## PSP2 Design Review Checklist

|  |  |  |  |
| --- | --- | --- | --- |
| Student | DIEGO ANDRES MONTEALEGRE GARCIA | Date | 25/02/2015 |
| Program | PSP 2 | Program # | 5 |
| Instructor | LUIS DANIEL BENAVIDES | Language | Java |

|  |  |
| --- | --- |
| * Purpose | To guide you in conducting an effective design review |
| * General | * Review the entire program for each checklist category; do not attempt to review for more than one category at a time! * As you complete each review step, check off that item in the box at the right. * Complete the checklist for one program or program unit before reviewing the next. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | 1.Gestor  Negocio | 2. Vista App | 3.Simpson | 4. PrettyPrinting |
| Complete | Verify that the design covers all of the applicable requirements.   * All specified outputs are produced. * All needed inputs are furnished. * All required includes are stated. | x | x | x | x |
| Logic | * Verify that program sequencing is proper.   Stacks, lists, and so on are in the proper order.  Recursion unwinds properly.   * Verify that all loops are properly initiated, incremented, and terminated. * Examine each conditional statement and verify all cases. * Verificar los flujos de control. | x | x | x | x |
| Internal Limits | Where the design assumes or relies upon internal limits, determine if behavior is correct at nominal values, at limits, and beyond limits. | x | x | x | x |
| Special Cases | * Check all special cases. * Ensure proper operation with empty, full, minimum, maximum, negative, and zero values for all variables. * Protect against out-of-limits, overflow, and underflow conditions. * Ensure “impossible” conditions are absolutely impossible. * Handle all possible incorrect or error conditions. | x | x | x | x |
| Functional Use | * Verify that all functions, procedures, or methods are fully understood and properly used. * Verify that all externally referenced abstractions are precisely defined. | x | x | x | x |
| System Considerations | * Verify that the program does not cause system limits to be exceeded. * Verify that all security-sensitive data are from trusted sources. * Verify that all safety conditions conform to the safety specifications. | x | x | x | x |
| Names | Verify that   * all special names are clear, defined, and authenticated * the scopes of all variables and parameters are self-evident or defined * all named items are used within their declared scopesall named items are used within their declared scopes | x | x | x | x |
| Standards | Ensure that the design conforms to all applicable design standards. | x | x | x | x |
| Specification | -Verificar:  -Todas las clases están documentadas.(Javadoc)  -Todos los miembros y métodos de la clase están documentados. (Javadoc)  -La visibilidad de las clases, miembros y métodos cumplen con el estándar de codificación. | x | x | x | x |
| Unit test | - Verificar que cada clase:  - Tenga una clase de prueba.  - Todos los método de la clase es probado.  - Todos los métodos y clases deben estar documentados | x | x | x | x |