Controller层部分代码：

Login层

@PostMapping("/login")  
 @ResponseBody  
 public ResponseEntity<Map<String, Object>> logIn(@RequestParam("sid") Integer sid, @RequestParam("spassword") String spassword)  
 //对比传入的学生信息，若学号错误或密码不正确则返回false  
 {  
 Map<String, Object> response = new HashMap<>();  
 Student student = studentService.getById(sid);  
 if (student == null || !student.getSpassword().equals(spassword)) {  
 response.put("success", false);  
 response.put("message", "账号或密码错误");  
 return ResponseEntity.*status*(401).body(response);  
 }  
 response.put("semail", student.getSemail());  
 response.put("success", true);  
 response.put("message", "成功登陆");  
 return ResponseEntity.*ok*(response);  
 }  
  
 @GetMapping("/registerPage")  
 public String showRegisterPage()  
 {  
 return "registerPage";  
 }  
  
 @PostMapping("/register")  
 @ResponseBody  
 public ResponseEntity<Map<String, Object>> judgeId(@RequestParam(value = "sid") Integer sid,  
 @RequestParam(value = "spassword1",required = false) String spassword1,  
 @RequestParam(value = "spassword2",required = false) String spassword2,  
 @RequestParam(value = "sphone" ,required = false) String sphone,  
 @RequestParam(value = "semail" ,required = false) String semail)  
 //注册界面，输入学号后判断账号是否已被注册或不存在  
 {  
 Map<String, Object> response = new HashMap<>();  
 Student student = studentService.getById(sid);  
 if(student == null)  
 {  
 response.put("success", false);  
 response.put("message", "该学号不存在！请重新输入");  
 return ResponseEntity.*status*(401).body(response);  
 }  
 else if(student.getRegisterAccount() != 0)  
 {  
 response.put("success", false);  
 response.put("message", "该账号已被注册！请重新输入");  
 return ResponseEntity.*status*(401).body(response);  
 }  
 else  
 {  
 student.setSid(sid);  
 student.setSpassword(spassword1);  
 student.setSphone(sphone);  
 student.setSemail(semail);  
 student.setRegisterAccount((byte)1);  
 studentService.updateById(student);  
 response.put("success", true);  
 return ResponseEntity.*ok*(response);  
 }  
 }  
  
 @GetMapping("/emailVerificationPage")  
 public String showEmailVerificationPage() {  
 return "emailVerificationPage";  
 }  
  
 @PostMapping("/emailVerification")  
 public ResponseEntity<Map<String, Object>> emailVerification(@RequestParam Integer sid) {  
 Map<String, Object> response = new HashMap<>();  
 Student student = studentService.getById(sid);  
 if(student==null)  
 {  
 response.put("success", false);  
 response.put("message", "学号不存在！请重新输入");  
 return ResponseEntity.*status*(401).body(response);  
 }  
 String semail = student.getSemail();  
 String code = mailService.generateVerificationCode();  
 mailService.sendVerificationCode(semail, code);  
 response.put("semail", semail);  
 response.put("success", true);  
 response.put("message", "验证码发送成功");  
 response.put("verification", code);  
 return ResponseEntity.*ok*(response);  
 }  
  
 @GetMapping("/setPasswordPage")  
 public String showSetPasswordPage() {  
 return "setPasswordPage";  
 }  
  
 @PostMapping("/setPassword")  
 public ResponseEntity<Map<String, Object>> setNewPassword(@RequestParam("sid") String sid, //定位学生  
 @RequestParam("spassword1") String spassword1,  
 @RequestParam("spassword2") String spassword2)  
 {  
 Map<String, Object> response = new HashMap<>();  
 Student student = studentService.getById(sid);  
 student.setSpassword(spassword1);  
 studentService.updateById(student);  
 response.put("success", true);  
 response.put("message", "密码修改成功!");  
 return ResponseEntity.*ok*(response);  
 }  
  
 @GetMapping("/authenticationPage")  
 public String showAuthenticationPage() {  
 return "authenticationPage";  
 }  
  
 @PostMapping("/authentication")  
 public ResponseEntity<Map<String, Object>> Authentication(@RequestParam String semail) { //邮箱二次验证  
 Map<String, Object> response = new HashMap<>();  
 String code = mailService.generateVerificationCode();  
 mailService.sendVerificationCode(semail, code);  
 response.put("success", true);  
 response.put("message", "验证码发送成功");  
 response.put("verification", code);  
 return ResponseEntity.*ok*(response);  
 }  
}

Student实体增加了一个属性判断账号是否已被注册：

private Byte registerAccount; //是否已注册账户