|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lab 3 | | | Detailed Design of Your System | | | | | |
| Issue Date | | 2024-12-08 | | | experimental types | | □validation experiment,  □comprehensive experiment  ☑design experiment | |
| Goal   1. Designing the modules of your system by using the detailed design models such as flow chart, PAD, N-S, etc.; 2. Using the decision table or decision tree to model the possible special business logic of algorithms. | | | | | | | | |
| * experimental contents and process   **1. Division of labor**  Our division of labor is as follows.  **Yutao Sun** is responsible for pseudocode design of backend database structure, program functional modules, and logical relationships (Including login function module, news viewing function module, comment posting function module, and administrator management of users, comments, and news function module), and ultimately writing code to implement basic functional modules as well as test the final program.  **Robin Zhang**  is responsible for front-end interface design and code implementation, including integrating all functional modules into complete code. Simultaneously responsible for checking the database logic and drawing structural diagrams.  After completing the above work, we will have a meeting to discuss whether everyone's work content is done correctly and whether the structure diagram of each module is correct.  **2.** **Refined pseudocode for modules**  2.1 Login module (Yutao Sun)  def confirm\_login(username,password,login\_window):  global current\_user # 引用全局变量 current\_user    if username == "" or password == "":  messagebox.showerror("Error", "Please enter both username and password.")  else:  try:  with connection.cursor() as cursor:  # 执行查询  query = "SELECT \* FROM user/admin WHERE userEmail = %s AND userPassword = %s"  cursor.execute(query, (username, password))  result = cursor.fetchone()  if result:  current\_user = result['userID'] # 将用户ID保存到全局变量 current\_user  messagebox.showinfo("Success", "Login successful!")  login\_window.destroy() # 关闭登录窗口  open\_category\_window()  else:  messagebox.showerror("Error", "Invalid username or password.")  except pymysql.Error as e:  messagebox.showerror("Error", f"Database error: {e}")  2.2 Create user module (Yutao Sun)  def create\_user():  def save\_user():  username = entry\_username.get()  password = entry\_password.get()  email = entry\_email.get()  gender = gender\_var.get()  password\_confirm=confirm\_entry.get()    # 检查两次输入的密码是否一致  if password != password\_confirm:  messagebox.showerror("错误", "两次输入的密码不一致")  return  try:  with connection.cursor() as cursor:    cursor.execute("SELECT MAX(userID) AS max\_id FROM user")  result = cursor.fetchone()  Max\_id=result['max\_id'] if result['max\_id'] else 0  max\_user\_id=Max\_id+1  # 插入用户数据到user表中  cursor.execute("INSERT INTO user (userID, userName, userPassword, sex, userEmail) VALUES (%s, %s, %s, %s, %s)", (max\_user\_id,username, password, gender, email))  connection.commit()  messagebox.showinfo("Success", "User created successfully!")  user\_window.destroy()  except pymysql.Error as e:  messagebox.showerror("Error", f"Database error: {e}")  Create\_windows()  Input\_user\_information()  2.3 See news category module (Yutao Sun)  def open\_category\_window():  try:  with connection.cursor() as cursor:  # 查询category表中的数据  cursor.execute("SELECT categoryID, categoryName FROM category")  categories = cursor.fetchall()  # 创建窗口  category\_window = tk.Tk()  category\_window.title("Categories")  category\_window.geometry("400x600")  # background\_label = tk.Label(category\_window, image=global\_photo)  # background\_label.image =global\_photo  # background\_label.place(x=0, y=0, relwidth=1, relheight=1)    #显示提示  category\_label = tk.Label(category\_window, text="请选择您想浏览的新闻类型！", font=("Arial", 16))  category\_label.pack(pady=10)  # 创建按钮  for category in categories:  category\_button = tk.Button(category\_window, text=f"{category['categoryID']}: {category['categoryName']}",command=lambda category\_id=category['categoryID']: open\_news\_window(category\_id))  category\_button.pack(fill=tk.X, padx=20, pady=10)  # 运行窗口的主循环  category\_window.mainloop()  except pymysql.Error as e:  print(f"Database error: {e}")  2.4 Choose news module (Yutao Sun)  # 查询news表中与给定category\_id相符的新闻  cursor.execute("SELECT newsID, newsTitle, newDesc, newsDate, newsTop FROM news WHERE category = %s AND newsIsCheck=1 AND newsTop=0 ORDER BY newsTop DESC" , (category\_id,))  # 创建新闻窗口  news\_window = tk.Toplevel()  Show\_news\_list()  2.5 View news detail module (Yutao Sun)  # 查询新闻详情  cursor.execute("SELECT \* FROM news WHERE newsID = %s", (news\_id,))  news\_detail = cursor.fetchone()  # 查询当前新闻下的所有评论  cursor.execute("SELECT comment.commentID, comment.commentTitle, comment.commentContent, comment.commentDate, user.userName FROM comment JOIN users\_comments ON comment.commentID = users\_comments.the\_CommentID JOIN user ON users\_comments.userID = user.userID WHERE comment.commentID IN (SELECT commentID FROM news\_comment WHERE newsID = %s)", (news\_id,))  comments = cursor.fetchall()  # 创建新闻详情窗口  detail\_window = tk.Toplevel()  # 创建“我也来发布评论”的按钮  comment\_button = tk.Button(detail\_frame, text="我也来发布评论"    2.6 Make comment module (Yutao Sun)  #要求用户输入评论内容  # 创建评论窗口  comment\_window = tk.Toplevel()  comment\_window.title("发表评论")  # 标题输入框  title\_entry = tk.Entry(comment\_window, width=50)  # 内容输入框  content\_label = tk.Label(comment\_window, text="内容:")  content\_label.grid(row=1, column=0, padx=10, pady=5, sticky=tk.W)  content\_entry = tk.Text(comment\_window, width=50, height=10)  content\_entry.grid(row=1, column=1, padx=10, pady=5)  #保存评论内容到数据库：  # 获取当前最大的 commentID  cursor.execute("SELECT MAX(commentID) AS max\_id FROM comment")  result = cursor.fetchone()  max\_id = result['max\_id'] if result['max\_id'] else 0  new\_comment\_id = max\_id + 1  # 插入新评论到 comment 表  query = "INSERT INTO comment (commentID, commentTitle, commentContent, commentDate) VALUES (%s, %s, %s, %s)"  cursor.execute(query, (new\_comment\_id, title, content, datetime.now()))  2.7 Administrator manages news module (YuTao Sun)  # 查询新闻详情  cursor.execute("SELECT \* FROM news WHERE newsID = %s", (news\_id,))  news\_detail = cursor.fetchone()  # # 查询当前新闻下的所有评论  cursor.execute("SELECT comment.commentID, comment.commentTitle, comment.commentContent, comment.commentDate, user.userName FROM comment JOIN users\_comments ON comment.commentID = users\_comments.the\_CommentID JOIN user ON users\_comments.userID = user.userID WHERE comment.commentID IN (SELECT commentID FROM news\_comment WHERE newsID = %s)", (news\_id,))  comments = cursor.fetchall()  # 创建相关管理按钮  comment\_button = tk.Button(detail\_frame, text="管理新闻相关评论", width=20,command=lambda: open\_comment\_management\_window2(news\_id,admin\_window))  comment\_button = tk.Button(detail\_frame, text="删除当前新闻",width=20, command=lambda: delete\_news() )  comment\_button = tk.Button(detail\_frame, text="审核/取消审核新闻",width=20, command=lambda: check\_window())  comment\_button = tk.Button(detail\_frame, text="置顶/取消置顶新闻", width=20,command=lambda: top\_news())  2.8 Administrator manages comment module (Yutao Sun)  # 查询评论信息  cursor.execute("SELECT \* FROM comment WHERE commentID = %s", (comment\_id,))  comment\_info = cursor.fetchone()  # 查询发表该评论的用户ID  cursor.execute("SELECT userID FROM users\_comments WHERE the\_CommentID = %s", (comment\_id,))  user\_id = cursor.fetchone()['userID']  # 查询用户信息  cursor.execute("SELECT userName FROM user WHERE userID = %s", (user\_id,))  user\_name = cursor.fetchone()['userName']  # 显示评论信息  comment\_label = tk.Label(comment\_window, text=f"评论ID: {comment\_info['commentID']}\n标题: {comment\_info['commentTitle']}\n内容: {comment\_info['commentContent']}\n日期时间: {comment\_info['commentDate']}\n发布者用户ID: {user\_id}\n发布者用户名: {user\_name}")    # 添加“删除评论”按钮  delete\_button = tk.Button(comment\_window, text="删除评论", command=lambda: delete\_comment())  # 添加“访问发布者用户信息”按钮  user\_info\_button = tk.Button(comment\_window, text="访问发布者用户信息", command=lambda: open\_user\_detail(user\_id,comment\_window,admin\_window))    2.9 Administrator manages user module (Yutao Sun)  # 查询用户信息  with connection.cursor() as cursor:  cursor.execute("SELECT \* FROM user WHERE userID = %s", (userid,))  user\_info = cursor.fetchone()  # 查询用户评论  cursor.execute("SELECT the\_CommentID FROM users\_comments WHERE userID = %s", (userid,))  comment\_ids = cursor.fetchall()  comments = []  for comment\_id in comment\_ids:  cursor.execute("SELECT \* FROM comment WHERE commentID = %s", (comment\_id['the\_CommentID'],))  comment = cursor.fetchone()  comments.append(comment)  # 显示用户信息  user\_label = tk.Label(user\_frame, text=f"用户ID: {user\_info['userID']}\n姓名: {user\_info['userName']}\n密码: {user\_info['userPassword']}\n性别: {user\_info['sex']}\n邮箱: {user\_info['userEmail']}\n封禁状态: {user\_info['is\_block']}")  user\_label.pack(pady=10)  # 显示用户评论  comment\_label = tk.Label(user\_frame, text="用户评论:")  # 创建封禁/解除封禁按钮  ban\_button = tk.Button(user\_frame, text="封禁/解除封禁",width=15, command=toggle\_ban)  # 创建删除用户按钮  delete\_button = tk.Button(user\_frame, text="删除用户", width=15,command=delete\_user)  2.10 Administrator launch news module (Yutao Sun) def open\_edit\_news\_window():  # 获取当前日期  today = date.today()  # 新闻标题输入框  title\_label = tk.Label(edit\_news\_window, text="News Title:")  title\_entry = tk.Entry(edit\_news\_window, width=50)  # 新闻内容输入框  content\_label = tk.Label(edit\_news\_window, text="News Content:")  content\_entry = tk.Text(edit\_news\_window, height=6, width=50)  # 新闻摘要输入框  desc\_label = tk.Label(edit\_news\_window, text="News Description:")  desc\_entry = tk.Entry(edit\_news\_window, width=50)  # 新闻类别输入框  category\_label = tk.Label(edit\_news\_window, text="Category:")  category\_entry = tk.Entry(edit\_news\_window, width=50)  # 审核勾选框  check\_box = tk.Checkbutton(edit\_news\_window, text="审核", variable=check\_var)  # 置顶勾选框  top\_box = tk.Checkbutton(edit\_news\_window, text="置顶", variable=top\_var)  # 保存按钮  save\_button = tk.Button(edit\_news\_window, text="Save", command=save\_news)  cursor.execute("SELECT MAX(newsID) AS max\_id FROM news")  result = cursor.fetchone()  Max\_id=result['max\_id'] if result['max\_id'] else 0  max\_news\_id=Max\_id+1  # 插入新闻数据  cursor.execute("INSERT INTO news (newsID, newsTitle, newsContent, newDesc, category, newsDate, newsIsCheck, newsTop) VALUES (%s, %s, %s, %s, %s, %s, %s, %s)",(max\_news\_id , title, content, desc, int(category), today, int(is\_checked), int(is\_top)))  **3. Detailed design models for the modules**  3.1 Flow Chart of user/administrator login module (Yutao Sun)  C:/Users/hp/AppData/Local/Temp/绘图1(23).png绘图1(23)  Figure 1 Flow Chart of user/administrator login module  3.2 PAD Chart of user browsing news module (Yutao Sun)  C:/Users/hp/AppData/Local/Temp/绘图2(1).png绘图2(1)  Figure 2 PAD Chart of user browsing news module  3.3 Flow and PAD Chart of user creation module (Yutao Sun)  绘图3  Figure 3 PAD Chart of Login module  绘图1(24)  Figure 4 Flow Chart of Login module  3.4 Flow and PAD Chart of Administrator manages news module (Yutao Sun)  绘图4  Figure 5 Flow Chart of Administrator manages news module  C:/Users/hp/AppData/Local/Temp/绘图1(26).png绘图1(26)  Figure 6 PAD Chart of Administrator manages news module  3.5 Flow and PAD Chart of Administrator manages users module (Yutao Sun)  绘图1(26)  Figure 7 Flow Chart of Administrator manages users module  C:/Users/hp/AppData/Local/Temp/绘图1(27).png绘图1(27)  Figure 8 PAD Chart of Administrator manages users module  3.6 Flow and PAD Chart of Administrator manages comments module (Yutao Sun)  绘图1(27)  Figure 9 Flow Chart of Administrator manages comments module    C:/Users/hp/AppData/Local/Temp/绘图1(28).png绘图1(28)  Figure 10 PAD Chart of Administrator manages comments module  3.7 Flow and PAD Chart of launch news module (Yutao Sun)  绘图1(28)C:/Users/hp/AppData/Local/Temp/绘图1(29).png绘图1(29)  Figure 11 PAD and Flow Chart of launch news module  **4. Meeting minutes**  Date: [12.03]  Time: [15:00]  Location: [The 26th teaching building]  Participants:  Leader: Yutao Sun  Member: Robin Zhang  Agenda:  Review of the experiment objectives and guidelines  Discussion of the detailed design models for the system modules  Check if the pseudocode logic and flowchart are correct  Review of the completed report  Any other business  Minutes:  The meeting commenced with all members present and the leader, Yutao Sun, outlining the agenda for the discussion.  ·Review of the experiment objectives and guidelines  The meeting began with Yutao Sun, the meeting chair, briefly reviewing the objectives and requirements of the experiment. The goals include designing detailed design models for system modules (such as flowcharts, PAD diagrams, N-S diagrams, etc.), and using decision tables or decision trees to model the special business logic of algorithms.  ·Discussion of the detailed design models for the system modules  The design plans for various system functionality modules, including the backend database structure and logical relationships of the modules, were discussed. The team discussed the specific design details for each module, ensuring that the design models met the experiment requirements.  ·Check if the pseudocode logic and flowchart are correct  Yutao Sun presented the pseudocode for the backend database structure and functional modules. The team conducted a detailed discussion to ensure that the logic was clear and aligned with the system requirements. Additionally, the flowcharts and PAD diagrams were reviewed to ensure the correct flow and logical relationships for each module.  ·Review of the completed report  The team reviewed the completion status of the experiment report and confirmed the work for each module's design and implementation. Yutao Sun and Robin Zhang shared their progress and challenges encountered, and discussed how to improve the content of the report.  ·Any other business  Before concluding the meeting, Yutao Sun made suggestions about system integration and future work, including how to conduct integration testing for the modules and improve the user interaction aspects of the system. Robin Zhang proposed design improvements for the front-end interface.  ·Meeting Summary:  The meeting proceeded smoothly, and the following consensus was reached:  1.The design and pseudocode logic for each functional module were thoroughly discussed to ensure there were no omissions or inconsistencies.  2.The experiment report will be further improved, ensuring all details align with the requirements.  3.The next steps include module integration testing and the finalization of the report.  At the end of the meeting, everyone was satisfied with their progress and confirmed their tasks for the next phase. | | | | | | | | |
| * Experimental summary/ Analysis   The experiment provided valuable hands-on experience in system design, helping us understand how to break down complex systems into manageable modules. We not only created detailed pseudocode but also designed flowcharts and structural diagrams, which helped us visualize the relationships between different modules in the system.  The experiment also emphasized the importance of clear communication and thorough analysis, especially when discussing logic and integration between modules. It allowed us to combine theoretical knowledge with practical implementation, making it a key learning experience.  Overall, the experiment achieved its objectives and prepared us for future system design tasks, giving us a solid foundation for teamwork and project development. | | | | | | | | |
|  | Criteria | | | | | | | scale |
| Goal | | | | | | | A B C D E |
| Process | | | | | | |
| Design | | | | | | |
| Algorithm | | | | | | |
| Code | | | | | | |
| Data/Results | | | | | | |
| summary | | | | | | |
| written | | | | | | |
| Score | | |  | | tutor Signature：  Date: : | | |
| * Lab Evaluation Criteria   A: This lab is exceptional, working and meeting all of the specifications. The code is exceptionally well organized and very easy to follow. The code could be reused as a whole or each routine could be reused. The documentation is well written and clearly explains what the code is accomplishing and how. The program was delivered on time. The code is extremely efficient without sacrificing readability and understanding.  B: This lab is very good-- works and produces the correct results and displays them correctly. It also meets most of the other specifications. The code is fairly easy to read. Most of the code could be reused in other programs. The documentation consists of embedded comment and some simple header documentation that is somewhat useful in understanding the code. The program was delivered within a week of the due date. The code is fairly efficient without sacrificing readability and understanding.  C: This lab is adequate, with only minor deficiencies. The program produces correct results but does not display them correctly. The code is readable only by someone who knows what it is supposed to be doing. Some parts of the code could be reused in other programs. The documentation is simply comments embedded in the code with some simple header comments separating routines. The code was within 2 weeks of the due date. The code is brute force and unnecessarily long.  D: This lab shows some effort but has at least one major deficiency. The program is producing incorrect results. The code is poorly organized and very difficult to read. The code is not organized for reusability. The documentation is simply comments embedded in the code and does not help the reader understand the code. The code was more than 2 weeks overdue. The code is huge and appears to be patched together.  E: This lab is poorly written and shows very little effort or understanding. | | | | | | | | |