimepickertabletrtd.day:hover {
background: #eeeeee;
cursor: pointer;
}

.datetimepickertabletrtd.old,
.datetimepickertabletrtd.new {
color: #999999;
}

.datetimepickertabletrtd.disabled,
.datetimepickertabletrtd.disabled:hover {
background: none;
color: #999999;
cursor: default;
}

.datetimepickertabletrtd.today,
.datetimepickertabletrtd.today:hover,
.datetimepickertabletrtd.today.disabled,
.datetimepickertabletrtd.today.disabled:hover {
background-color: #fde19a;
background-image: -moz-linear-gradient(top, #fdd49a, #fdf59a);
background-image: -ms-linear-gradient(top, #fdd49a, #fdf59a);
background-image: -webkit-gradient(linear, 00, 0100%, from(#fdd49a), to(#fdf59a));
background-image: -webkit-linear-gradient(top, #fdd49a, #fdf59a);
background-image: -o-linear-gradient(top, #fdd49a, #fdf59a);
background-image: linear-gradient(top, #fdd49a, #fdf59a);
background-repeat: repeat-x;
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#fdd49a',
endColorstr='#fdf59a', GradientType=0);
border-color: #fdf59a#fdf59a#fbed50;
border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25);
filter: progid:DXImageTransform.Microsoft.gradient(enabled=false);
}

.datetimepickertabletrtd.today:hover,
.datetimepickertabletrtd.today:hover:hover,
.datetimepickertabletrtd.today.disabled:hover,
.datetimepickertabletrtd.today.disabled:hover:hover,
.datetimepickertabletrtd.today:active,
.datetimepickertabletrtd.today:hover:active,
.datetimepickertabletrtd.today.disabled:active,
.datetimepickertabletrtd.today.disabled:hover:active,
.datetimepickertabletrtd.today.active,
.datetimepickertabletrtd.today:hover.active,
.datetimepickertabletrtd.today.disabled.active,
.datetimepickertabletrtd.today.disabled:hover.active,
.datetimepickertabletrtd.today.disabled,
.datetimepickertabletrtd.today:hover.disabled,
.datetimepickertabletrtd.today.disabled.disabled,
.datetimepickertabletrtd.today.disabled:hover.disabled,
.datetimepickertabletrtd.today[disabled],
.datetimepickertabletrtd.today:hover[disabled],
.datetimepickertabletrtd.today.disabled[disabled],
.datetimepickertabletrtd.today.disabled:hover[disabled] {
background-color: #fdf59a;
}

.datetimepickertabletrtd.today:active,
.datetimepickertabletrtd.today:hover:active,
.datetimepickertabletrtd.today.disabled:active,
.datetimepickertabletrtd.today.disabled:hover:active,
.datetimepickertabletrtd.today.active,
.datetimepickertabletrtd.today:hover.active,
.datetimepickertabletrtd.today.disabled.active,
.datetimepickertabletrtd.today.disabled:hover.active {
background-color: #fbf069;
}

.datetimepickertabletrtd.active,
.datetimepickertabletrtd.active:hover,
.datetimepickertabletrtd.active.disabled,
.datetimepickertabletrtd.active.disabled:hover {
background-color: #006dcc;
background-image: -moz-linear-gradient(top, #0088cc, #0044cc);
background-image: -ms-linear-gradient(top, #0088cc, #0044cc);
background-image: -webkit-gradient(linear, 00, 0100%, from(#0088cc), to(#0044cc));
background-image: -webkit-linear-gradient(top, #0088cc, #0044cc);
background-image: -o-linear-gradient(top, #0088cc, #0044cc);
background-image: linear-gradient(top, #0088cc, #0044cc);
background-repeat: repeat-x;
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#0088cc',
endColorstr='#0044cc', GradientType=0);
border-color: #0044cc#0044cc#002a80;
border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25);
filter: progid:DXImageTransform.Microsoft.gradient(enabled=false);
color: #ffffff;
text-shadow: 0-1px0rgba(0, 0, 0, 0.25);
}

.datetimepickertabletrtd.active:hover,
.datetimepickertabletrtd.active:hover:hover,
.datetimepickertabletrtd.active.disabled:hover,
.datetimepickertabletrtd.active.disabled:hover:hover,
.datetimepickertabletrtd.active:active,
.datetimepickertabletrtd.active:hover:active,
.datetimepickertabletrtd.active.disabled:active,
.datetimepickertabletrtd.active.disabled:hover:active,
.datetimepickertabletrtd.active.active,
.datetimepickertabletrtd.active:hover.active,
.datetimepickertabletrtd.active.disabled.active,
.datetimepickertabletrtd.active.disabled:hover.active,
.datetimepickertabletrtd.active.disabled,
.datetimepickertabletrtd.active:hover.disabled,
.datetimepickertabletrtd.active.disabled.disabled,
.datetimepickertabletrtd.active.disabled:hover.disabled,
.datetimepickertabletrtd.active[disabled],
.datetimepickertabletrtd.active:hover[disabled],
.datetimepickertabletrtd.active.disabled[disabled],
.datetimepickertabletrtd.active.disabled:hover[disabled] {
background-color: #0044cc;
}

.datetimepickertabletrtd.active:active,
.datetimepickertabletrtd.active:hover:active,
.datetimepickertabletrtd.active.disabled:active,
.datetimepickertabletrtd.active.disabled:hover:active,
.datetimepickertabletrtd.active.active,
.datetimepickertabletrtd.active:hover.active,
.datetimepickertabletrtd.active.disabled.active,
.datetimepickertabletrtd.active.disabled:hover.active {
background-color: #003399;
}

.datetimepickertabletrtdspan {
display: block;
width: 23%;
height: 54px;
line-height: 54px;
float: left;
margin: 1%;
cursor: pointer;
-webkit-border-radius: 4px;
-moz-border-radius: 4px;
border-radius: 4px;
}

.datetimepicker.datetimepicker-hoursspan {
height: 26px;
line-height: 26px;
}

.datetimepicker.datetimepicker-hourstabletrtdspan.hour_am,
.datetimepicker.datetimepicker-hourstabletrtdspan.hour_pm {
width: 14.6%;
}

.datetimepicker.datetimepicker-hoursfieldsetlegend,
.datetimepicker.datetimepicker-minutesfieldsetlegend {
margin-bottom: inherit;
line-height: 30px;
}

.datetimepicker.datetimepicker-minutesspan {
height: 26px;
line-height: 26px;
}

.datetimepickertabletrtdspan:hover {
background: #eeeeee;
}

.datetimepickertabletrtdspan.disabled,
.datetimepickertabletrtdspan.disabled:hover {
background: none;
color: #999999;
cursor: default;
}

.datetimepickertabletrtdspan.active,
.datetimepickertabletrtdspan.active:hover,
.datetimepickertabletrtdspan.active.disabled,
.datetimepickertabletrtdspan.active.disabled:hover {
background-color: #006dcc;
background-image: -moz-linear-gradient(top, #0088cc, #0044cc);
background-image: -ms-linear-gradient(top, #0088cc, #0044cc);
background-image: -webkit-gradient(linear, 00, 0100%, from(#0088cc), to(#0044cc));
background-image: -webkit-linear-gradient(top, #0088cc, #0044cc);
background-image: -o-linear-gradient(top, #0088cc, #0044cc);
background-image: linear-gradient(top, #0088cc, #0044cc);
background-repeat: repeat-x;
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#0088cc',
endColorstr='#0044cc', GradientType=0);
border-color: #0044cc#0044cc#002a80;
border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25);
filter: progid:DXImageTransform.Microsoft.gradient(enabled=false);
color: #ffffff;
text-shadow: 0-1px0rgba(0, 0, 0, 0.25);
}

.datetimepickertabletrtdspan.active:hover,
.datetimepickertabletrtdspan.active:hover:hover,
.datetimepickertabletrtdspan.active.disabled:hover,
.datetimepickertabletrtdspan.active.disabled:hover:hover,
.datetimepickertabletrtdspan.active:active,
.datetimepickertabletrtdspan.active:hover:active,
.datetimepickertabletrtdspan.active.disabled:active,
.datetimepickertabletrtdspan.active.disabled:hover:active,
.datetimepickertabletrtdspan.active.active,
.datetimepickertabletrtdspan.active:hover.active,
.datetimepickertabletrtdspan.active.disabled.active,
.datetimepickertabletrtdspan.active.disabled:hover.active,
.datetimepickertabletrtdspan.active.disabled,
.datetimepickertabletrtdspan.active:hover.disabled,
.datetimepickertabletrtdspan.active.disabled.disabled,
.datetimepickertabletrtdspan.active.disabled:hover.disabled,
.datetimepickertabletrtdspan.active[disabled],
.datetimepickertabletrtdspan.active:hover[disabled],
.datetimepickertabletrtdspan.active.disabled[disabled],
.datetimepickertabletrtdspan.active.disabled:hover[disabled] {
background-color: #0044cc;
}

.datetimepickertabletrtdspan.active:active,
.datetimepickertabletrtdspan.active:hover:active,
.datetimepickertabletrtdspan.active.disabled:active,
.datetimepickertabletrtdspan.active.disabled:hover:active,
.datetimepickertabletrtdspan.active.active,
.datetimepickertabletrtdspan.active:hover.active,
.datetimepickertabletrtdspan.active.disabled.active,
.datetimepickertabletrtdspan.active.disabled:hover.active {
background-color: #003399;
}

.datetimepickertabletrtdspan.old {
color: #999999;
}

.datetimepickerth.switch {
width: 145px;
}

.datetimepickerthspan.glyphicon {
pointer-events: none;
}

.datetimepickertheadtr:first-childth,
.datetimepickertfootth {
cursor: pointer;
}

.datetimepickertheadtr:first-childth:hover,
.datetimepickertfootth:hover {
background: #eeeeee;
}

.input-append.date.add-oni,
.input-prepend.date.add-oni,
.input-group.date.input-group-addonspan {
cursor: pointer;
width: 14px;
height: 14px;
}
```

29.main.css

```
body {
background-color: transparent;
}

html {
background: url("/images/111sbg.png") no-repeat center center fixed;
-webkit-background-size: cover;
-moz-background-size: cover;
-o-background-size: cover;
background-size: cover;
}

img {
display: block;
margin-left: auto;
margin-right: auto;
}

.form-basic {
max-width: 640px;
margin: 0 auto;
padding: 55px;
box-sizing: border-box;
background-color: #ffffff;
box-shadow: 01px3px0rgba(0, 0, 0, 0.1);
font: bold 14px sans-serif;
text-align: center;
}

.form-basic.form-row {
text-align: left;
margin-bottom: 22px;
}

.form-basic.form-title-row {
text-align: center;
margin-bottom: 55px;
}

/* The form title */
.form-basich1 {
display: inline-block;
box-sizing: border-box;
color: #4c565e;
font-size: 24px;
padding: 010px15px;
border-bottom: 2px solid #6caee0;
margin: 0;
}

.form-basic.form-row>labelspan {
display: inline-block;
box-sizing: border-box;
color: #5F5F5F;
width: 180px;
text-align: right;
vertical-align: top;
padding: 12px25px;
}

.form-basicinput {
color: #5f5f5f;
box-sizing: border-box;
width: 240px;
box-shadow: 1px2px4px0rgba(0, 0, 0, 0.08);
padding: 12px;
border: 1px solid #dbdbdb;
}

.form-basicinput[type=radio], .form-basicinput[type=checkbox] {
box-shadow: none;
width: auto;
}

.form-basicinput[type=checkbox] {
margin-top: 13px;
}

.form-basicselect {
background-color: #ffffff;
color: #5f5f5f;
box-sizing: border-box;
max-width: 240px;
box-shadow: 1px2px4px0rgba(0, 0, 0, 0.08);
padding: 12px8px;
border: 1px solid #dbdbdb;
}

.form-basictextarea {
color: #5f5f5f;
box-sizing: border-box;
width: 240px;
height: 80px;
box-shadow: 1px2px4px0rgba(0, 0, 0, 0.08);
font: normal 13px sans-serif;
padding: 12px;
border: 1px solid #dbdbdb;
resize: vertical;
}

.form-basic.form-radio-buttons {
display: inline-block;
vertical-align: top;
}

.form-basic.form-radio-buttons>div {
margin-top: 10px;
}
```

```
.form-basic.form-radio-buttonslabelspan {
margin-left: 8px;
color: #5f5f5f;
font-weight: normal;
}

.form-basic.form-radio-buttonsinput {
width: auto;
}

.form-basicbutton {
display: block;
border-radius: 2px;
background-color: #6caee0;
color: #ffffff;
font-weight: bold;
box-shadow: 1px2px4px0rgba(0, 0, 0, 0.08);
padding: 14px22px;
border: 0;
margin: 40px183px0;
}

/*     Making the form responsive. Remove this media query
    if you don't need the form to work on mobile devices. */
@media ( max-width: 600px) {
.form-basic {
padding: 30px;
max-width: 480px;
    }

.form-basic.form-row {
max-width: 300px;
margin: 25px auto;
text-align: left;
    }

.form-basic.form-title-row {
margin-bottom: 50px;
    }

.form-basic.form-row>labelspan {
display: block;
text-align: left;
padding: 0015px;
    }

.form-basicselect {
width: 240px;
    }

.form-basicinput[type=checkbox] {
margin-top: 0;
    }
```

```
.form-basic.form-radio-buttons>div {
margin: 0010px;
    }

.form-basicbutton {
margin: 0;
    }
}

.main {
margin-top: 70px;
}

.main-center {
margin-top: 30px;
margin: 0 auto;
max-width: 330px;
padding: 40px40px;
}

.responstable {
margin: 1em0;
width: 100%;
overflow: hidden;
background: #FFF;
color: #024457;
border-radius: 10px;
border: 1px solid #167F92;
}

.responstabletr {
border: 1px solid #D9E4E6;
}

.responstabletr:nth-child(odd) {
background-color: #EAF3F3;
}

.responstableth {
display: none;
border: 1px solid #FFF;
background-color: #167F92;
color: #FFF;
padding: 1em;
}

.responstableth:first-child {
display: table-cell;
text-align: center;
}

.responstableth:nth-child(2) {
display: table-cell;
```

```
}

.responstableth:nth-child(2) span {
display: none;
}

.responstableth:nth-child(2):after {
content: attr(data-th);
}

@media ( min-width: 480px) {
.responstableth:nth-child(2) span {
display: block;
    }

.responstableth:nth-child(2):after {
display: none;
    }
}

.responstabletd {
display: block;
word-wrap: break-word;
max-width: 7em;
}

.responstabletd:first-child {
display: table-cell;
text-align: center;
border-right: 1px solid #D9E4E6;
}

@media ( min-width: 480px) {
.responstabletd {
border: 1px solid #D9E4E6;
    }
}

.responstableth, .responstabletd {
text-align: left;
margin: .5em1em;
}

@media ( min-width: 480px) {
.responstableth, .responstabletd {
display: table-cell;
padding: 1em;
    }
}

body {
padding: 02em;
font-family: Arial, sans-serif;
color: #024457;
```

```
/*background: #f2f2f2;*/
}

h1 {
font-family: Verdana;
font-weight: normal;
/*color: #024457;*/
}

h1span {
color: #167F92;
}

.box {
border: 2px solid red;
}

hr {
-moz-border-bottom-colors: none;
-moz-border-image: none;
-moz-border-left-colors: none;
-moz-border-right-colors: none;
-moz-border-top-colors: none;
border-color: #EEEEEE -moz-use-text-color #FFFFFF;
border-style: solid none;
border-width: 1px0;
margin: 18px0;
border: 0;
height: 1px;
background: #333;
background-image: linear-gradient(to right, #ccc, #333, #ccc);
}
```

==========================THE END=====================

THANK YOU SIMPLILEARN

TATA SAI VINEETH