|  |  |  |  |
| --- | --- | --- | --- |
| **BAHAGIAN PENDIDIKAN DAN LATIHAN TEKNIK VOKASIONAL**  **KEMENTERIAN PENDIDIKAN MALAYSIA**  **ARAS 5 & 6, BLOK E14, KOMPLEKS E,**  **PUSAT PENTADBIRAN KERAJAAN PERSEKUTUAN**  **KERTAS PENERANGAN**  ***(INFORMATION SHEET)*** | | | |
| **KOD DAN NAMA PROGRAM NOSS** | | IT-010-3:2016 PEMBANGUNAN APLIKASI | |
| **TAHAP** | | 3 | |
| **KOD, NAMA CU DAN WA NOSS** | | **CU2/WA1 – INTERPRET APPLICATION MODULE DEVELOPMENT REQUIREMENT** | |
| **KOD DAN NAMA PROGRAM KV** | | SISTEM PENGURUSAN PANGKALAN DATA DAN APLIKASI WEB | |
| **KOD DAN NAMA KURSUS KSKV** | | KPD 2023 WEB PROGRAMMING | |
| **NO.DAN TAJUK STANDARD KANDUNGAN KSKV** | | K1 INTERPRET APPLICATION MODULE DEVELOPMENT REQUIREMENT | |
| **NO. KOD** | **NO. KOD NOSS** | IT-010-3:2016-C02/P(3/70) | Muka Surat : 1 Drp : 8 |
| **NO. KOD KSKV** | KPD 2023 / P(1/11) |

**TAJUK/***TITLE* **:**

**KEPERLUAN PEMBANGUNAN APLIKASI MODUL**

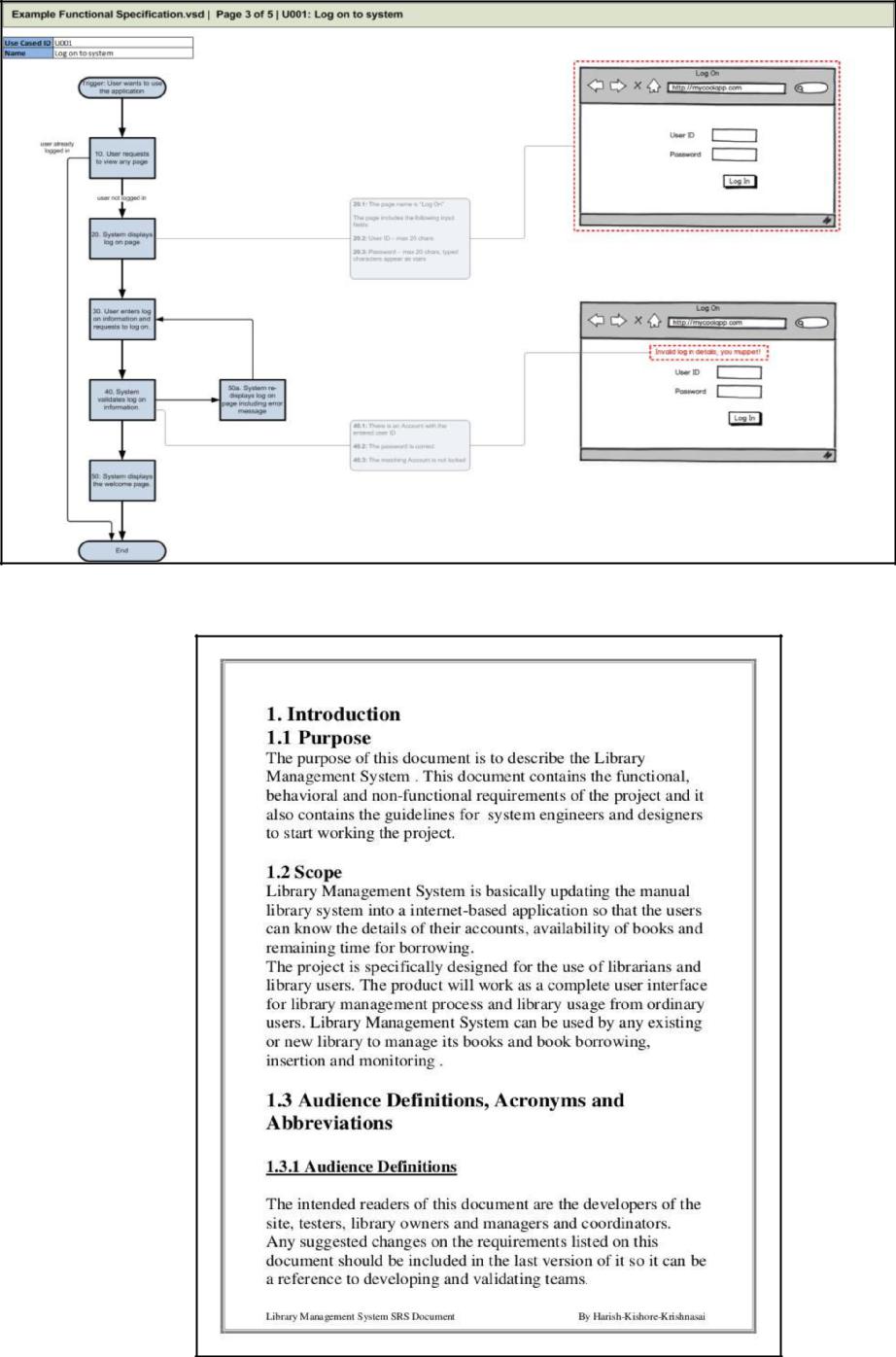
**TUJUAN/***PURPOSE* **:**

Kertas penerangan ini adalah bertujuan menerangkan mengenai :

1. Jenis-jenis dokumen spesifikasi kefungsian
2. Fungsi *Software Requirement Specification (SRS)*
3. Pengenalan kepada *scripting language*
4. Fungsi dalam *Integrated Development Environment (IDE)*

***1.1 FUNCTIONAL SPECIFICATION* DOCUMENT (FSD)**

***Functional specification document (FSD)*** boleh dibuat dalam pelbagai bentuk. Iaboleh di terjemahkan dalam bentuk gambarajah mahupun teks.



**Rajah 1**

**Tujuan *Functional specification* document (FSD)**

* untuk mencapai kata sepakat antara pasukan projek sebelum sistem dibangunkan.
* Untuk memaklumkan kepada pembangun mengenai sistem yang dibina
* Untuk penguji mengetahui ujian yang bakal dijalankan ke atas sistem
* Untuk membolehkan pihak berkepentingan tahu apa yang mereka dapat

**JENIS *FUNCTIONAL SPECIFICATION* DOCUMENT (FSD)**

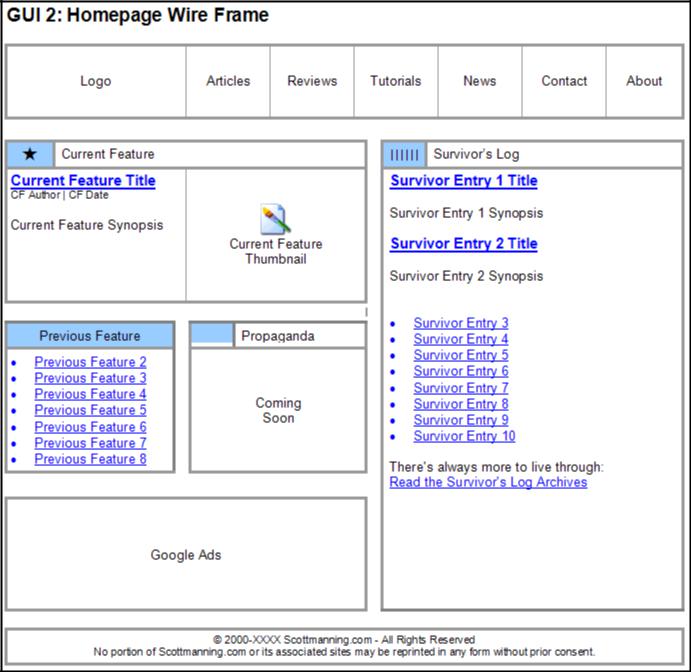
1. **Functional Design Specification (FDS)**

Spesifikasi Rekabentuk Berfungsi atau FDS adalah suatu dokumen yang digunakan oleh pembangun dalam fasa pra-pembangunan untuk menterjemahkan semua nota, konsep dan skop ke dalam dokumen keperluan yang lengkap. Dokumen ini boleh mengandungi carta alir, *screenshot* dan *wireframe* yang menerangkan senarai keperluan yang boleh digunakan untukpembangunan dan pengujian.

Dengan menggunakan FDS:

1. Pembangun tahu dengan tepat apa yang akan dibangunkan o Bahagian Jaminan Kualiti akan tahu apa yang akan diuji

o Pengguna tahu apa yang akan mereka dapat pada akhir fasa pembangunan



**Rajah 2 contoh FDS dalam bentuk *wireframe***

1. **Software Design Document (SDD)**

Dokumen Rekabentuk Perisian (SDD) adalah dokumen yang memberi penerangan bertulis mengenai produk perisian (*software).* Dokumen ini ditulis oleh pereka perisian kepada pembangun perisian sebagai panduan mengenai keseluruhan seni bina projek.

SDD mengandungi informasi berikut:

Reka bentuk data yang menerangkan struktur dalam perisian



Reka bentuk antaramuka *(Interface)* menerangkan antaramuka dalaman dan luaran sistem serta antaramuka pengguna



Reka bentuk prosedur menerangkan konsep pengaturcaraan berstruktur menggunakan grafik, jadual dan teks.



**1.2 FUNGSI SPESIFIKASI KEPERLUAN PERISIAN (SRS)**

**Definisi Spesifikasi Rekabentuk Perisian**

Sebuah spesifikasi keperluan perisian SRS adalah dokumen yang mengandungi penerangan lengkap tentang bagaimana sistem itu dijangka untuk berfungsi. Ia biasanya ditandatangani pada akhir fasa keperluan kejuruteraan.

* Lebih terperinci daripada spesifikasi keperluan sistem
* Deskripsi yang abstrak mengenai rekabentuk perisian sebagai rujukan utama untuk rekabentuk dan implementasi.

**Kualiti SRS**

* Betul: Setiap keperluan yang diberikan dalam SRS adalah satu keperluan perisian.
* Jelas: Setiap keperluan mempunyai tepat satu tafsiran.
* Lengkap: Termasuk semua fungsi, prestasi, reka bentuk, keperluan antara muka luaran; Takrif sambutan perisian untuk semua input.
* Konsisten: Konsisten Dalaman.
* Kepentingan Kedudukan: Penting vs. wajar.
* Disahkan: Sebuah keperluan boleh disahkan jika dan hanya jika wujud beberapa proses yang berkesan kos terhingga dengan mana orang atau mesin boleh menyemak bahawa perisian memenuhi keperluan.
* Diubah suai: SRS mestilah berstruktur untuk membolehkan pengubahsuaian berkesan (cth jangan berlebihan, menjaga keperluan memisahkan)
* Dikesan: Asal setiap keperluan adalah jelas.

**Jenis Keperluan**

Gambar rajah di bawah menggambarkan pelbagai jenis keperluan yang dirakamkan semasa SRS.



**Rajah 3**

**Tujuan SRS**

* SRS dengan tepat mentakrifkan produk perisian yang akan dibina.
* SRS digunakan untuk mengetahui semua keperluan untuk pembangunan perisian dan dengan itu yang akan membantu dalam mereka bentuk perisian.
* Ia menyediakan maklum balas kepada pelanggan.

**Perbezaan SRS dengan SDD**

|  |  |  |  |
| --- | --- | --- | --- |
| **SRS** | | **SDD** | |
| ✓ SRS menerangkan tentang | | ✓ Reka bentuk menggariskan | |
| proses pembangunan sistem. | | bagaimana sistem akan | |
| • | Ia berdasarkan kepada | berfungsi. | |
|  | keperluan pengguna | • | Ia berdasarkan kepada |
| • | Ia menetapkan peringkat |  | SRS |
|  | untuk reka bentuk dan ujian | • | Ia menetapkan peringkat |
|  | sistem |  | untuk ujian pelaksanaan |
| ✓ Pelanggan perlu menyemak | |  | dan integrasi |
|  | SRS dan membantu untuk | ✓ Biasanya hanya ahli pasukan | |
|  | mengesahkannya. | pembangunan yang perlu | |
| ✓ Pengesahan keperluan dibuat | | menyemak reka bentuk. | |
|  | dengan memeriksa keperluan |  |  |
|  | bertulis terhadap realiti (tidak |  |  |
|  | tertulis) |  |  |

**1.3 PENGENALAN KEPADA BAHASA SKRIP (*Scripting Language)***

*Scripting Language*  adalah bahasa pengaturcaraan aras tinggi yang diterjemah oleh perisian yang lain pada masa larian (*runtime)*

Terdapat dua jenis *scripting language* iaitu

a) *server- side scripting language*

b) *client-side scripting language*

Perbandingan antara bahasa pengaturcaraan skrip *server-side* dan *client-side*

|  |  |  |
| --- | --- | --- |
| Asas perbandingan | *Server-side* | *Client-side* |
| asas | 1. Kod atau skrip Berfungsi,dijalankan dan diproses pada peringkat komputer pelayan.  2. Kod tidak dilihat oleh *client* | 1. Berfungsi pada peringkat pengguna (*client*)  2. skrip *atau* kodboleh dilihat oleh *client* (*browser)* |
| Pemprosesan | Memerlukan interaksi dengan pelayan (*server)* | Tidak memerlukan interaksi dengan pelayan |
| Kesan | Boleh membuat pengubahsuaian dan memberi kesan dinamik kepada laman web | Boleh mengurankan beban kepada komputer pelayan |
| Keselamatan | Lebih selamat | Kurang selamat |
| contoh | PHP, ASP.net, ColdFusion, Python, C++,C# , Javascript dan lain-lain | HTML, CSS, Javascript, Vbscript, AJAX dan lain-lain |

**1.4 PENGENALAN KEPADA INTEGRATED DEVELOPMENT ENVIRONMENT (IDE)**

**Maksud IDE**

*Integrated Development Environment* (IDE) atau lebih dikenali sebagai Pembangunan Persekitaran Bersepadu merupakan aplikasi perisian yang **menyediakan kemudahan menyeluruh kepada pengaturcara komputer untuk membuat satu program**. IDE biasanya terdiri daripada *source code**editor build automation tools dan debugger*. Kebanyakkan IDE moden mempunyai *object- oriented software development* (objek berorientasikan pembangunan perisian).

Sesetengah IDE mengandungi *compiler, interpreter* atau kedua-duanya, seperti *NetBeans* dan *Eclipse*. Terdapat juga IDE yang tidak mempunyai *compiler* dan *interpreter* seperti *SharpDevelop* dan *Lazarus*. Sempadan antara IDE adalah masih kurang jelas. Kadang kala sistem kawalan versi, atau pelbagai alat untuk memudahkan pembinaan *Graphical User Interface* (GUI) telah disepadukan. Kebanyakkan IDE moden juga mempunyai *class browser, object browser* dan gambar rajah kelas hierarki, untuk digunakan dalam *object-oriented software development* (objek berorientasikan pembangunan perisian).

**Fungsi IDE**

Secara asasnya, IDE memiliki kemudahan seperti;

1. *EditKor*: Untuk menulis *source code*.
2. *Compiler*: Untuk menganalisis sintaks kod yang dibuat dan mengubahnya ke dalam bentuk *binary* yang sesuai dengan bahasa mesin.
3. *Debugger*: Untuk melakukan percubaan terhadap kod yang telah dibuat.
4. *Linker*: Untuk menggabungkan data *binary* dari beberapa kod telah dihasilkan oleh *compiler* sehingga data – data *binary* menjadi satu dan menjadi satu program komputer yang dapat digunakan.

**Contoh IDE**

* C/C++: MS Visual C++
* Delphi: Borland Delphi
* Java: Netbeans, Eclipse
* Pascal: Turbo Pascal

**SOALAN /** *QUESTION:*

1. Nyatakan tujuan Spesifikasi Keperluan Perisian (SRS).
2. Senaraikan contoh *Integrated Development Environment (IDE)* yang biasa digunakan.

**RUJUKAN /** *REFERENCE:*

|  |  |
| --- | --- |
| 1. | http://slideplayer.info/slide/2772293/#, Kompleksitas Pengembangan Perangkat Lunak, 27 Ogos 2019,16:23. |
| 2. | https://www.slideshare.net/CherryBerry2, uml-diagram-software-engineeringdiscussion, 27 Ogos 2019,17:00. |
| 3. | http://www.onextrapixel.com, Web Design and Development, 27 Ogos 2019 16:45 |
| 4. | https://techdifferences.com, Difference between server-side-scripting and client-side-scripting, 27 Ogos 2019, 17:00 |