

SOURSE CODE OF ICIN-BANK

Developer Name -SURAJ DUBEY

BACKEND(admin_service_new)

admin_service_new>src/main/java

Packages---com.admin

class-AdminServiceApplication.java

```
package com.admin;
```

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

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

```
@SpringBootApplication
```

```
public class AdminServiceApplication {
```

```
    public static void main(String[] args) {
```

```
        SpringApplication.run(AdminServiceApplication.class, args);
```

```
    }
```

```
}
```

package>com.admin.controller

class-AdminController.java

```
package com.admin.controller;
```

```
import java.util.List;
```

```
import org.apache.commons.mail.EmailException;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.web.bind.annotation.CrossOrigin;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.PathVariable;
```

```
import org.springframework.web.bind.annotation.RestController;
```

```
import com.admin.model.ChequebookRequest;
```

```
import com.admin.model.User;
```

```
import com.admin.model.UserDisplay;
```

```
import com.admin.service.impl.AdminServiceImpl;
```

```
import com.admin.service.impl.MailServiceImpl;
```

```
@RestController
```

```
@CrossOrigin
```

```
public class AdminController {
```

```
    @Autowired
```

```
    private AdminServiceImpl service;
```

```
    @Autowired
```

```
    private MailServiceImpl mailService;
```

```

    @GetMapping("user/{username}/features/{featureId}")

    public void setUserFeatures(@PathVariable("username") String username,
    @PathVariable("featureId") int featureId) {

        service.setUserFeatures(username, featureId);

    }

```

```

    @GetMapping("user/{username}/authorize")

    public void authorizeUser(@PathVariable("username") String username) {

        try {

            service.authorizeUser(username);

            mailService.sendAuthorizedEmail(username);

        } catch (EmailException e) {

            // TODO Auto-generated catch block

            e.printStackTrace();

        }

    }

```

```

    @GetMapping("user/{username}/authorize/cancel")

    public void cancelAuthorization(@PathVariable("username") String username) {

        try {

            mailService.sendAuthorizeCancelEmail(username);

        } catch (EmailException e) {

            // TODO Auto-generated catch block

            e.printStackTrace();

        }

        service.cancelAuthorization(username);
    }

```

```
}
```

```
@GetMapping("user/unauthorized/all")
```

```
public List<User> getAllUnauthorizedUsers()
```

```
{
```

```
    return service.getAllUnauthorizedUsers();
```

```
}
```

```
@GetMapping("/user/all")
```

```
public List<UserDisplay> getAllUsers()
```

```
{
```

```
    return service.getAllUsers();
```

```
}
```

```
// @GetMapping("/user/transfers/{account}")
```

```
// public List<Transfer> getTransactionList(@PathVariable("account") int account)
```

```
// {
```

```
//     return service.getAllTransactions(account);
```

```
// }
```

```
@GetMapping("/chequebook/request/all")
```

```
public List<ChequebookRequest> getAllChequeBookRequests()
```

```
{
```

```
    return service.getAllChequebookRequests();
```

```
}
```

```
@GetMapping("/user/{accNo}/chequebook/request/confirm")

public void confirmChequeBookRequest(@PathVariable("accNo") long accNo)

{

    service.acceptChequebookRequest(accNo);

}
```

```
@GetMapping("/user/{username}/enable")

public void enableUser(@PathVariable("username") String username)

{

    service.enableUser(username);

}
```

```
@GetMapping("/user/{username}/disable")

public void disableUser(@PathVariable("username") String username)

{

    service.disableUser(username);

}
```

```
@GetMapping("search/user/{userDetail}")

public UserDisplay searchUser(@PathVariable("userDetail") String userDetail) {

    return service.searchUser(userDetail);

}
```

```
}
```

package>com.admin.dao

Interface>AccountRepository.java

package com.admin.dao;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

import com.admin.model.Account;

@Repository

public interface AccountRepository extends CrudRepository<Account, Integer>{

 public Account findByUsername(String username);

 public Account findByAccno(long accno);

}

second interface

package com.admin.dao;

import java.util.List;

import org.springframework.data.jpa.repository.Modifying;

import org.springframework.transaction.annotation.Transactional;

import org.springframework.data.jpa.repository.Query;

import org.springframework.data.repository.CrudRepository;

```
import org.springframework.stereotype.Repository;
```

```
import com.admin.model.ChequebookRequest;
```

```
@Repository
```

```
public interface ChequeBookRequestsRepository extends CrudRepository<ChequebookRequest,  
Integer>{
```

```
    @Modifying
```

```
    @Transactional
```

```
    @Query("update ChequebookRequest c set c.requestStatus=1 where c.account = ?1")
```

```
    void setChequebookInfoByAccount(long accNo);
```

```
    @Query("FROM ChequebookRequest c where c.requestStatus=FALSE")
```

```
    public List<ChequebookRequest> findAllChequebookRequests();
```

```
}
```

third Intreface

```
package com.admin.dao;
```

```
import org.springframework.data.jpa.repository.JpaRepository;
```

```
import org.springframework.stereotype.Repository;
```

```
import com.admin.model.Saccount;
```

```
@Repository
```

```
public interface SaccountRepository extends JpaRepository<Saccount, Integer>{

    public Saccount findByAccno(long accNo);

}
```

Fourth Inteface

```
package com.admin.dao;
```

```
import java.util.List;
```

```
import org.springframework.data.repository.CrudRepository;
```

```
import org.springframework.stereotype.Repository;
```

```
import com.admin.model.Transfer;
```

```
@Repository
```

```
public interface TransferRepository extends CrudRepository<Transfer, Integer> {
```

```
    public List<Transfer> findBySaccount(long saccount);
```

```
    public List<Transfer> findByRaccount(long racount);
```

```
}
```

Fifth Interface

```
package com.admin.dao;
```

```
import java.util.List;
```



```
import org.springframework.data.jpa.repository.JpaRepository;
```

```
import org.springframework.data.jpa.repository.Query;
```

```
import com.admin.model.User;
```

```
import com.admin.model.UserDisplay;
```

```
public interface UserDisplayRepository extends JpaRepository<User, Integer>{
```

```
    @Query("SELECT new  
com.admin.model.UserDisplay(u.fname,u.lname,u.phone,u.username,u.status,u.featureStatus,a.accno,a  
.balance,s.accno,s.balance)" + "FROM User u ,Account a,Saccount s WHERE u.username=a.username  
and u.username=s.username")
```

```
    public List<UserDisplay> getAllUsers();
```

```
    @Query("SELECT new  
com.admin.model.UserDisplay(u.fname,u.lname,u.phone,u.username,u.status,u.featureStatus,a.accno,a  
.balance,s.accno,s.balance)" + "FROM User u ,Account a,Saccount s WHERE u.username=?1 and  
u.username=a.username and u.username=s.username")
```

```
    public UserDisplay getUserDetailsByUsername(String userDetail);
```

```
    @Query("SELECT new  
com.admin.model.UserDisplay(u.fname,u.lname,u.phone,u.username,u.status,u.featureStatus,a.accno,a  
.balance,s.accno,s.balance)" + "FROM User u ,Account a,Saccount s WHERE s.accno=?1 and  
u.username=a.username and u.username=s.username")
```

```
    public UserDisplay getUserDetailsByAccountNo(long accNo);
```

```
}
```

Sixth Interface

```
package com.admin.dao;
```

```
import org.springframework.data.repository.CrudRepository;
```

```
import org.springframework.stereotype.Repository;
```

```
import org.springframework.transaction.annotation.Transactional;
```

```
import com.admin.model.User;
```

```
import java.util.List;
```

```
import org.springframework.data.jpa.repository.Modifying;
```

```
import org.springframework.data.jpa.repository.Query;
```

```
@Repository
```

```
public interface UserRepository extends CrudRepository<User, Integer> {
```

```
    @Query("FROM User u WHERE u.username=?1")
```

```
    public User findByUsername(String username);
```

```
    @Modifying
```

```
    @Transactional
```

```
    @Query("update User u set u.status=1 where u.username = ?1")
```

```
    void disableUser(String username);
```

@Modifying

@Transactional

@Query("update User u set u.status=0 where u.username = ?1")

void enableUser(String username);

@Modifying

@Transactional

@Query("update User u set u.authorizationStatus=1 where u.username = ?1")

void authorizeUser(String username);

@Modifying

@Transactional

@Query("delete from User u where u.username = ?1")

void cancelAuthorization(String username);

@Query("FROM User u where u.authorizationStatus=FALSE")

public List<User> findAllUnauthorizedAccounts();

@Modifying

@Transactional

@Query("update User u set u.featureStatus=?2 where u.username = ?1")

void setUserFeatureStatus(String username, int featureId);

}

Package>come.admin.model

class>account.java

```
package com.admin.model;
```

```
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.JoinColumn;
```

```
import javax.persistence.OneToOne;
```

```
import javax.persistence.Table;
```

```
@Entity
```

```
@Table(name="account")
```

```
public class Account {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.AUTO)
```

```
    @Column(name = "id")
```

```
    private int id;
```

```
private long accno;
```

```
private int balance;
```

```
private String username;
```

```
@OneToOne(fetch = FetchType.LAZY)
```

```
@JoinColumn(name = "user_id")
```

```
private User user;
```

```
public String getUsername() {
```

```
    return username;
```

```
}
```

```
public void setUsername(String username) {
```

```
    this.username = username;
```

```
}
```

```
public int getId() {
```

```
    return id;
```

```
}
```

```
public void setId(int id) {
```

```
    this.id = id;
```

```
}
```

```
public long getAccno() {  
    return accno;  
}  
  
public void setAccno(long accno) {  
    this.accno = accno;  
}  
  
public int getBalance() {  
    return balance;  
}  
  
public void setBalance(int balance) {  
    this.balance = balance;  
}  
  
public User getUser() {  
    return user;  
}  
  
public void setUser(User user) {  
    this.user = user;  
}  
}
```

Second Class

```
package com.admin.model;

import java.time.LocalDate;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table
public class ChequebookRequest {

    @Id

    @GeneratedValue(strategy = GenerationType.AUTO)

    private int id;

    private long account;

    private String accType;

    private boolean requestStatus;

    private LocalDate date;

    private int no_of_pages;

    public long getAccount() {

        return account;
    }
}
```

```
}

public void setAccount(long account) {

    this.account = account;

}

public LocalDate getDate() {

    return date;

}

public void setDate(LocalDate date) {

    this.date = date;

}

public int getNo_of_pages() {

    return no_of_pages;

}

public void setNo_of_pages(int no_of_pages) {

    this.no_of_pages = no_of_pages;

}

public String getAccType() {

    return accType;

}

public void setAccType(String accType) {

    this.accType = accType;

}

public boolean getRequestStatus() {

    return requestStatus;

}
```



```

        public void setRequestStatus(boolean requestStatus) {

            this.requestStatus = requestStatus;

        }

        public int getId() {

            return id;

        }

        public void setId(int id) {

            this.id = id;

        }

    }

```

third class

```

package com.admin.model;

```

```

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.OneToOne;

import javax.persistence.Table;

```

```

@Entity

```

```

@Table

```

```

public class Saccount {

```

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private int id;

private long accno;

private int balance;

private String username;

@OneToOne(fetch = FetchType.LAZY)

@JoinColumn(name="user_id")

private User user;

public User getUser() {

 return user;

}

public void setUser(User user) {

 this.user = user;

}

public String getUsername() {

 return username;

```
}
```

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

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

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

```
public long getAccno() {  
    return accno;  
}
```

```
public void setAccno(long accno) {  
    this.accno = accno;  
}
```

```
public int getBalance() {  
    return balance;  
}
```

```

        public void setBalance(int balance) {
            this.balance = balance;
        }
    }
}

```

Fourth Class

```

package com.admin.model;

import java.time.LocalDate;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table

public class Transfer implements Comparable<Transfer>{

    @Id

    @GeneratedValue(strategy = GenerationType.AUTO)

    private int id;

    private long saccount;

```

```
private long raccout;  
  
private int amount;  
  
private LocalDate date;  
  
public int getId() {  
    return id;  
}  
  
public void setId(int id) {  
    this.id = id;  
}  
  
public long getSaccount() {  
    return saccount;  
}  
  
public void setSaccount(long saccount) {  
    this.saccount = saccount;  
}  
  
public long getRaccount() {  
    return raccout;  
}  
  
public void setRaccount(long raccout) {  
    this.raccout = raccout;  
}  
  
public int getAmount() {  
    return amount;  
}
```

```

        public void setAmount(int amount) {

            this.amount = amount;

        }

        public LocalDate getDate() {

            return date;

        }

        public void setDate(LocalDate date) {

            this.date = date;

        }

        @Override

        public int compareTo(Transfer o) {

            Integer i1=this.id;

            Integer i2=o.id;

            return i2.compareTo(i1);

        }

    }

```

Fifth Class

```

package com.admin.model;

import java.sql.Date;

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.OneToOne;

import javax.persistence.Table;


import com.fasterxml.jackson.annotation.JsonFormat;
```

```
@Entity
```

```
@Table(name="user")
```

```
public class User{
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.AUTO)
```

```
    @Column(name = "id")
```

```
    private int id;
```

```
    private String fname;
```

```
    private String lname;
```

```
    private long phone;
```

```
    private String address;
```

```
    private String email;
```

```
    private String username;
```

```
    private String password;
```

```
    @JsonFormat(shape=JsonFormat.Shape.STRING, pattern="dd-MM-yyyy")
```

```
private Date dob;
```

```
private String identityType;
```

```
private String identity;
```

```
@Column(columnDefinition = "boolean default false")
```

```
private boolean status;
```

```
@Column(columnDefinition = "boolean default false")
```

```
private boolean authorizationStatus;
```

```
@Column(columnDefinition = "integer default 3",nullable=false)
```

```
private int featureStatus=3;
```

```
@OneToOne(targetEntity = Account.class,mappedBy = "user",orphanRemoval = false, fetch = FetchType.LAZY)
```

```
private Account account;
```

```
@OneToOne(targetEntity = Saccount.class,mappedBy = "user",orphanRemoval = false, fetch = FetchType.LAZY)
```

```
private Saccount sAccount;
```

```
public Date getDob() {
```

```
    return dob;
```

```
}
```

```
public void setDob(Date dob) {
```

```
    this.dob = dob;
```

```
}
```

```
public String getIdentityType() {
```



```
        return identityType;
    }

    public void setIdentityType(String identityType) {
        this.identityType = identityType;
    }

    public String getIdentity() {
        return identity;
    }

    public void setIdentity(String identity) {
        this.identity = identity;
    }

    public int getFeatureStatus() {
        return featureStatus;
    }

    public void setFeatureStatus(int featureStatus) {
        this.featureStatus = featureStatus;
    }

    public boolean isStatus() {
        return status;
    }

    public boolean isAuthorizationStatus() {
        return authorizationStatus;
    }

    public void setAuthorizationStatus(boolean authorizationStatus) {
        this.authorizationStatus = authorizationStatus;
    }
}
```

```
}

public void setId(int id) {

    this.id = id;

}

public void setStatus(boolean status) {

    this.status = status;

}

public Integer getId() {

    return id;

}

public void setId(Integer id) {

    this.id = id;

}

public String getFname() {

    return fname;

}

public void setFname(String fname) {

    this.fname = fname;

}

public String getLname() {

    return lname;

}

public void setLname(String lname) {

    this.lname = lname;

}
```

```
public long getPhone() {  
    return phone;  
}  
  
public void setPhone(long phone) {  
    this.phone = phone;  
}  
  
public String getAddress() {  
    return address;  
}  
  
public void setAddress(String address) {  
    this.address = address;  
}  
  
public String getEmail() {  
    return email;  
}  
  
public void setEmail(String email) {  
    this.email = email;  
}  
  
public String getUsername() {  
    return username;  
}  
  
public void setUsername(String username) {  
    this.username = username;  
}  
  
public String getPassword() {
```

```
        return password;
    }

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

Sixth Class

```
package com.admin.model;

public class UserDisplay {

    private String fname;
    private String lname;
    private long phone;
    private String username;
    private boolean status;
    private int featureStatus;
    private long primaryAccno;
    private int primaryBalance;
    private long savingsAccno;
    private int savingsBalance;

    public UserDisplay() {

    }
}
```

```
    public UserDisplay(String fname, String lname, long phone, String username,
boolean status, int featureStatus, long primaryAccno,

        int primaryBalance, long savingsAccno, int savingsBalance) {

    super();

    this.fname = fname;

    this.lname = lname;

    this.phone = phone;

    this.username = username;

    this.status = status;

    this.featureStatus = featureStatus;

    this.primaryAccno = primaryAccno;

    this.primaryBalance = primaryBalance;

    this.savingsAccno = savingsAccno;

    this.savingsBalance = savingsBalance;

}
```

```
public String getFname() {

    return fname;

}
```

```
public void setFname(String fname) {

    this.fname = fname;

}
```

```
public String getLname() {
```

```
        return lname;
    }

    public void setLname(String lname) {
        this.lname = lname;
    }

    public long getPhone() {
        return phone;
    }

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

    public String getUsername() {
        return username;
    }

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

    public boolean isStatus() {
        return status;
    }

    public void setStatus(boolean status) {
```

```
        this.status = status;
    }

    public long getPrimaryAccno() {
        return primaryAccno;
    }

    public void setPrimaryAccno(long primaryAccno) {
        this.primaryAccno = primaryAccno;
    }

    public int getPrimaryBalance() {
        return primaryBalance;
    }

    public void setPrimaryBalance(int primaryBalance) {
        this.primaryBalance = primaryBalance;
    }

    public long getSavingsAccno() {
        return savingsAccno;
    }

    public void setSavingsAccno(long savingsAccno) {
        this.savingsAccno = savingsAccno;
    }

    public int getSavingsBalance() {
```

```

        return savingsBalance;
    }

    public void setSavingsBalance(int savingsBalance) {
        this.savingsBalance = savingsBalance;
    }

    public int getFeatureStatus() {
        return featureStatus;
    }

    public void setFeatureStatus(int featureStatus) {
        this.featureStatus = featureStatus;
    }
}

```

Package>com.admin.service

Interface>adminService.java

```
package com.admin.service;
```

```
import java.util.List;
```

```
import com.admin.model.ChequebookRequest;
```

```
import com.admin.model.Transfer;
```

```
import com.admin.model.User;
```

```
import com.admin.model.UserDisplay;
```



```
public interface AdminService {

    public List<UserDisplay> getAllUsers();

    public List<Transfer> getAllTransactions(long accountNo);

    public List<ChequebookRequest> getAllChequebookRequests();

    //public void acceptChequebookRequest(int accNo);

    public void enableUser(String username);

    public void disableUser(String username);

    public void authorizeUser(String username);

    public void cancelAuthorization(String username);

    public List<User> getAllUnauthorizedUsers();

    public void setUserFeatures(String username, int featureId);

    public UserDisplay searchUser(String userDetail);

    void acceptChequebookRequest(long accNo);

}
```

Package >com.admin.service.impl

```
package com.admin.service.impl;

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

import org.springframework.stereotype.Service;

import com.admin.dao.AccountRepository;

import com.admin.dao.UserRepository;
```

```
import com.admin.model.Account;
```

```
@Service
```

```
public class AccountsCreationImpl {
```

```
    @Autowired
```

```
    AccountRepository dao;
```

```
    @Autowired
```

```
    private UserRepository udao;
```

```
    private final String bankCode = "3914";
```

```
    private final String countryCode = "91";
```

```
    private final String branchCode = "820";
```

```
    private final String accountcode="1";
```

```
    public Account newAccount(String username,int userId) {
```

```
        Account account=new Account();
```

```
        account.setUsername(username);
```

```
        account.setAccno(generate_saving(userId));
```

```
        account.setUser(udao.findByUsername(username));
```

```
        return dao.save(account);
```

```
    }
```

```

        public long generate_saving(int userId) {

            String accNo =
bankCode+countryCode+branchCode+accountcode+String.valueOf(userId);

            return Long.parseLong(accNo);

        }

    }

```

```

package com.admin.service.impl;

```

```

import java.util.ArrayList;

```

```

import java.util.Collections;

```

```

import java.util.List;

```

```

import org.apache.commons.mail.EmailException;

```

```

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

```

```

import org.springframework.stereotype.Service;

```

```

import com.admin.dao.AccountRepository;

```

```

import com.admin.dao.ChequeBookRequestsRepository;

```

```

import com.admin.dao.SaccountRepository;

```

```

import com.admin.dao.TransferRepository;

```

```

import com.admin.dao.UserDisplayRepository;

```

```

import com.admin.dao.UserRepository;

```

```

import com.admin.model.ChequebookRequest;

```

```

import com.admin.model.Transfer;

```

```
import com.admin.model.User;
```

```
import com.admin.model.UserDisplay;
```

```
import com.admin.service.AdminService;
```

```
@Service
```

```
public class AdminServiceImpl implements AdminService{
```

```
    @Autowired
```

```
    private TransferRepository transDAO;
```

```
    @Autowired
```

```
    private ChequeBookRequestsRepository chequeBookDAO;
```

```
    @Autowired
```

```
    private UserRepository userDao;
```

```
    @Autowired
```

```
    private UserDisplayRepository displayDao;
```

```
    @Autowired
```

```
    private AccountRepository accDao;
```

```
    @Autowired
```

```
    private SaccountRepository sAccDao;
```

@Autowired

```
private MailServiceImpl mailService;
```

@Autowired

```
private AccountsCreationImpl accService;
```

@Autowired

```
private SaccountCreationImpl sAccService;
```

@Override

```
public void authorizeUser(String username) {  
    userDao.authorizeUser(username);  
    System.out.println("error here top");  
    User currUser = userDao.findByUsername(username);  
    int userId = currUser.getId();  
    System.out.println("error here 1");  
    accService.newAccount(username,userId);  
    System.out.println("error here 2");  
    sAccService.newAccount(username,userId);  
}
```

@Override

```
public void cancelAuthorization(String username) {  
    userDao.cancelAuthorization(username);  
}
```

@Override

```
public void acceptChequebookRequest(long accNo) {  
    String username = "";  
    chequeBookDAO.setChequebookInfoByAccount(accNo);  
    if(Long.toString(accNo).length() == 7) {  
        username = accDao.findByAccno(accNo).getUsername();  
    }  
    else {  
        username = sAccDao.findByAccno(accNo).getUsername();  
    }  
    try {  
        mailService.sendChequebookConfirmedEmail(username);  
    } catch (EmailException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
}
```

@Override

```
public void enableUser(String username) {  
    userDao.enableUser(username);  
  
}
```

@Override

```
public void disableUser(String username) {  
    userDao.disableUser(username);  
  
}
```

@Override

```
public List<UserDisplay> getAllUsers() {  
    return displayDao.getAllUsers();  
}
```

@Override

```
public List<Transfer> getAllTransactions(long accountNo) {  
    List<Transfer> sender=transDAO.findBySaccount(accountNo);  
    List<Transfer> receiver=transDAO.findByRaccount(accountNo);  
    List<Transfer> merged=new ArrayList<>();  
    merged.addAll(sender);  
    merged.addAll(receiver);  
    Collections.sort(merged);  
    return merged;  
}
```

@Override

```
public List<ChequebookRequest> getAllChequebookRequests() {  
    return chequeBookDAO.findAllChequebookRequests();  
}
```

```
}
```

```
@Override
```

```
public List<User> getAllUnauthorizedUsers() {  
    return userDao.findAllUnauthorizedAccounts();  
}
```

```
@Override
```

```
public void setUserFeatures(String username, int featureId) {  
    userDao.setUserFeatureStatus(username,featureId);  
}
```

```
static boolean isNumber(String s)
```

```
{  
    for (int i = 0; i < s.length(); i++)  
        if (Character.isDigit(s.charAt(i))  
            == false)  
            return false;  
  
    return true;  
}
```

```
@Override
```

```
public UserDisplay searchUser(String userDetail) {
```



```
        if(isNumber(userDetail)) {  
            return displayDao.getUserDetailsByAccountNo(Long.parseLong(userDetail));  
        }  
        else {  
            return displayDao.getUserDetailsByUsername(userDetail);  
        }  
    }  
}
```

```
package com.admin.service.impl;
```

```
import org.apache.commons.mail.DefaultAuthenticator;
```

```
import org.apache.commons.mail.EmailException;
```

```
import org.apache.commons.mail.HtmlEmail;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Service;
```

```
import com.admin.dao.AccountRepository;
```

```
import com.admin.dao.UserRepository;
```

```
import com.admin.model.Account;
```

```
import com.admin.model.User;
```

```
@Service
```

```
public class MailServiceImpl {

    @Autowired
    UserRepository userDAO;

    @Autowired
    AccountRepository accountDAO;

    public void sendAuthorizedEmail(String username) throws EmailException {

        User user = userDAO.findByUsername(username);

        String receiver = user.getEmail();

        String fullName = user.getFname()+" "+user.getLname();

        Account account = accountDAO.findByUsername(username);

        long accountNo = account.getAccno();

        int balance = account.getBalance();

        HtmlEmail mail = new HtmlEmail();

        mail.setSSL(true);

        mail.setHostName("smtp.gmail.com");

        mail.setSmtpPort(465);

        mail.setAuthenticator(new DefaultAuthenticator("icinbank.noreply@gmail.com",
"icin@1155"));

        mail.setFrom("icinbank.noreply@gmail.com", "ICIN");

        mail.addTo(receiver, "To");
```

```

mail.setSubject(user.getFname()+" , Welcome to your new ICIN Bank Account");

mail.setHtmlMsg("<h1 style='font-size:27px;color:blue'>Hi "+fullName+",</h1>"
                + "<br><h2 style='font-size:21px;color:black'>Thank you for creating a ICIN Bank
account. Your account has been activated,"
                + " Here are your details.</h2><br>"
                + "<p style='font-size:18px;'>Username: "+username+"<br><p style='font-
size:18px;'>Account Number: "+accountNo+"<br>"
                + "<p style='font-size:18px;'>Current Balance: Rs.
"+balance+"</p>");

mail.send();

System.out.println("email sent");
}

```

```

public void sendAuthorizeCancelEmail(String username) throws EmailException {

    User user = userDAO.findByUsername(username);

    String receiver = user.getEmail();

    String fullName = user.getFname()+" "+user.getLname();

    HtmlEmail mail = new HtmlEmail();

    mail.setSSL(true);

    mail.setHostName("smtp.gmail.com");

    mail.setSmtpPort(465);

    mail.setAuthenticator(new DefaultAuthenticator("icinbank.noreply@gmail.com",
"icin@1155"));

    mail.setFrom("icinbank.noreply@gmail.com", "ICIN");
}

```

```

        mail.addTo(receiver, "To");

        mail.setSubject("ICIN Bank Account Cancelled");

        mail.setHtmlMsg("<h1 style='font-size:27px;color:red'>Hi "+fullName+",</h1>"
            + "<br><p style='font-size:19px;color:black'>Sorry, Your account with
ICIN Bank with username "+username+ " has been cancelled. Please contact nearest branch for more
details.</p><br><br>");

        mail.send();

        System.out.println("email sent");

    }

```

```

public void sendChequebookConfirmedEmail(String username) throws EmailException {

```

```

    User user = userDAO.findByUsername(username);

    String receiver = user.getEmail();

    String fullName = user.getFname()+" "+user.getLname();

    HtmlEmail mail = new HtmlEmail();

    mail.setSSL(true);

    mail.setHostName("smtp.gmail.com");

    mail.setSmtpport(465);

    mail.setAuthenticator(new DefaultAuthenticator("icinbank.noreply@gmail.com",
"icin@1155"));

    mail.setFrom("icinbank.noreply@gmail.com", "ICIN");

    mail.addTo(receiver, "To");

    mail.setSubject("ICIN Bank Chequebook Request Confirmed");

    mail.setHtmlMsg("<h1 style='font-size:25px;color:green'>Hi "+fullName+",</h1>"

```

```
        + "<br><p style='font-size:19px;color:black'>Your chequebook request has been  
processed and confirmed it will be sent to you via courier in 4-5 business days.</p><br><br>");
```

```
        mail.send();
```

```
        System.out.println("email sent");
```

```
    }
```

```
}
```

```
package com.admin.service.impl;
```

```
import java.util.List;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Service;
```

```
import com.admin.dao.SaccountRepository;
```

```
import com.admin.dao.UserRepository;
```

```
import com.admin.model.Saccount;
```

```
@Service
```

```
public class SaccountCreationImpl {
```

```
    @Autowired
```

```
    SaccountRepository dao;
```

```
    @Autowired
```

```

    UserRepository udao;

    private final String bankCode = "3914";

    private final String countryCode = "91";

    private final String branchCode = "820";

    private final String accountcode="2";

    public long generate_saving(int userId) {

        String accNo =
bankCode+countryCode+branchCode+accountcode+String.valueOf(userId);

        return Long.parseLong(accNo);

    }

    public Saccount newAccount(String username, int userId) {

        Saccount account =new Saccount();

        account.setUsername(username);

        account.setAccno(generate_saving(userId));

        account.setUser(udao.findByUsername(username));

        return dao.save(account);

    }

}

```

src/main/rresources

```

spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:mysql://localhost:3306/icinbank

```

spring.datasource.username=root

spring.datasource.password=Host

server.port=8084