South China University of Technology

《Software Engineering》Experiment Report

Experiment Title： Experiment 2: Prototype and software testing

Name： 向天翼 宋浩瑞 Student ID：201836020389 201830581404

Class： 18计算机联合班 Group：

Collaborator：

Teacher： 李剑

|  |
| --- |
| **Description** |
| 【Objective and Requirement】  **Problem Description:**  The department of public works for a large city has decided to develop a on-line pothole tracking and repair system (PHTRS). A description follows:   * *Citizens can log onto this system and report the location and severity of potholes. As potholes are reported they are logged within a “public works department repair system” and are assigned an identifying number, stored by street address, size (on a scale of 1 to 10), location (middle, curb, etc.), district (determined from street address), and repair priority (determined from the size of the pothole).* * *Work order data are associated with each pothole and includes pothole location and size, repair crew identifying number, number of people on crew, equipment assigned, hours applied to repair, hole status (work in progress, repaired, temporary repair, not repaired), amount of filler material used and cost of repair (computed from hours applied, number of people, material and equipment used). Finally, a damage file is created to hold information about reported damage due to the pothole and includes citizen's name, address, phone number, type of damage, dollar amount of damage. PHTRS is an on-line system; all queries are to be made interactively.*   **Requirements:**   1. Development: Implement a prototype of PHTRS system  * Include user interfaces and one function * Implemented in Ruby (Rails) or other object-oriented programming languages (or Java, C++) * Object-oriented * MVC(Model-View-Controller) architecture pattern  1. Test:  * Draw a graph to show the structure (execution) of a selected procedure in your program (or your Ruby program). * Compute the complexity of this procedure according to the structure graph  1. Test case design: Design test case for the selected procedure  * Should cover all statements (or methods) of the procedure * Every branch should be exercised for true and false conditions * List inputs and corresponding execution paths   【Environment】  Windows 10 |
| **Content** |
| 【Procedure】  **Experiment 2.1**  We use MVC architecture to implement the prototype. The view is implemented by Vue.js framework and the model and controller are implemented by Spring Boot framework.  **View:**  According to the description of the problem, we define three kinds of user including citizens, workers, and system administrators.  Fig.1 shows the main interface for normal citizen, they can view the pothole information and its current status, indicate the process of repairing the pothole that they have reported. They can click ‘New Record’ button to submit new pothole information, then a new order will be created whose status is ‘submitted’.    Fig.1 Main interface for citizen user    Fig.2 Reporting dialog for citizen user  When a new order is created, workers can take this order in this system by submit repairing requirement information and modifying current state of this order, which is showed in Figure 4. Before the order is submitted to the inspection department, workers can modify and update the repairing requirements and status information of the work order at any time. If the repair is done, workers can click ‘Finish construction’ to submit this order to inspection department.      Fig.3 Main interface for worker user    Fig.4 Adding or editing order repairing information  The administrator can submit the inspection result from inspection department. If the inspect result is pass, then the administrator can delete this order. Otherwise it will rework by workers. Workers can resubmit the order after finished reworking.    Fig.5 Main interface for administrator  **Model:**  We implemented Model part of MVC using spring framework in JAVA language. We use Java JPA repository to visit the SQL database. We created several tables. We use several data access objects (DAO) to access data from the model. We use User table to store information of users such as username, password, mobile, name and address of citizens, workers and admin. We use Pothole table to store the information of the pothole situation such as workers who are responsible for it, the size, material and priority of it etc.  C:\Users\pc001\Documents\Tencent Files\384883654\Image\C2C\E2X1~H{HTG5@OB_J2RXJ4ZD.png  Fig.6 Part of source code of user model  **Controller:**  The controller part consists of two controllers, the work order controller and the login controller. The login controller gets requests of information of users from the frontend, query or update data from the model part, and respond the information back to the frontend. For example, the frontend and send a request using url “/user\_info/get” to get the information of the user. The work order controller handles requests about the information of potholes. It implements the four basic operations on the database, creating, updating, deleting and querying of potholes.  C:\Users\pc001\Documents\Tencent Files\384883654\Image\C2C\{N{7_2RHBZSM%B]ZGFH]J)6.png  Fig.7 Source code (function declaration of web interface) of login controller    Fig.8 Source code (function declaration of web interface) of workorder controller  This prototype has been published on <http://siriusxiang.xyz:8080/>.  **Experiment 2.2**  We will do the test on the following function:   |  | | --- | | void func(int x, int y) {  while (x < y)  {  y--;  if (y == 0)  break;  else if (y < 0)  {  y = y \* 2;  continue;  }  x += 2;  }  y++;  } |   The procedure structure graph is showed in following figure.    Fig.9 Procedure structure graph for testing case design  Since there are 13 edges and 11 nodes, the complexity is according to Euler’s principle.  **Experiment 2.3**  Since the complexity is 4, we can obtain 4 independent paths. Thus, we can design 4 different test cases to cover all the branches and all the statements.   |  |  |  |  | | --- | --- | --- | --- | | **Case No.** | **x** | **y** | **Execution path** | | 1 | 1 | 2 | 1 2 3 4 6 9 2 10 11 | | 2 | -2 | -1 | 1 2 3 4 6 7 8 2 10 11 | | 3 | 0 | 1 | 1 2 3 4 5 10 11 | | 4 | 1 | 0 | 1 2 10 11 |   We use a program to do testing on these four testing cases, the result is showed in following figure.    Fig.10 Inputs and corresponding execution paths  **Appendix I: Complete source code for testing**  #include <stdio.h>  void func(int x, int y) {  printf("1 ");  while (printf("2 "), x < y)  {  y--; printf("3 ");  if (printf("4 "), y == 0)  {  printf("5 "); break;  }  else if (printf("6 "), y < 0)  {  y = y \* 2; printf("7 ");  printf("8 "); continue;  }  x += 2; printf("9 ");  }  y++; printf("10 ");  printf("11 ");  }  void testFunc(int x, int y) {  printf("Test inputs: x = %d, y = %d\n", x, y);  printf("Execution path: ");  func(x, y);  printf("\n\n");  }  int main()  {  testFunc(1, 2);  testFunc(-2, -1);  testFunc(0, 1);  testFunc(1, 0);  } |
| **Conclusion** |
| From this lab session we ‘ve learned how to implement a prototype of a system from given description and design test cases based on the functions and structure of the software then apply these test cases on the software testing and show the testing results. This process meaningfully enhances our understanding of the designation of prototype and test case. |
| **Teacher’s Comments and Score** |
| Comment：  Score：           Signature：                                                 Date： |