Operational Specification Template

|  |  |  |  |
| --- | --- | --- | --- |
| **Student** | 康润 | **Date** | 2019.4.23 |
| **Program** | 助成招聘 | **Program #** |  |
| **Instructor** | 王辉 | **Language** | Java |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **场景编号** | **1** | **用户目的** | 投递简历 | |
| **Scenario 场景目的** | | 描绘一次完整投递简历的过程 | | |
| **操作来源** | **步骤** | **动作** | | **备注信息** |
| 用户 | 1 | 启动系统 | |  |
| 系统 | 2 | 要求输入用户名和密码 | |  |
| 用户 | 3 | 用户进行输入 | |  |
| 系统 | 4 | 检测，成功则进入 | | 检查用户名和密码是否正确 |
| 用户 | 5 | 搜索招聘信息 | |  |
| 系统 | 6 | 显示招聘信息 | |  |
| 用户 | 7 | 点击招聘职位 | |  |
| 系统 | 8 | 显示职位详细介绍 | |  |
| 用户 | 9 | 点击投递简历 | |  |
| 系统 | 10 | 将简历发送到招聘方 | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **场景编号** | **2** | **用户目的** | 发布招聘信息 | |
| **Scenario 场景目的** | | 描绘一次成功发布招聘信息的过程 | | |
| **操作来源** | **步骤** | **动作** | | **备注信息** |
| 用户 | 1 | 启动系统 | |  |
| 系统 | 2 | 要求输入用户名和密码 | |  |
| 用户 | 3 | 用户进行输入 | |  |
| 系统 | 4 | 检测，成功则进入 | | 检查用户名和密码是否正确 |
| 用户 | 5 | 点击发布招聘信息 | |  |
| 系统 | 6 | 显示填写招聘信息打得界面 | |  |
| 用户 | 7 | 填写招聘信息内容 | |  |
| 系统 | 8 | 进行预览 | |  |
| 用户 | 9 | 点击发布 | |  |
| 系统 | 10 | 系统显示招聘信息 | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **场景编号** | **3** | **用户目的** | 查看简历 | |
| **Scenario 场景目的** | | 描绘一次成功查看简历的过程 | | |
| **操作来源** | **步骤** | **动作** | | **备注信息** |
| 用户 | 1 | 启动系统 | |  |
| 系统 | 2 | 要求输入用户名和密码 | |  |
| 用户 | 3 | 用户进行输入 | |  |
| 系统 | 4 | 检测，成功则进入 | | 检查用户名和密码是否正确 |
| 用户 | 5 | 点击对应职位 | |  |
| 系统 | 6 | 显示职位接收简历界面 | |  |
| 用户 | 7 | 点击对应简历 | |  |
| 系统 | 8 | 显示简历详细信息 | |  |
| 用户 | 9 | 点击通过简历 | |  |
| 系统 | 10 | 通知求职者用户简历通过 | |  |







Operational Specification Template Instructions

|  |  |
| --- | --- |
| Purpose | * To hold descriptions of the likely operational scenarios followed during program use * To ensure that all significant usage issues are considered during program design * To specify test scenarios |
| General | * Use this template for complete programs, subsystems, or systems. * Group multiple small scenarios on a single template, as long as they are clearly distinguished and have related objectives. * List the major scenarios and reference other exception, error, or special cases under comments. * Use this template to document the operational specifications during planning, design, test development, implementation, and test. * After implementation and testing, update the template to reflect the actual implemented product. |
| Header | * Enter your name and the date. * Enter the program name and number. * Enter the instructor’s name and the programming language you are using. |
| Scenario Number | Where several scenarios are involved, reference numbers are needed. |
| User Objective | List the users’ likely purpose for the scenario, for example, to log onto the system or to handle an error condition. |
| Scenario Objective | List the designer’s purpose for the scenario, for example, to define common user errors or to detail a test scenario. |
| Source | * Enter the source of the scenario action. * Example sources could be user, program, and system. |
| Step | Provide sequence numbers for the scenario steps. These facilitate reviews and inspections. |
| Action | Describe the action taken, such as   * Enter incorrect mode selection. * Provide error message. |
| Comments | List significant information relating to the action, such as   * User enters an incorrect value. * An error is possible with this action. |

Functional Specification Template

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Student** | | |  | | **Date** |  |
| **Program** | | |  | | **Program #** |  |
| **Instructor** | | |  | | **Language** |  |
|  | | | | | | |
| **Class Name** | |  | | | | |
| **Parent Class** | |  | | | | |
|  | |  | | | | |
|  | |  | | | | |
|  | |  | | | | |
|  | | | | | | |
| **Attributes** | | | | | | |
|  | **Declaration** | | | **Description** | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  | | | | | | |
| **Items** | | | | | | |
|  | **Declaration** | | | **Description** | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |

Functional Specification Template Instructions

|  |  |
| --- | --- |
| Purpose | * To hold a part’s functional specifications * To describe classes, program modules, or entire programs |
| General | * Use this template for complete programs, subsystems, or systems. * Use this template to document the functional specifications during planning, design, test development, implementation, and test. * After implementation and testing, update the template to reflect the actual implemented product. |
| Header | * Enter your name and the date. * Enter the program name and number. * Enter the instructor’s name and the programming language you are using. |
| Class Name | * Enter the part or class name and the classes from which it directly inherits. * List the class names starting with the most immediate. * Where practical, list the full inheritance hierarchy. |
| Attributes | * Provide the declaration and description for each global or externally visible variable or parameter with any constraints. * List pertinent relationships of this part with other parts together with the multiplicity and constraints. |
| Items | * Provide the declaration and description for each item. * Precisely describe the conditions that govern each item’s return values. * Describe any initialization or other key item responsibilities. |
| Example Items | An item could be a class method, procedure, function, or database query, for example. |

State Specification Template

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Student | |  | | | Date | |  |
| Program | |  | | | Program # | |  |
| Instructor | |  | | | Language | |  |
|  | | |  | | | | |
| **State Name** | | | | **Description** | | | |
|  | | | |  | | | |
|  | | | |  | | | |
|  | | | |  | | | |
|  | | | |  | | | |
|  | | | |  | | | |
| **Function/Parameter** | | | | **Description** | | | |
|  | | | |  | | | |
|  | | | |  | | | |
|  | | | |  | | | |
|  | | | |  | | | |
|  | | | |  | | | |
|  | | | |  | | | |
| **States/Next States** | | | | **Transition Condition** | | **Action** | |
|  | | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |

|  |  |  |
| --- | --- | --- |
| State Specification Template Instructions | | |
| Purpose | * To hold the state and state transition specifications for a system, class, or program * To support state-machine analysis during design, design reviews, and design inspections |
| General | * This form shows each system, program, or routine state, the attributes of that state, and the transition conditions among the states. * Use this template to document the state specifications during planning, design, test development, implementation, and test. * After implementation and testing, update the template to reflect the actual implemented product. |
| Header | * Enter your name and the date. * Enter the program name and number. * Enter the instructor’s name and the programming language you are using. |
| State Name | * Name all of the program’s states. * Also enter each state name in the header space at the top of each “States/Next States” section of the template. |
| State Name Description | * Describe each state and any parameter values that characterize it. * For example, if a state is described by SetSize=10 and SetPosition=3, list SetSize=10 and SetPosition=3. |
| Function/Parameter | * List the principal functions and parameters. * Include all key variables or methods used to define state transitions or actions. |
| Function/Parameter Description | * For each function, provide its declaration, parameters, and returns. * For each parameter, define its type and significant values. |
| Next State | * For each state, list the names of all possible next states. * Include the state itself. |
| Transition Condition | List the conditions for transition to each next state.   * Use a mathematical or otherwise precise notation. * If the transition is impossible, list "impossible," with a note saying why. |
| Action | List the actions taken with each state transition. |

Logic Specification Template

|  |  |  |  |
| --- | --- | --- | --- |
| Student |  | Date |  |
| Program |  | Program # |  |
| Instructor |  | Language |  |

|  |  |
| --- | --- |
| **Design** |  |
| **References** |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parameters** |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Logic Specification Template Instructions

|  |  |
| --- | --- |
| Purpose | * To contain the pseudocode for a program, component, or system * To enable precise and complete program implementation * To facilitate thorough design and implementation reviews and inspections |
| General | * Use this template to document the program’s detailed logic. * After implementation and testing, update the template to reflect the actual implemented product. * During detailed design, write the pseudocode needed to describe all of the program’s logic. * Use plain language and avoid using programming instructions wherever practical. |
| Header | * Enter your name and the date. * Enter the program name and number. * Enter the instructor’s name and the programming language you are using. |
| Design References | List the references used to produce the program’s logical design.   * the Operational, Functional, and State templates * the program’s requirements * any other pertinent source |
| Parameters | * Where needed, define any parameters or abbreviations used. * Avoid duplicating definitions on other templates and reference these other definitions where they are needed. |