import javax.swing.\*;  
import java.awt.\*;  
import java.net.URI;  
  
class SmartCityApp {  
 public static void main(String[] args) {  
 Accounts acc = new Accounts();  
  
 acc.accGui(); // Launch the login GUI  
 }  
}  
//=================================================login + Register page  
class Accounts {  
 private final CategoryClass category = new CategoryClass(); // Link to CategoryClass  
  
 public void accGui() {  
 JFrame frame = new JFrame("Smart City");  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 frame.setLayout(null);  
 JPanel panel1 = new JPanel();  
 panel1.setLayout(null);  
 panel1.setBackground(new Color(0, 102, 102));  
 panel1.setBounds(0, 0, 800, 1080);  
 JLabel imageLabel = new JLabel();  
 imageLabel.setBounds(30, 150, 600, 400); // Centered image in the teal panel  
 ImageIcon icon = new ImageIcon("C:\\Users\\HP 640 G2\\Downloads\\images\\Smart\_city.png"); // Replace with your image path  
 Image img = icon.getImage().getScaledInstance(700, 500, Image.*SCALE\_SMOOTH*); // Scaled image  
 imageLabel.setIcon(new ImageIcon(img));  
 panel1.add(imageLabel);  
 JLabel text = new JLabel("Welcome To Smart City");  
 text.setForeground(Color.*WHITE*);  
 text.setFont(new Font("Serif", Font.*BOLD*, 38));  
 text.setBounds(190, 50, 400, 50);  
 panel1.add(text);  
 JPanel logPanel = new JPanel(null);  
 logPanel.setBackground(Color.*WHITE*);  
 logPanel.setBounds(800, 0, 1200, 1280);  
 int leftMargin = 50;  
 JLabel logLabel1 = new JLabel("Email/Phone Number");  
 JTextField logField1 = new JTextField();  
 JLabel logLabel2 = new JLabel("Password");  
 JPasswordField logField2 = new JPasswordField();  
 JButton logButton1 = new JButton("Log in");  
 JButton registerButton = new JButton("Register");  
 logLabel1.setBounds(leftMargin, 140, 200, 30);  
 logField1.setBounds(leftMargin + 200, 140, 200, 30);  
 logLabel2.setBounds(leftMargin, 210, 200, 30);  
 logField2.setBounds(leftMargin + 200, 210, 200, 30);  
 logButton1.setBounds(leftMargin + 100, 270, 100, 30);  
 registerButton.setBounds(leftMargin + 250, 270, 100, 30);  
 logButton1.setBackground(new Color(0, 102, 102));  
 logButton1.setForeground(Color.*WHITE*);  
 registerButton.setBackground(new Color(0, 102, 102));  
 registerButton.setForeground(Color.*WHITE*);  
 logButton1.addActionListener(e -> {  
 try {  
 String email = logField1.getText();  
 String password = new String(logField2.getPassword());  
  
 // Regex for email validation  
 String emailRegex = "^[A-Za-z0-9+\_.-]+@[A-Za-z0-9.-]+$";  
 // Regex for password validation (minimum 8 characters, at least one letter and one number)  
 String passwordRegex = "^(?=.[A-Za-z])(?=.\\d)[A-Za-z\\d]{8,}$";  
  
 if (email.isEmpty() || password.isEmpty()) {  
 throw new IllegalArgumentException("Please enter both email and password.");  
 }  
 if (!email.matches(emailRegex)) {  
 throw new IllegalArgumentException("Invalid email format. Please enter a valid email address.");  
 }  
 if (!password.matches(passwordRegex)) {  
 throw new IllegalArgumentException(  
 "Invalid password. Password must be at least 8 characters long and include at least one letter and one number."  
 );  
 }  
 JOptionPane.*showMessageDialog*(frame, "Login successful! Welcome to Smart City.");  
 category.catGUI();  
 frame.dispose();  
 } catch (Exception ex) {  
 JOptionPane.*showMessageDialog*(frame, ex.getMessage(), "Input Error", JOptionPane.*ERROR\_MESSAGE*);  
 }  
 });  
 registerButton.addActionListener(e -> {  
 new RegisterPage().registerGUI();  
 frame.dispose();  
 });  
  
 logPanel.add(logLabel1);  
 logPanel.add(logField1);  
 logPanel.add(logLabel2);  
 logPanel.add(logField2);  
 logPanel.add(logButton1);  
 logPanel.add(registerButton);  
  
 frame.add(panel1);  
 frame.add(logPanel);  
  
 frame.setSize(1980, 1080);  
 frame.setVisible(true);  
 }  
  
 public void validateCredentials(String email, String password) {  
 String emailRegex = "^[A-Za-z0-9+\_.-]+@[A-Za-z0-9.-]+$";  
 String passwordRegex = "^(?=.[A-Za-z])(?=.\\d)[A-Za-z\\d]{8,}$";  
  
 if (email.isEmpty() || password.isEmpty()) {  
 throw new IllegalArgumentException("Please enter both email and password.");  
 }  
  
 if (!email.matches(emailRegex)) {  
 throw new IllegalArgumentException("Invalid email format. Please enter a valid email address.");  
 }  
  
 if (!password.matches(passwordRegex)) {  
 throw new IllegalArgumentException(  
 "Invalid password. Password must be at least 8 characters long and include at least one letter and one number."  
 );  
 }  
 }  
  
  
  
 class RegisterPage {  
 public <T extends JComponent> void addField(JPanel panel, JLabel label, T field, int x, int y) {  
 label.setBounds(x, y, 200, 30);  
 field.setBounds(x + 200, y, 200, 30);  
 panel.add(label);  
 panel.add(field);  
 }  
 public <T extends JButton> void addButton(JPanel panel, T button, int x, int y, String text) {  
 button.setText(text);  
 button.setBounds(x, y, 100, 30);  
 button.setBackground(new Color(0, 102, 102));  
 button.setForeground(Color.*WHITE*);  
 panel.add(button);  
 }  
 public void registerGUI() {  
 JFrame frame = new JFrame("Smart City - Register");  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 frame.setLayout(null);  
 JPanel panel1 = new JPanel();  
 panel1.setLayout(null);  
 panel1.setBackground(new Color(0, 102, 102));  
 panel1.setBounds(0, 0, 800, 1080);  
 JLabel imageLabel = new JLabel();  
 imageLabel.setBounds(30, 150, 600, 400); // Centered image in the teal panel  
 ImageIcon icon = new ImageIcon("C:\\Users\\HP 640 G2\\Downloads\\images\\Smart\_city.png"); // Replace with your image path  
 Image img = icon.getImage().getScaledInstance(700, 500, Image.*SCALE\_SMOOTH*); // Scaled image  
 imageLabel.setIcon(new ImageIcon(img));  
 panel1.add(imageLabel);  
 JLabel text = new JLabel("Join Smart City");  
 text.setForeground(Color.*WHITE*);  
 text.setFont(new Font("Serif", Font.*BOLD*, 38));  
 text.setBounds(230, 50, 400, 50);  
 panel1.add(text);  
 JPanel regPanel = new JPanel(null);  
 regPanel.setBackground(Color.*WHITE*);  
 regPanel.setBounds(800, 0, 1180, 1080);  
 int leftMargin = 50;  
 JLabel nameLabel = new JLabel("Name");  
 JTextField nameField = new JTextField();  
 JLabel emailLabel = new JLabel("Email");  
 JTextField emailField = new JTextField();  
 JLabel passwordLabel = new JLabel("Password");  
 JPasswordField passwordField = new JPasswordField();  
 JLabel confirmPasswordLabel = new JLabel("Confirm Password");  
 JPasswordField confirmPasswordField = new JPasswordField();  
 JButton registerButton = new JButton("Register");  
 JButton loginButton = new JButton("Login");  
 nameLabel.setBounds(leftMargin, 100, 200, 30);  
 nameField.setBounds(leftMargin + 200, 100, 200, 30);  
 emailLabel.setBounds(leftMargin, 170, 200, 30);  
 emailField.setBounds(leftMargin + 200, 170, 200, 30);  
 passwordLabel.setBounds(leftMargin, 240, 200, 30);  
 passwordField.setBounds(leftMargin + 200, 240, 200, 30);  
 confirmPasswordLabel.setBounds(leftMargin, 310, 200, 30);  
 confirmPasswordField.setBounds(leftMargin + 200, 310, 200, 30);  
  
  
 registerButton.setBounds(leftMargin + 100, 380, 100, 30);  
 loginButton.setBounds(leftMargin + 250, 380, 100, 30);  
  
 registerButton.setBackground(new Color(0, 102, 102));  
 registerButton.setForeground(Color.*WHITE*);  
 loginButton.setBackground(new Color(0, 102, 102));  
 loginButton.setForeground(Color.*WHITE*);  
  
  
  
 registerButton.addActionListener(e -> {  
 try {  
 String name = nameField.getText();  
 String email = emailField.getText();  
 String password = new String(passwordField.getPassword());  
 String confirmPassword = new String(confirmPasswordField.getPassword()); // Confirm Password field value  
  
 // Regex for email validation  
 String emailRegex = "^[A-Za-z0-9+\_.-]+@[A-Za-z0-9.-]+$";  
 // Regex for password validation (minimum 8 characters, at least one letter and one number)  
 String passwordRegex = "^(?=.[A-Za-z])(?=.\\d)[A-Za-z\\d]{8,}$";  
  
 if (name.isEmpty() || email.isEmpty() || password.isEmpty() || confirmPassword.isEmpty()) {  
 throw new IllegalArgumentException("Please fill in all fields.");  
 }  
  
 if (!email.matches(emailRegex)) {  
 throw new IllegalArgumentException("Invalid email format. Please enter a valid email address.");  
 }  
  
 if (!password.matches(passwordRegex)) {  
 throw new IllegalArgumentException(  
 "Invalid password. Password must be at least 8 characters long and include at least one letter and one number."  
 );  
 }  
  
 if (!password.equals(confirmPassword)) {  
 throw new IllegalArgumentException("Passwords do not match. Please confirm your password.");  
 }  
  
 JOptionPane.*showMessageDialog*(frame, "Registration successful! Please log in.");  
 new Accounts().accGui();  
 frame.dispose();  
 } catch (Exception ex) {  
 JOptionPane.*showMessageDialog*(frame, ex.getMessage(), "Input Error", JOptionPane.*ERROR\_MESSAGE*);  
 }  
 });  
  
  
  
 regPanel.add(nameLabel);  
 regPanel.add(nameField);  
 regPanel.add(emailLabel);  
 regPanel.add(emailField);  
 regPanel.add(passwordLabel);  
 regPanel.add(passwordField);  
 regPanel.add(confirmPasswordLabel);  
 regPanel.add(confirmPasswordField);  
 regPanel.add(registerButton);  
 regPanel.add(loginButton);  
  
 frame.add(panel1);  
 frame.add(regPanel);  
  
 frame.setSize(1980, 1080);  
 frame.setVisible(true);  
 }  
 }  
  
  
 //categoty  
 class CategoryClass {  
 void catGUI() {  
 JFrame frame = new JFrame("Category Selection");  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 frame.setLayout(new BorderLayout());  
 frame.setExtendedState(JFrame.*MAXIMIZED\_BOTH*);  
 JPanel panel = new JPanel();  
 panel.setLayout(null);  
 panel.setBackground(new Color(0, 102, 102)); // Dark theme background  
 JLabel imageLabel = new JLabel();  
 imageLabel.setBounds(50, 100, 600, 400); // Position and size of the image  
 ImageIcon icon = new ImageIcon("C:\\Users\\HP 640 G2\\Downloads\\images\\Profession.png"); // Replace with actual image path  
 Image img = icon.getImage().getScaledInstance(600, 400, Image.*SCALE\_SMOOTH*); // Scale image to fit  
 imageLabel.setIcon(new ImageIcon(img));  
 panel.add(imageLabel);  
  
 //e text for category selection  
 JLabel text = new JLabel("Select Your Profession");  
 text.setForeground(Color.*WHITE*);  
 text.setFont(new Font("Serif", Font.*BOLD*, 50));  
 text.setHorizontalAlignment(SwingConstants.*CENTER*);  
 text.setBounds(590, 30, 800, 60); // Adjusted position for left margin  
 panel.add(text);  
  
 // Category buttons  
 JPanel categoryPanel = new JPanel();  
 categoryPanel.setLayout(new GridLayout(4, 2, 15, 15)); // Grid layout for buttons  
 categoryPanel.setBounds(750, 150, 500, 400);  
 categoryPanel.setBackground(new Color(0, 102, 102)); // Match background with main panel  
  
 String[] categories = { "Student", "Traveler", "Doctor", "Restaurant", "Engineer", "Villas", "Others" };  
 JButton[] categoryButtons = new JButton[categories.length];  
  
 for (int i = 0; i < categories.length; i++) {  
 categoryButtons[i] = new JButton(categories[i]);  
  
 // Button design  
 categoryButtons[i].setFont(new Font("Arial", Font.*BOLD*, 18));  
 categoryButtons[i].setBackground(new Color(0, 102, 102)); // Blue background  
 categoryButtons[i].setForeground(Color.*WHITE*); // White text  
 categoryButtons[i].setBorder(BorderFactory.*createLineBorder*(new Color(102, 230, 230, 232), 2));  
  
 // Add hover effect  
 categoryButtons[i].addMouseListener(new java.awt.event.MouseAdapter() {  
 @Override  
 public void mouseEntered(java.awt.event.MouseEvent evt) {  
 JButton button = (JButton) evt.getSource();  
 button.setBackground(new Color(0, 102, 102)); // Slightly darker blue on hover  
 }  
  
 @Override  
 public void mouseExited(java.awt.event.MouseEvent evt) {  
 JButton button = (JButton) evt.getSource();  
 button.setBackground(new Color(0, 102, 102)); // Original blue color  
 }  
 });  
  
 // Button click action  
 categoryButtons[i].addActionListener(e -> {  
 String selectedCategory = ((JButton) e.getSource()).getText();  
 openCategoryPage(selectedCategory, frame);  
 });  
  
 categoryPanel.add(categoryButtons[i]);  
 }  
  
 panel.add(categoryPanel);  
  
 // Footer with additional options  
 JPanel footerPanel = new JPanel();  
 footerPanel.setBackground(new Color(0, 102, 102));  
 footerPanel.setLayout(new FlowLayout(FlowLayout.*CENTER*, 10, 10));  
  
 JButton backButton = new JButton("Back");  
 JButton logoutButton = new JButton("Log Out");  
 logoutButton.setFont(new Font("Arial", Font.*PLAIN*, 12));  
 logoutButton.setBackground(Color.*WHITE*); // Green background  
 logoutButton.setForeground(new Color(0, 102, 102));  
 logoutButton.setBorder(BorderFactory.*createLineBorder*(new Color(0, 102, 102), 2));  
 logoutButton.addActionListener(e -> {  
 int confirm = JOptionPane.*showConfirmDialog*(frame, "Are you sure you want to log out?", "Log Out", JOptionPane.*YES\_NO\_OPTION*);  
 if (confirm == JOptionPane.*YES\_OPTION*) {  
 frame.dispose();  
 new Accounts().accGui(); // Redirect to login  
 }  
 });  
  
 footerPanel.add(backButton);  
 footerPanel.add(logoutButton);  
 footerPanel.setBounds(0, 800, 1990, 70);  
  
 panel.add(footerPanel);  
  
 frame.add(panel);  
 frame.setVisible(true);  
 }  
 private void openCategoryPage(String category, JFrame parentFrame) {  
 parentFrame.dispose(); // Close the category selection page  
 System.*out*.println("Selected category: " + category); // Debugging line  
 switch (category) {  
 case "Student":  
 new StudentCategory().displayGUI();  
 break;  
 case "Traveler":  
 new TravelerCategory().displayGUI();  
 break;  
 case "Doctor":  
 new DoctorCategory().displayGUI();  
 break;  
 case "Restaurant":  
 new RestaurantCategory().displayGUI();  
 break;  
 case "Engineer":  
 new EngineerCategory().displayGUI();  
 break;  
 case "Villas":  
 new VillasCategory().displayGUI();  
 break;  
 case "Others":  
 new OthersCategory().displayGUI();  
 break;  
 default:  
 JOptionPane.*showMessageDialog*(null, "Category not implemented yet!");  
 break;  
 }  
 }  
 }  
 class StudentCategory {  
 void displayGUI() {  
 JFrame frame = new JFrame("Student - Educational Institutions");  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 JPanel panel = new JPanel();  
 panel.setLayout(new BorderLayout());  
 panel.setBackground(new Color(240, 248, 255));  
  
 // Title Label  
 JLabel label = new JLabel("Student - Educational Institutions", JLabel.*CENTER*);  
 label.setFont(new Font("Serif", Font.*BOLD*, 30));  
 label.setForeground(new Color(0, 102, 102));  
  
 // Info Panel with images  
 JPanel infoPanel = new JPanel(new GridLayout(2, 2, 20, 20));  
 infoPanel.setBackground(new Color(240, 248, 255));  
  
 // Adding individual panels for institutions  
 infoPanel.add(createInstitutionPanel("School: ABC High School", "C:\\Users\\HP 640 G2\\Downloads\\images\\school.jpg",  
 "ABC High School, City Address"));  
 infoPanel.add(createInstitutionPanel("Library: City Library", "C:\\Users\\HP 640 G2\\Downloads\\images\\library.jpg",  
 "City Library, Address"));  
 infoPanel.add(createInstitutionPanel("University: XYZ University", "C:\\Users\\HP 640 G2\\Downloads\\images\\university.jfif",  
 "XYZ University, City Address"));  
 infoPanel.add(createInstitutionPanel("Hostel: Downtown Hostel", "C:\\Users\\HP 640 G2\\Downloads\\images\\hostel.jpg",  
 "Downtown Hostel, City Address"));  
  
 // Back and Logout Buttons  
 JButton backButton = new JButton("Back");  
 backButton.setBackground(new Color(0, 102, 102));  
 backButton.setForeground(Color.*WHITE*);  
 backButton.addActionListener(e -> {  
 new CategoryClass().catGUI();  
 frame.dispose();  
 });  
  
 JButton logoutButton = new JButton("Log Out");  
 logoutButton.setBackground(new Color(0, 102, 102));  
 logoutButton.setForeground(Color.*WHITE*);  
 logoutButton.addActionListener(e -> {  
 int confirm = JOptionPane.*showConfirmDialog*(frame, "Do you want to log out?", "Log Out", JOptionPane.*YES\_NO\_OPTION*);  
 if (confirm == JOptionPane.*YES\_OPTION*) {  
 new Accounts().accGui();  
 frame.dispose();  
 }  
 });  
  
 JPanel buttonPanel = new JPanel();  
 buttonPanel.setBackground(new Color(240, 248, 255));  
 buttonPanel.add(backButton);  
 buttonPanel.add(logoutButton);  
  
 // Adding components to the main panel  
 panel.add(label, BorderLayout.*NORTH*);  
 panel.add(infoPanel, BorderLayout.*CENTER*);  
 panel.add(buttonPanel, BorderLayout.*SOUTH*);  
  
 // Final Frame Setup  
 frame.add(panel);  
 frame.setExtendedState(JFrame.*MAXIMIZED\_BOTH*); // Open in maximized mode  
 frame.setVisible(true);  
 frame.toFront(); // Ensure it comes to the front  
 }  
  
 private JPanel createInstitutionPanel(String name, String imagePath, String destination) {  
 JPanel institutionPanel = new JPanel(new BorderLayout());  
 institutionPanel.setBackground(Color.*WHITE*);  
  
 JLabel nameLabel = new JLabel(name, JLabel.*CENTER*);  
 nameLabel.setFont(new Font("Serif", Font.*PLAIN*, 18));  
 nameLabel.setForeground(Color.*BLACK*);  
  
 ImageIcon icon = new ImageIcon(imagePath);  
 Image img = icon.getImage().getScaledInstance(260, 200, Image.*SCALE\_SMOOTH*);  
 JLabel imageLabel = new JLabel(new ImageIcon(img));  
  
 JButton directionsButton = new JButton("Get Directions");  
 directionsButton.setBackground(new Color(0, 102, 102));  
 directionsButton.setForeground(Color.*WHITE*);  
 directionsButton.addActionListener(e -> openMap(destination));  
  
 institutionPanel.add(imageLabel, BorderLayout.*CENTER*);  
 institutionPanel.add(nameLabel, BorderLayout.*NORTH*);  
 institutionPanel.add(directionsButton, BorderLayout.*SOUTH*);  
  
 return institutionPanel;  
 }  
  
 private void openMap(String destination) {  
 try {  
 String mapUrl = "https://www.google.com/maps/dir/?api=1&destination=" +  
 destination.replace(" ", "+");  
 Desktop.*getDesktop*().browse(new URI(mapUrl));  
 } catch (Exception e) {  
 JOptionPane.*showMessageDialog*(null, "Unable to open map. Please try again!");  
 }  
 }  
 }  
  
 class TravelerCategory {  
 void displayGUI() {  
 JFrame frame = new JFrame("Traveler - Tourist Destinations");  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 JPanel panel = new JPanel();  
 panel.setLayout(new BorderLayout());  
 panel.setBackground(new Color(240, 248, 255)); // Light background  
  
 JLabel label = new JLabel("Traveler - Tourist Destinations", JLabel.*CENTER*);  
 label.setFont(new Font("Serif", Font.*BOLD*, 30));  
 label.setForeground(new Color(0, 102, 102));  
  
 // Info Panel with images  
 JPanel infoPanel = new JPanel(new GridLayout(2, 2, 20, 20));  
 infoPanel.setBackground(new Color(240, 248, 255));  
  
 // Destination Section  
 infoPanel.add(createSectionPanel("Destination: Beachside Resort", "C:\\Users\\HP 640 G2\\Downloads\\images\\beach.jpg", "Beachside Resort"));  
  
 // Transport Section  
 infoPanel.add(createSectionPanel("Transport: City Metro", "C:\\Users\\HP 640 G2\\Downloads\\images\\city metro.jpg", "City Metro Station"));  
  
 // Attraction Section  
 infoPanel.add(createSectionPanel("Attraction: National Museum", "C:\\Users\\HP 640 G2\\Downloads\\images\\museum.jpg", "National Museum"));  
  
 // Hotel Section  
 infoPanel.add(createSectionPanel("Hotel: Paradise Inn", "C:\\Users\\HP 640 G2\\Downloads\\images\\paradise in.jpg", "Paradise Inn"));  
  
 // Back and Logout Buttons  
 JButton backButton = new JButton("Back");  
 backButton.setBackground(new Color(0, 102, 102));  
 backButton.setForeground(Color.*WHITE*);  
 backButton.addActionListener(e -> {  
 new CategoryClass().catGUI();  
 frame.dispose();  
 });  
  
 JButton logoutButton = new JButton("Log Out");  
 logoutButton.setBackground(new Color(0, 102, 102));  
 logoutButton.setForeground(Color.*WHITE*);  
 logoutButton.addActionListener(e -> {  
 int confirm = JOptionPane.*showConfirmDialog*(frame, "Do you want to log out?", "Log Out", JOptionPane.*YES\_NO\_OPTION*);  
 if (confirm == JOptionPane.*YES\_OPTION*) {  
 new Accounts().accGui();  
 frame.dispose();  
 }  
 });  
  
 JPanel buttonPanel = new JPanel();  
 buttonPanel.setBackground(new Color(240, 248, 255));  
 buttonPanel.add(backButton);  
 buttonPanel.add(logoutButton);  
  
 panel.add(label, BorderLayout.*NORTH*);  
 panel.add(infoPanel, BorderLayout.*CENTER*);  
 panel.add(buttonPanel, BorderLayout.*SOUTH*);  
  
 frame.add(panel);  
 frame.setExtendedState(JFrame.*MAXIMIZED\_BOTH*); // Open in maximized mode  
 frame.setVisible(true);  
 frame.toFront(); // Ensure it comes to the front  
 }  
  
 private JPanel createSectionPanel(String title, String imagePath, String location) {  
 JPanel sectionPanel = new JPanel(new BorderLayout());  
 sectionPanel.setBackground(Color.*WHITE*);  
  
 JLabel titleLabel = new JLabel(title, JLabel.*CENTER*);  
 titleLabel.setFont(new Font("Serif", Font.*PLAIN*, 18));  
 titleLabel.setForeground(Color.*BLACK*);  
  
 ImageIcon icon = new ImageIcon(imagePath);  
 Image img = icon.getImage().getScaledInstance(260, 200, Image.*SCALE\_SMOOTH*);  
 JLabel imageLabel = new JLabel(new ImageIcon(img));  
  
 JButton directionButton = new JButton("Get Direction");  
 directionButton.setBackground(new Color(0, 102, 102));  
 directionButton.setForeground(Color.*WHITE*);  
 directionButton.addActionListener(e -> openGoogleMaps(location));  
  
 sectionPanel.add(imageLabel, BorderLayout.*CENTER*);  
 sectionPanel.add(titleLabel, BorderLayout.*NORTH*);  
 sectionPanel.add(directionButton, BorderLayout.*SOUTH*);  
  
 return sectionPanel;  
 }  
  
 private void openGoogleMaps(String location) {  
 try {  
 String url = "https://www.google.com/maps/search/?api=1&query=" + location.replace(" ", "+");  
 Desktop.*getDesktop*().browse(new URI(url));  
 } catch (Exception ex) {  
 JOptionPane.*showMessageDialog*(null, "Failed to open Google Maps. Please check your connection.");  
 }  
 }  
 }  
 class DoctorCategory {  
 void displayGUI() {  
 JFrame frame = new JFrame("Doctor - Medical Facilities");  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 JPanel panel = new JPanel();  
 panel.setLayout(new BorderLayout());  
 panel.setBackground(new Color(245, 245, 245)); // Light gray background  
  
 JLabel label = new JLabel("Doctor - Medical Facilities", JLabel.*CENTER*);  
 label.setFont(new Font("Serif", Font.*BOLD*, 30));  
 label.setForeground(new Color(0, 102, 102)); // Dark red for healthcare theme  
  
 // Info Panel with images  
 JPanel infoPanel = new JPanel(new GridLayout(2, 2, 20, 20));  
 infoPanel.setBackground(new Color(245, 245, 245));  
  
 // Adding individual panels for medical facilities  
 infoPanel.add(createFacilityPanel("Hospital: City Care Hospital", "C:\\Users\\HP 640 G2\\Downloads\\images\\city care.jpg", "City Care Hospital"));  
 infoPanel.add(createFacilityPanel("Clinic: Downtown Clinic", "C:\\Users\\HP 640 G2\\Downloads\\images\\down town.jpg", "Downtown Clinic"));  
 infoPanel.add(createFacilityPanel("Pharmacy: Health Mart", "C:\\Users\\HP 640 G2\\Downloads\\images\\healthmart1.jpg", "Health Mart"));  
 infoPanel.add(createFacilityPanel("Rehabilitation: Recovery Center", "C:\\Users\\HP 640 G2\\Downloads\\images\\recovery center.jpg", "Recovery Center"));  
  
 // Back and Logout Buttons  
 JButton backButton = new JButton("Back");  
 backButton.setBackground(new Color(0, 102, 102));  
 backButton.setForeground(Color.*WHITE*);  
 backButton.addActionListener(e -> {  
 new CategoryClass().catGUI();  
 frame.dispose();  
 });  
  
 JButton logoutButton = new JButton("Log Out");  
 logoutButton.setBackground(new Color(0, 102, 102));  
 logoutButton.setForeground(Color.*WHITE*);  
 logoutButton.addActionListener(e -> {  
 int confirm = JOptionPane.*showConfirmDialog*(frame, "Do you want to log out?", "Log Out", JOptionPane.*YES\_NO\_OPTION*);  
 if (confirm == JOptionPane.*YES\_OPTION*) {  
 new Accounts().accGui();  
 frame.dispose();  
 }  
 });  
  
 JPanel buttonPanel = new JPanel();  
 buttonPanel.setBackground(new Color(245, 245, 245));  
 buttonPanel.add(backButton);  
 buttonPanel.add(logoutButton);  
  
 panel.add(label, BorderLayout.*NORTH*);  
 panel.add(infoPanel, BorderLayout.*CENTER*);  
 panel.add(buttonPanel, BorderLayout.*SOUTH*);  
  
 frame.add(panel);  
 frame.setExtendedState(JFrame.*MAXIMIZED\_BOTH*); // Open in maximized mode  
 frame.setVisible(true);  
 frame.toFront(); // Ensure it comes to the front  
 }  
  
 private JPanel createFacilityPanel(String title, String imagePath, String location) {  
 JPanel facilityPanel = new JPanel(new BorderLayout());  
 facilityPanel.setBackground(Color.*WHITE*);  
  
 JLabel titleLabel = new JLabel(title, JLabel.*CENTER*);  
 titleLabel.setFont(new Font("Serif", Font.*PLAIN*, 18));  
 titleLabel.setForeground(Color.*BLACK*);  
  
 ImageIcon icon = new ImageIcon(imagePath); // Replace with the actual path  
 Image img = icon.getImage().getScaledInstance(260, 200, Image.*SCALE\_SMOOTH*);  
 JLabel imageLabel = new JLabel(new ImageIcon(img));  
  
 JButton mapButton = new JButton("View on Map");  
 mapButton.setBackground(new Color(0, 102, 102));  
 mapButton.setForeground(Color.*WHITE*);  
 mapButton.addActionListener(e -> openGoogleMaps(location));  
  
 facilityPanel.add(imageLabel, BorderLayout.*CENTER*);  
 facilityPanel.add(titleLabel, BorderLayout.*NORTH*);  
 facilityPanel.add(mapButton, BorderLayout.*SOUTH*);  
  
 return facilityPanel;  
 }  
  
 private void openGoogleMaps(String location) {  
 try {  
 String url = "https://www.google.com/maps/search/?api=1&query=" + location.replace(" ", "+");  
 Desktop.*getDesktop*().browse(new URI(url));  
 } catch (Exception ex) {  
 JOptionPane.*showMessageDialog*(null, "Failed to open Google Maps. Please check your connection.");  
 }  
 }  
 }  
 class RestaurantCategory {  
 void displayGUI() {  
 JFrame frame = new JFrame("Restaurant - Culinary Experiences");  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 JPanel panel = new JPanel(new BorderLayout());  
 panel.setBackground(new Color(230, 230, 250)); // Light pink background  
  
 JLabel label = new JLabel("Restaurant - Culinary Experiences", JLabel.*CENTER*);  
 label.setFont(new Font("Serif", Font.*BOLD*, 30));  
 label.setForeground(new Color(0, 102, 102)); // Tomato red for food theme  
  
 // Info Panel with images  
 JPanel infoPanel = new JPanel(new GridLayout(2, 2, 20, 20));  
 infoPanel.setBackground(new Color(230, 230, 250));  
  
  
 // Adding sections for different types of restaurants  
 infoPanel.add(createSectionPanel("Fine Dining", "C:\\Users\\HP 640 G2\\Downloads\\images\\fine dining.jpg", "Fine Dining Restaurant"));  
 infoPanel.add(createSectionPanel("Café", "C:\\Users\\HP 640 G2\\Downloads\\images\\cafee.jpg", "Nearby Café"));  
 infoPanel.add(createSectionPanel("Fast Food", "C:\\Users\\HP 640 G2\\Downloads\\images\\fastfood.jpg", "Fast Food Restaurant"));  
 infoPanel.add(createSectionPanel("Desserts", "C:\\Users\\HP 640 G2\\Downloads\\images\\dessert.jpg", "Dessert Shop"));  
  
 // Back and Logout Buttons  
 JButton backButton = new JButton("Back");  
 backButton.setBackground(new Color(0, 102, 102));  
 backButton.setForeground(Color.*WHITE*);  
 backButton.addActionListener(e -> {  
 new CategoryClass().catGUI();  
 frame.dispose();  
 });  
  
 JButton logoutButton = new JButton("Log Out");  
 logoutButton.setBackground(new Color(0, 102, 102));  
 logoutButton.setForeground(Color.*WHITE*);  
 logoutButton.addActionListener(e -> {  
 int confirm = JOptionPane.*showConfirmDialog*(frame, "Do you want to log out?", "Log Out", JOptionPane.*YES\_NO\_OPTION*);  
 if (confirm == JOptionPane.*YES\_OPTION*) {  
 new Accounts().accGui();  
 frame.dispose();  
 }  
 });  
  
 JPanel buttonPanel = new JPanel();  
 buttonPanel.setBackground(new Color(230, 230, 250));  
 buttonPanel.add(backButton);  
 buttonPanel.add(logoutButton);  
  
 panel.add(label, BorderLayout.*NORTH*);  
 panel.add(infoPanel, BorderLayout.*CENTER*);  
 panel.add(buttonPanel, BorderLayout.*SOUTH*);  
  
 frame.add(panel);  
 frame.setExtendedState(JFrame.*MAXIMIZED\_BOTH*); // Open in maximized mode  
 frame.setVisible(true);  
 frame.toFront(); // Ensure it comes to the front  
 }  
  
 private JPanel createSectionPanel(String title, String imagePath, String locationQuery) {  
 JLabel titleLabel = new JLabel(title, JLabel.*CENTER*);  
 titleLabel.setFont(new Font("Serif", Font.*PLAIN*, 18));  
 ImageIcon icon = new ImageIcon(imagePath); // Replace with actual image path  
 Image scaledImage = icon.getImage().getScaledInstance(260, 200, Image.*SCALE\_SMOOTH*);  
 JLabel imageLabel = new JLabel(new ImageIcon(scaledImage));  
  
 JButton mapButton = new JButton("Get Direction");  
 mapButton.setBackground(new Color(0, 102, 102));  
 mapButton.setForeground(Color.*WHITE*);  
 mapButton.addActionListener(e -> openGoogleMaps(locationQuery));  
  
 JPanel sectionPanel = new JPanel(new BorderLayout());  
 sectionPanel.setBackground(Color.*WHITE*);  
 sectionPanel.add(imageLabel, BorderLayout.*CENTER*);  
 sectionPanel.add(titleLabel, BorderLayout.*NORTH*);  
 sectionPanel.add(mapButton, BorderLayout.*SOUTH*);  
  
 return sectionPanel;  
 }  
  
 private void openGoogleMaps(String query) {  
 try {  
 String url = "https://www.google.com/maps/search/?api=1&query=" + query.replace(" ", "+");  
 Desktop.*getDesktop*().browse(new URI(url));  
 } catch (Exception e) {  
 JOptionPane.*showMessageDialog*(null, "Unable to open Google Maps. Please check your connection.");  
 }  
 }  
 }  
 class EngineerCategory {  
 void displayGUI() {  
 JFrame frame = new JFrame("Engineer - Professional Services");  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 JPanel panel = new JPanel(new BorderLayout());  
 panel.setBackground(new Color(224, 255, 255)); // Light cyan background  
  
 JLabel label = new JLabel("Engineer - Professional Services", JLabel.*CENTER*);  
 label.setFont(new Font("Serif", Font.*BOLD*, 30));  
 label.setForeground(new Color(0, 102, 102)); // Dark teal for a technical theme  
  
 // Info Panel with images  
 JPanel infoPanel = new JPanel(new GridLayout(2, 2, 20, 20));  
 infoPanel.setBackground(new Color(224, 255, 255));  
  
 // Adding sections for different types of engineers  
 infoPanel.add(createSectionPanel("Software Engineer", "C:\\Users\\HP 640 G2\\Downloads\\images\\software.jpg", "Software Engineers"));  
 infoPanel.add(createSectionPanel("Civil Engineer", "C:\\Users\\HP 640 G2\\Downloads\\images\\cicil.jpg", "Civil Engineering Firms"));  
 infoPanel.add(createSectionPanel("Mechanical Engineer", "C:\\Users\\HP 640 G2\\Downloads\\images\\mechanical.jpg", "Mechanical Engineering Services"));  
 infoPanel.add(createSectionPanel("Electrical Engineer", "C:\\Users\\HP 640 G2\\Downloads\\images\\electrical.jpg", "Electrical Engineering Services"));  
  
 // Back and Logout Buttons  
 JButton backButton = new JButton("Back");  
 backButton.setBackground(new Color(0, 102, 102));  
 backButton.setForeground(Color.*WHITE*);  
 backButton.addActionListener(e -> {  
 new CategoryClass().catGUI();  
 frame.dispose();  
 });  
  
 JButton logoutButton = new JButton("Log Out");  
 logoutButton.setBackground(new Color(0, 102, 102));  
 logoutButton.setForeground(Color.*WHITE*);  
 logoutButton.addActionListener(e -> {  
 int confirm = JOptionPane.*showConfirmDialog*(frame, "Do you want to log out?", "Log Out", JOptionPane.*YES\_NO\_OPTION*);  
 if (confirm == JOptionPane.*YES\_OPTION*) {  
 new Accounts().accGui();  
 frame.dispose();  
 }  
 });  
  
 JPanel buttonPanel = new JPanel();  
 buttonPanel.setBackground(new Color(224, 255, 255));  
 buttonPanel.add(backButton);  
 buttonPanel.add(logoutButton);  
  
 panel.add(label, BorderLayout.*NORTH*);  
 panel.add(infoPanel, BorderLayout.*CENTER*);  
 panel.add(buttonPanel, BorderLayout.*SOUTH*);  
  
 frame.add(panel);  
 frame.setExtendedState(JFrame.*MAXIMIZED\_BOTH*); // Open in maximized mode  
 frame.setVisible(true);  
 frame.toFront(); // Ensure it comes to the front  
 }  
  
 private JPanel createSectionPanel(String title, String imagePath, String locationQuery) {  
 JLabel titleLabel = new JLabel(title, JLabel.*CENTER*);  
 titleLabel.setFont(new Font("Serif", Font.*PLAIN*, 18));  
 ImageIcon icon = new ImageIcon(imagePath); // Replace with actual image path  
 Image scaledImage = icon.getImage().getScaledInstance(260, 200, Image.*SCALE\_SMOOTH*);  
 JLabel imageLabel = new JLabel(new ImageIcon(scaledImage));  
  
 JButton mapButton = new JButton("Get Direction");  
 mapButton.setBackground(new Color(0, 102, 102));  
 mapButton.setForeground(Color.*WHITE*);  
 mapButton.addActionListener(e -> openGoogleMaps(locationQuery));  
  
 JPanel sectionPanel = new JPanel(new BorderLayout());  
 sectionPanel.setBackground(Color.*WHITE*);  
 sectionPanel.add(imageLabel, BorderLayout.*CENTER*);  
 sectionPanel.add(titleLabel, BorderLayout.*NORTH*);  
 sectionPanel.add(mapButton, BorderLayout.*SOUTH*);  
  
 return sectionPanel;  
 }  
  
 private void openGoogleMaps(String query) {  
 try {  
 String url = "https://www.google.com/maps/search/?api=1&query=" + query.replace(" ", "+");  
 Desktop.*getDesktop*().browse(new URI(url));  
 } catch (Exception e) {  
 JOptionPane.*showMessageDialog*(null, "Unable to open Google Maps. Please check your connection.");  
 }  
 }  
 }  
  
 class VillasCategory {  
 void displayGUI() {  
 JFrame frame = new JFrame("Villas - Luxurious Stays");  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 JPanel panel = new JPanel(new BorderLayout());  
 panel.setBackground(new Color(230, 230, 250)); // Light khaki background  
  
 JLabel label = new JLabel("Villas - Luxurious Stays", JLabel.*CENTER*);  
 label.setFont(new Font("Serif", Font.*BOLD*, 30));  
 label.setForeground(new Color(0, 102, 102)); // Brown color for rustic theme  
  
 // Info Panel with images  
 JPanel infoPanel = new JPanel(new GridLayout(2, 2, 20, 20));  
 infoPanel.setBackground(new Color(230, 230, 250));  
  
  
 // Adding sections for different types of villas  
 infoPanel.add(createSectionPanel("Oceanfront Villas", "C:\\Users\\HP 640 G2\\Downloads\\images\\ocean viila.jpg", "Oceanfront Villas"));  
 infoPanel.add(createSectionPanel("Mountain View Villas", "C:\\Users\\HP 640 G2\\Downloads\\images\\mountain villa.jpg", "Mountain View Villas"));  
 infoPanel.add(createSectionPanel("Poolside Villas", "C:\\Users\\HP 640 G2\\Downloads\\images\\pool villa.jpg", "Poolside Villas"));  
 infoPanel.add(createSectionPanel("Luxury Villas", "C:\\Users\\HP 640 G2\\Downloads\\images\\luxuary villa.png", "Luxury Villas"));  
  
 // Back and Logout Buttons  
 JButton backButton = new JButton("Back");  
 backButton.setBackground(new Color(0, 102, 102));  
 backButton.setForeground(Color.*WHITE*);  
 backButton.addActionListener(e -> {  
 new CategoryClass().catGUI();  
 frame.dispose();  
 });  
  
 JButton logoutButton = new JButton("Log Out");  
 logoutButton.setBackground(new Color(0, 102, 102));  
 logoutButton.setForeground(Color.*WHITE*);  
 logoutButton.addActionListener(e -> {  
 int confirm = JOptionPane.*showConfirmDialog*(frame, "Do you want to log out?", "Log Out", JOptionPane.*YES\_NO\_OPTION*);  
 if (confirm == JOptionPane.*YES\_OPTION*) {  
 new Accounts().accGui();  
 frame.dispose();  
 }  
 });  
  
 JPanel buttonPanel = new JPanel();  
 buttonPanel.setBackground(new Color(230, 230, 250));  
 buttonPanel.add(backButton);  
 buttonPanel.add(logoutButton);  
  
 panel.add(label, BorderLayout.*NORTH*);  
 panel.add(infoPanel, BorderLayout.*CENTER*);  
 panel.add(buttonPanel, BorderLayout.*SOUTH*);  
  
 frame.add(panel);  
 frame.setExtendedState(JFrame.*MAXIMIZED\_BOTH*); // Open in maximized mode  
 frame.setVisible(true);  
 frame.toFront(); // Ensure it comes to the front  
 }  
  
 private JPanel createSectionPanel(String title, String imagePath, String locationQuery) {  
 JLabel titleLabel = new JLabel(title, JLabel.*CENTER*);  
 titleLabel.setFont(new Font("Serif", Font.*PLAIN*, 18));  
 ImageIcon icon = new ImageIcon(imagePath); // Replace with actual image path  
 Image scaledImage = icon.getImage().getScaledInstance(260, 200, Image.*SCALE\_SMOOTH*);  
 JLabel imageLabel = new JLabel(new ImageIcon(scaledImage));  
  
 JButton mapButton = new JButton("Get Direction");  
 mapButton.setBackground(new Color(0, 102, 102));  
 mapButton.setForeground(Color.*WHITE*);  
 mapButton.addActionListener(e -> openGoogleMaps(locationQuery));  
  
 JPanel sectionPanel = new JPanel(new BorderLayout());  
 sectionPanel.setBackground(Color.*WHITE*);  
 sectionPanel.add(imageLabel, BorderLayout.*CENTER*);  
 sectionPanel.add(titleLabel, BorderLayout.*NORTH*);  
 sectionPanel.add(mapButton, BorderLayout.*SOUTH*);  
  
 return sectionPanel;  
 }  
  
 private void openGoogleMaps(String query) {  
 try {  
 String url = "https://www.google.com/maps/search/?api=1&query=" + query.replace(" ", "+");  
 Desktop.*getDesktop*().browse(new URI(url));  
 } catch (Exception e) {  
 JOptionPane.*showMessageDialog*(null, "Unable to open Google Maps. Please check your connection.");  
 }  
 }  
 }  
  
 class OthersCategory {  
 void displayGUI() {  
 JFrame frame = new JFrame("Others - Miscellaneous Categories");  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 JPanel panel = new JPanel(new BorderLayout());  
 panel.setBackground(new Color(230, 230, 250)); // Lavender background  
  
 JLabel label = new JLabel("Others - Miscellaneous Categories", JLabel.*CENTER*);  
 label.setFont(new Font("Serif", Font.*BOLD*, 30));  
 label.setForeground(new Color(0, 102, 102)); // Purple for miscellaneous theme  
  
 // Info Panel with images  
 JPanel infoPanel = new JPanel(new GridLayout(2, 2, 20, 20));  
 infoPanel.setBackground(new Color(230, 230, 250));  
  
  
 // Add sections dynamically  
 infoPanel.add(createSectionPanel("Parks", "C:\\Users\\HP 640 G2\\Downloads\\images\\parks.jpg", "Parks Nearby"));  
 infoPanel.add(createSectionPanel("Museums", "C:\\Users\\HP 640 G2\\Downloads\\images\\museum dinosaurs.jpg", "Museums Nearby"));  
 infoPanel.add(createSectionPanel("Shopping Malls", "C:\\Users\\HP 640 G2\\Downloads\\images\\mall.jpg", "Shopping Malls Nearby"));  
 infoPanel.add(createSectionPanel("Theaters", "C:\\Users\\HP 640 G2\\Downloads\\images\\threats.jpg", "Theaters Nearby"));  
  
 // Back and Logout Buttons  
 JButton backButton = new JButton("Back");  
 backButton.setBackground(new Color(0, 102, 102));  
 backButton.setForeground(Color.*WHITE*);  
 backButton.addActionListener(e -> {  
 new CategoryClass().catGUI();  
 frame.dispose();  
 });  
  
 JButton logoutButton = new JButton("Log Out");  
 logoutButton.setBackground(new Color(0, 102, 102));  
 logoutButton.setForeground(Color.*WHITE*);  
 logoutButton.addActionListener(e -> {  
 int confirm = JOptionPane.*showConfirmDialog*(frame, "Do you want to log out?", "Log Out", JOptionPane.*YES\_NO\_OPTION*);  
 if (confirm == JOptionPane.*YES\_OPTION*) {  
 new Accounts().accGui();  
 frame.dispose();  
 }  
 });  
  
 JPanel buttonPanel = new JPanel();  
 buttonPanel.setBackground(new Color(230, 230, 250));  
 buttonPanel.add(backButton);  
 buttonPanel.add(logoutButton);  
  
 panel.add(label, BorderLayout.*NORTH*);  
 panel.add(infoPanel, BorderLayout.*CENTER*);  
 panel.add(buttonPanel, BorderLayout.*SOUTH*);  
  
 frame.add(panel);  
 frame.setExtendedState(JFrame.*MAXIMIZED\_BOTH*); // Open in maximized mode  
 frame.setVisible(true);  
 frame.toFront(); // Ensure it comes to the front  
 }  
  
 private JPanel createSectionPanel(String title, String imagePath, String locationQuery) {  
 JLabel titleLabel = new JLabel(title, JLabel.*CENTER*);  
 titleLabel.setFont(new Font("Serif", Font.*PLAIN*, 18));  
 ImageIcon icon = new ImageIcon(imagePath); // Replace with actual image path  
 Image scaledImage = icon.getImage().getScaledInstance(260, 200, Image.*SCALE\_SMOOTH*);  
 JLabel imageLabel = new JLabel(new ImageIcon(scaledImage));  
  
 JButton mapButton = new JButton("Get Direction");  
 mapButton.setBackground(new Color(0, 102, 102));  
 mapButton.setForeground(Color.*WHITE*);  
 mapButton.addActionListener(e -> openGoogleMaps(locationQuery));  
  
 JPanel sectionPanel = new JPanel(new BorderLayout());  
 sectionPanel.setBackground(Color.*WHITE*);  
 sectionPanel.add(imageLabel, BorderLayout.*CENTER*);  
 sectionPanel.add(titleLabel, BorderLayout.*NORTH*);  
 sectionPanel.add(mapButton, BorderLayout.*SOUTH*);  
  
 return sectionPanel;  
 }  
  
 private void openGoogleMaps(String query) {  
 try {  
 String url = "https://www.google.com/maps/search/?api=1&query=" + query.replace(" ", "+");  
 Desktop.*getDesktop*().browse(new URI(url));  
 } catch (Exception e) {  
 JOptionPane.*showMessageDialog*(null, "Unable to open Google Maps. Please check your connection.");  
 }  
 }  
 }}  
  
import org.junit.Test;  
import static org.junit.Assert.\*;  
  
public class AccountsTest {  
  
 private final Accounts accounts = new Accounts();  
  
 @Test  
 public void testValidateCredentials\_ValidInput() {  
 // Test with valid email and password  
 String email = "test@example.com";  
 String password = "Password1";  
 try {  
 accounts.validateCredentials(email, password);  
 } catch (IllegalArgumentException e) {  
 fail("Validation failed for valid input: " + e.getMessage());  
 }  
 }  
  
 @Test(expected = IllegalArgumentException.class)  
 public void testValidateCredentials\_EmptyEmail() {  
 // Test with empty email  
 accounts.validateCredentials("", "Password1");  
 }  
  
 @Test(expected = IllegalArgumentException.class)  
 public void testValidateCredentials\_InvalidEmailFormat() {  
 // Test with invalid email format  
 accounts.validateCredentials("invalid-email", "Password1");  
 }  
  
 @Test(expected = IllegalArgumentException.class)  
 public void testValidateCredentials\_EmptyPassword() {  
 // Test with empty password  
 accounts.validateCredentials("test@example.com", "");  
 }  
  
 @Test(expected = IllegalArgumentException.class)  
 public void testValidateCredentials\_InvalidPassword() {  
 // Test with password that doesn't meet criteria  
 accounts.validateCredentials("test@example.com", "short");  
 }  
  
 @Test(expected = IllegalArgumentException.class)  
 public void testValidateCredentials\_PasswordWithoutNumber() {  
 // Test with password that doesn't contain a number  
 accounts.validateCredentials("test@example.com", "Password");  
 }  
}