SambaKodiPi A Personal File Server and Media Center

Nurfauza Jali, Vincent Meringgai Birang, Cheah Wai Shiang, Ahmad Hadinata Fauzi and Suriati Khartini Jali
Faculty of Computer Science & Information Technology,
Universiti Malaysia Sarawak, 94300 Kota Samarahan, Sarawak, Malaysia.
jnurfauza@unimas.my

Abstract—Conventional file sharing and media viewing usually involve slow and tedious data transfer. This project aims to provide access convenience for sharing data and direct-viewing of media files by having file server (FS) and media center (MC) capability using Raspberry Pi (RPi). The system consists of several functionalities that were developed through iterative and incremental development. The resulting system has its FS function catered by Samba program and its MC function catered by Kodi program. Direct MC output is on high-definition television (HDTV). The web-based user interface (WebUI) provides administrative functions for the system, its FS and users management and indirect access to its MC function through web player for all registered users. The system has undergone several testing processes, and it is a working prototype of an economical and feasible file server and media center using RPi. The system can still be improved with other functions and features in the future.

***Index Terms*—File Server; Media Center; Raspberry Pi; Samba.**

I. Introduction

In a modern world, data and media are mostly in digital form. Nowadays, data storing has become more crucial and important. Data storage has also steadily evolved from the early usage of diskettes and compact-discs (CDs) to Digital Video Discs (DVDs) and Blu-Ray Discs (BDs), and USB flash drives to external Hard-Disk Drives (HDDs). While storing data externally in bigger size storage eases the huge file size problem, it, however, gave rise to the tedious way of accessing them.

The conventional way of storing data and media files usually include several distributed electronic devices for example documents are stored in personal computers (PCs) while large size media files and backup data are stored in external HDDs. The process of obtaining a desired file usually involves a time-consuming copy-and-paste process. For media files, one needs to transfer it to another external storage device and connect it to media output devices. All these can be quite annoying if one need those files on a frequent basis.

A typical solution to solve the problem with access convenience is to use commercial file server which usually has very large storage size to store all the files in one place. However, the current value of a commercial file server is quite expensive and thus many still prefer the conventional way of managing their files. Moreover, a commercial file server may have limited usage and capability (Hanson). The objectives of this project are; (1) To create a file server which hosts both data (documents, executables, generic files) and media files (audio, video, image files) using Raspberry Pi. (2) To equip the file server with media center capability by installing a media center program and connect it with media output devices (monitor/TV/speakers). And (3) To build a main system interface (web-based) for users to interact and use the functions easily. The resulting system is to have at least two main functions: file server capability to host and share stored files; and media player capability to access media files directly. The proposed personal file server and media center is a working prototype of a personal file server equipped with media player capability.

II. Literature Review

The Raspberry Pi (RPi) device is essentially a low-cost, credit-card sized computer that plugs into a computer monitor or TV [1] [2]. It is a small single-board computer that also uses standard keyboard and mouse. Despite its tiny size, it has a broad range of usage potential in various kinds of electronic projects. Like an ordinary computer, it has the basic capabilities of a desktop PC ranging from functionality to browse the Internet, making spreadsheets, word-processing, playing games and even capable of playing high-definition video. Due to its cheap cost and yet capable and portable, the device has become increasingly popular to be used in various electronic projects.

According to [2], a file server is a computer or device that is connected within a network, dedicated to storing files and any user connected on the network can store files on the server. While [3] described file server as a computer that acts as a central storage and manage data so that they are accessible to other computers on the same network. And [4] defines file server as computer in a Local Area Network (LAN) that serve as centralized data storage for other machines as part of the client-server model of computer networking.

File servers enable users to share information over a network without the need to physically transfer them to other external storage devices. There are two main type of file server – dedicated and non-dedicated. A dedicated file server only serves as a file server and does not do any other tasks such as processing [2]. It serves solely as a central storage and hosts the data for other computers in its network.

Media center can be described as an electronic device that has the capability to support the playback of media files such as video, audio and image on external video and audio output hardware. In the context of this project, media center refers to the Home Theater Personal Computer (HTPC). By definition, an HTPC is a “personal computer that is used to store and play music and movies as well as display photos” [5]. An HTPC is typically physically connected to several other output devices such as television, PC monitor and speakers. It may include a remote control and a GUI interface for users to easily navigate between the media files stored in the HTPC storage. Alternatively, users can simply use standard keyboard and mouse. There are several media center programs available that can provide an interface, navigation and access to the media files on an HTPC. The most popular ones are the Windows Media Center, XBMC and Boxee.

Overall, the XBMC is the best media center among the three simply because it has the highest supports for media format which is very important as a key feature for the proposed file server in this project. Besides, it can run on various other platforms and has great customizable GUI design aside from having support for wide range of remote control.

Based on the analysis of comparisons done, it was decided that the system be developed builds on the best selection of RPi model, file server software and media center program. Thus, the proposed system was built on RPi model B+, using Samba as its file server software and XBMC program to provide it with media center capability.

These choices are much preferable due to the reasons described in the comparison analysis. And since the proposed system includes media center capability, this directly makes it a non-dedicated file server. The system is also improved with a web-based interface for users to interact with the system and utilise all the system functions. As for the architectural design and schematics, it was built based on the proposed architecture shown in Figure 1.

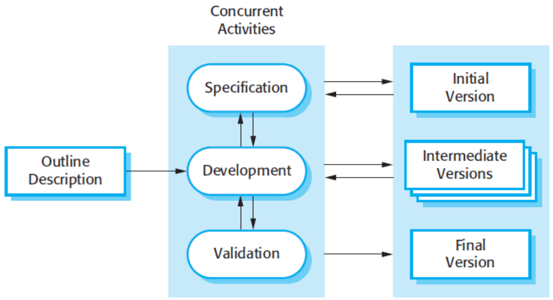


Figure 1: The architecture of incremental development

III. Methodology

The methodology for the implementation of the personal file server and media center using RPi was based on the iterative and incremental development. The development of the main system interface used this methodology as the functions on the web-based user interface were added iteratively and incrementally. Incremental development is chosen because the system is to have several functionalities that need to be delivered separately and added incrementally.

According to [5], incremental development is basically the process of developing an initial implementation, expose it for user feedback and improve it through several versions until an adequate system is finally developed. He further stated that specification, development and validation activities are interleaved as concurrent activities rather than separate, with rapid feedback across activities. Incremental development consists of several elements: outline description, concurrent activities (consist of specification, development and validation) and the system versions (consist of initial, intermediate and final version). The details for each are as described in Table 1. The resulting system functionalities and features will be implemented through a series of specification, development and validation as described previously.

Table 1
The concurrent activities in incremental development

Elements	Description
Outline Description	The basic outline that describes the system requirement (the system functionalities to be developed).
Concurrent Activities	The interleaved activities of main implementation process. They are to be performed iteratively to implement each of the system functionalities based on the outline description from end-user.
a) Specification	The specification activity is where system functionality is specified in details terms of technicality and expected outcome.
b) Development	The development activity is where system functionality is being implemented based on its specification.
c) Validation	The validation activity is where the implemented system functionality is being tested and validated with its specification.
System Versions	The variation of system versions as it is being developed throughout the implementation phases.
a) Initial version	The initial version is where the system has its most basic function and feature implemented.
b) Intermediate version	The intermediate version is where the system has some additional functions and features being added incrementally through a series of iterative implementation process.
c) Final version	The final version is where the system has all its functions and features fully implemented and ready for delivery to end-user.

IV. System Architecture SambaKodiPi

A. System Architecture

The architecture for this system refers to the system physical and hardware configurations. Figure 2 shows the architecture for intermediate/final version of the system.

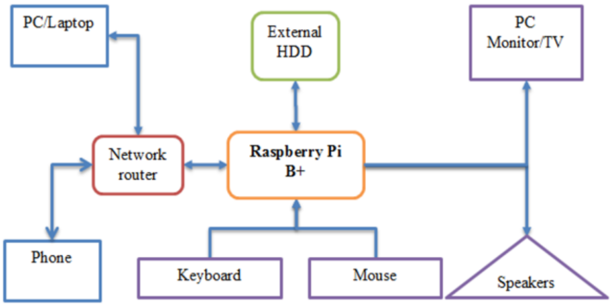


Figure 2: The Architecture for SambaKodiPi

The architecture consists of the system of its main and additional functions and features: file server function, media center function and web-based interface. The file server uses external hard-disk drives (HDDs) as its main storage and a network router for its network connection. The system ideally, is to work with two external HDDs to separate the data and media files for accessibility performance. The media center function uses standard video output devices (PC monitor or TV), standard audio output device (speakers) and standard input devices (keyboard and mouse). The file server hosts both data and media files, and they are accessible to users connected to the same network through LAN or Wi-Fi. Admin user can configure the Samba file server directly through Raspbian OS while all users can access its media center function directly through Kodi media center program. Alternatively, admin can configure Samba server through the web-based user interface (WebUI).

B. System Schematics

In term of proper physical connection through wirings, the schematics of the final system are as shown in Figure 3.

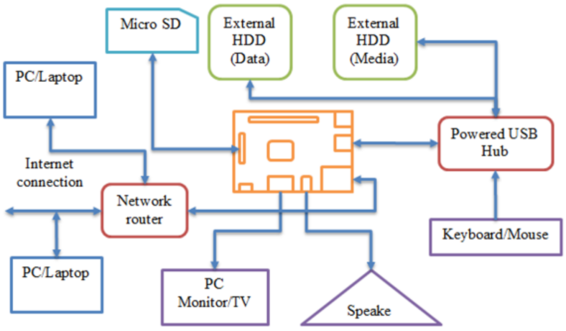


Figure 3: The schematic of the system

C. System Setup and Specification

The initial setup between the RPi and other devices are as shown in Figure 4 below:

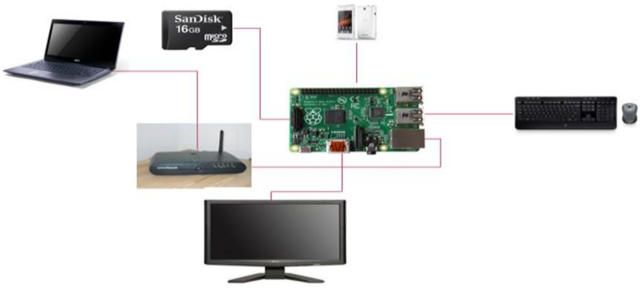


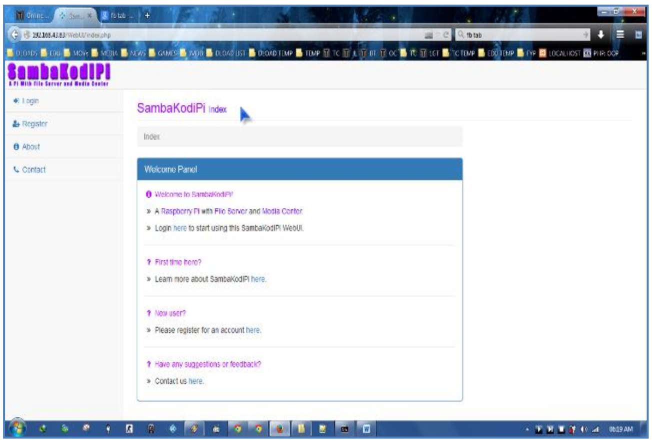
Figure 4: Initial setup of the system

Table 5
System components and their specification

Name	Type	Model/Version	Description
Raspberry Pi	Device	Model B+	Central controller device
Modem / Router	Device	Speedtouch585 v6	Main router for interconnection
PC Monitor	Device	Acer x193HQ	Default output display for Rasp Pi
Laptop	Device	Acer Aspire 4750G	Default PC for access/programming
Keyboard	Device	Logitech K120	Default input device
Mouse	Device	Logitech M185	Default input device, wireless
Laptop	Device	Acer 4750G	Default PC for access/programming
Smartphone	Device	Sony Xperia E	Default Wi-Fi hotspot for Internet
Raspbian	Software	3.12.22+	Default OS for Rasp Pi
HDMI cable	Cable	Standard HDMI	For video output to monitor/screen
LAN cables	Cable	Standard LAN	For physical connection with router
HDMI to DVI	Adapter	F-M adapter	For connection with PC monitor

D. Web-based User Interface (WebUI)

The Web-based UI can configure Samba server shares and server settings. The WebUI provides secondary media function where the user can play and view media files through web player. Other than that, the WebUI has function that can manage its users. Figure 5 – Figure 10 show the screenshots of the WebUI pages.



Welcome to SambaKodiPi!

» A Raspberry Pi with File Server and Media Center.

» Login here to start using this SambaKodiPi WebUI.

First time here?

» Learn more about SambaKodiPi here.

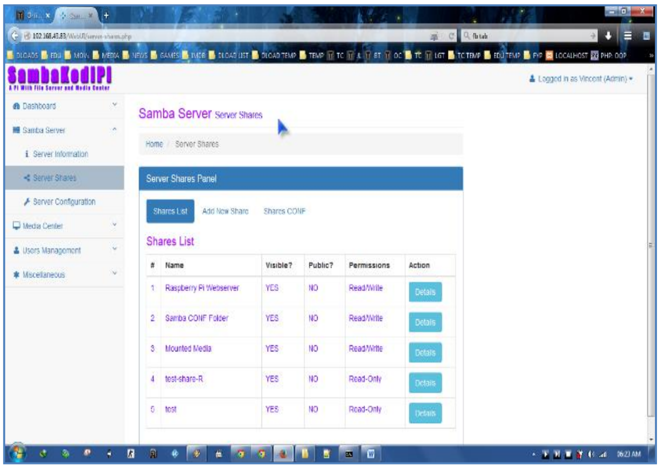
New user?

» Please register for an account here.

Have any suggestions or feedback?

» Contact us here.

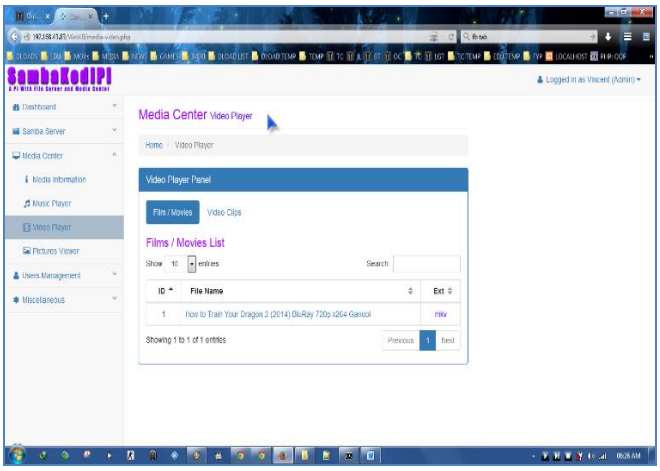
Figure 5: Screenshot for index.php page



Shares List

#	Name	Visible?	Public?	Permissions	Action
1	Raspberry Pi Webserver	YES	NO	Read/Write	Details
2	Samba CONF Folder	YES	NO	Read/Write	Details
3	Mounted Media	YES	NO	Read/Write	Details
4	test-share-R	YES	NO	Read-Only	Details
5	test	YES	NO	Read-Only	Details

Figure 6: Screenshot for server-shares.php page, tab shares info.

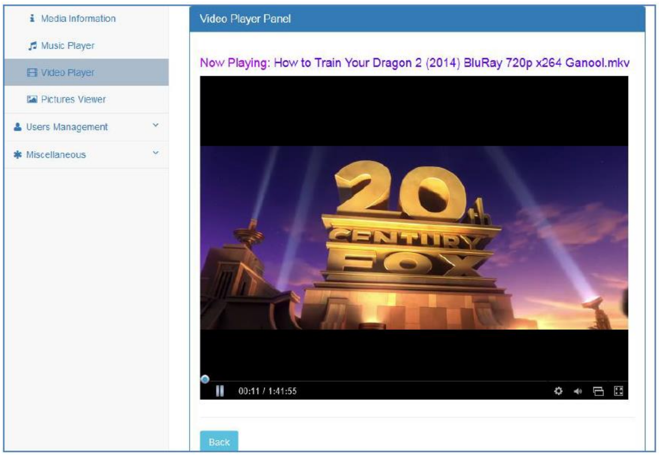


Films / Movies List

ID	File Name	Ext
1	How to Train Your Dragon 2 (2014) BluRay 720p x264 Ganool	mkv

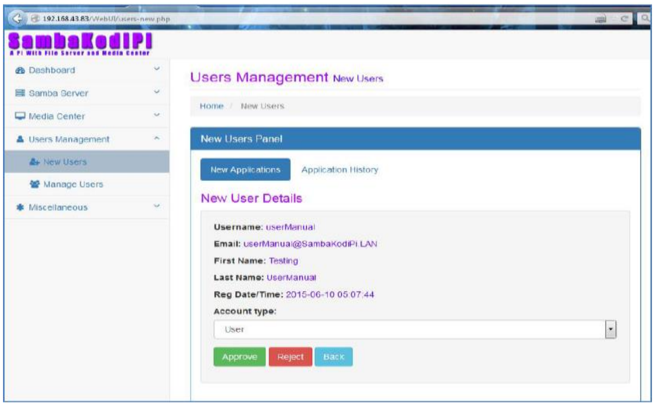
Showing 1 to 1 of 1 entries

Figure 7: Screenshot for media-video.php page.



Now Playing: How to Train Your Dragon 2 (2014) BluRay 720p x264 Ganool.mkv

Figure 8: Web video player supports full-screen display



New User Details

Username: userManual

Email: userManual@SambaKodiPi.LAN

First Name: Testing

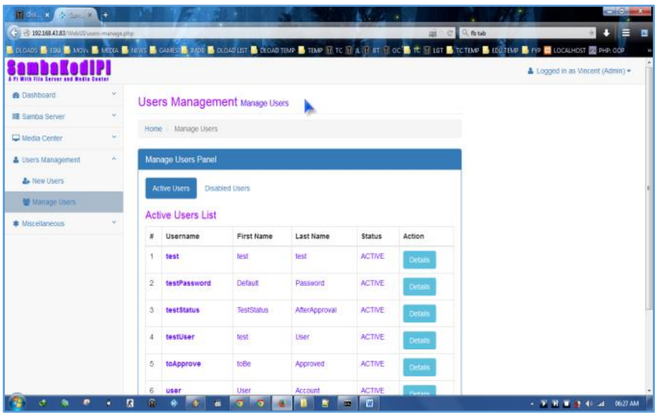
Last Name: UserManual

Reg Date/Time: 2015-06-10 05:07:44

Account type:

User

Figure 9: New Users page can approve or reject new user applications



Active Users List

#	Username	First Name	Last Name	Status	Action
1	test	test	test	ACTIVE	Details
2	testPassword	Default	Password	ACTIVE	Details
3	testStatus	TestStatus	AfterApproval	ACTIVE	Details
4	testUser	test	User	ACTIVE	Details
5	toApprove	toBe	Approved	ACTIVE	Details
6	user	User	Account	ACTIVE	Details

Figure 10: Screenshot for users-manage.php page (active users tab).

E. User-System Interaction

User-system interaction refers to how users interact with the system as a whole in order to use its functions. The proposed user-system interaction for each function is as shown in Figure 11 – Figure 14 below:

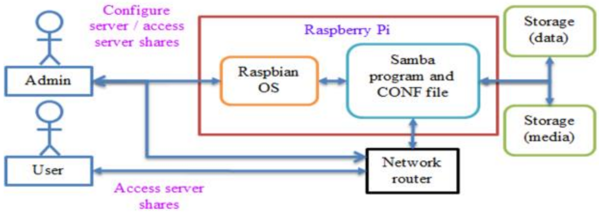


Figure 11: The interaction for file server function (direct)

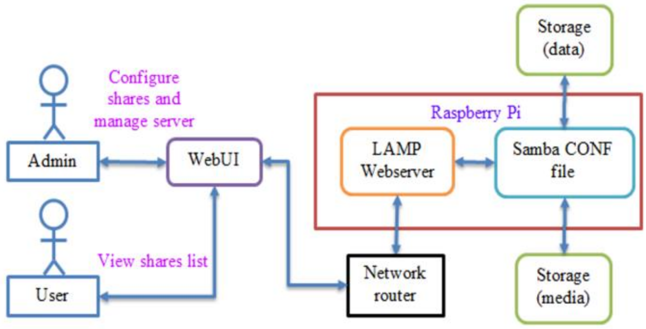


Figure 12: The interaction for WebUI file server function

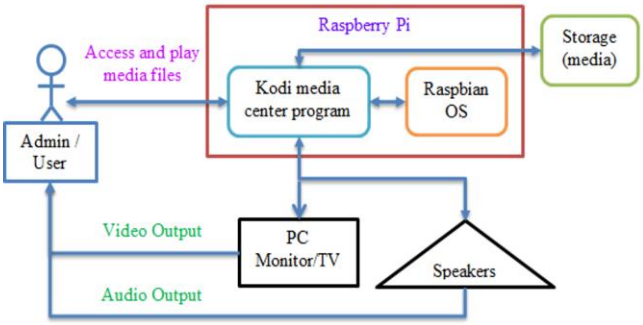


Figure 13: The interaction for media center function (Kodi)

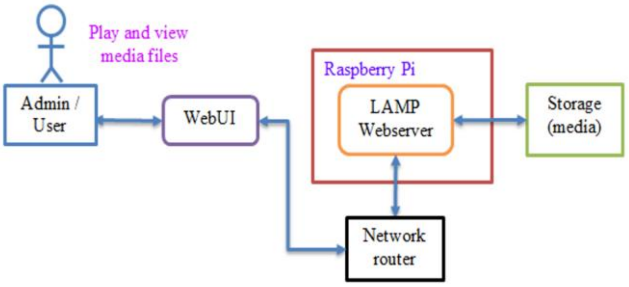


Figure 14: The interaction for WebUI media center function

V. Implementation and Testing

The functionalities of the system were implemented using iterative and incremental development methodology.

Table 6
The six development phases for this system

Phase	Main activities
Development phase 1	Preparation of the Raspberry Pi device with its operating system, essential programs and drivers.
Development phase 2	Implementation of file server capability. Will be using Samba file server software.
Development phase 3	Implementation of media center capability. Will be using XBMC media center software.
Development phase 4	Implementation of webserver capability. Will be using LAMP webserver component.
Development phase 5	Implementation of system interface (web-based interface) mainly for the file server function and secondary media center function.
Development phase 6	System integration and validation. Involves the process of integrating all the components and validate their functionalities.

In order to develop the system, its development was divided into six phases, which were done chronologically.

The main activities for each phase are as described in Table 6 above. For the development of WebUI in phase 5, a bootstrap theme website was used as base template. A logo for the WebUI was created and it was decided at that time to name the system SambaKodiPi, in honour of the two main programs used and the Pi device as its controller. Each of the WebUI functions was added iteratively and incrementally. The login function to the website and other base operations was implemented using Dreamweaver. Modifications were made to the PHP base operation codes, resulted in 28 base functions, stored in a function file named SKP-Base.php.

The effort to implement file server configuration function was focusing on parsing, and manipulating the content of Samba configuration file. This later resulted in 41 PHP functions created to handle the file server configuration operations. Those functions are stored in a function file named SKP-FS.php. Meanwhile, the effort to implement the secondary media center function on the WebUI was focusing on scanning directory for media files and filtering them for specific media files based on their extensions. This later resulted in 9 PHP functions that were created to handle the WebUI media center function, stored in another function file called SKP-MC.php.

Once the system functions were fully implemented, the separate components were then integrated and their functionalities and interaction were validated in phase 6. Once this is done, the system as a whole was subjected to four levels of testing process namely Component testing, Integration testing, System testing and Acceptance testing. After the testing, it was determined that the system has passed all the test cases for each test levels.

The testing activities were done based on ISTQB's testing principles, using concept from the ISO 9126 standards as reference [6]. Thus, the system testing for this system covers four main test levels:

i. Component testing;
ii. Integration testing;
iii. System testing; and
iv. Acceptance testing.

Each test level may consist of several sub-testing levels. Each level consists of description about the procedures on how the testing activities were done along with their results. Testing activities involves both static and dynamic testing for the validation and verification of the final system. Overall testing process focuses on measuring the quality of the system in terms of both its functional and non-functional requirements.

During the testing, some functions were found to be incorrect and they were fixed and re-tested again until they are working as intended. After the testing, all PHP functions for base, file server and media center operations were verified to be coded correctly in terms of structure, parameters, and function calling and thus verified to be working correctly. The results of few of these testing are as shown in Table 7, Table 8 and Table 9.

Table 7
Unit testing results for system components and devices.

Name	Type	Model/ Version	Condition	Functionality
Raspberry Pi	Device	Model B+	OK	OK
Modem/Homer	Device	Speedtouch585 v6	OK	OK
PC Monitor	Device	Acer x193HQ	OK	OK
Stereo speakers	Device	Logitech Z103	OK	OK
Laptop	Device	Acer Aspire 4750G	OK	OK
Keyboard	Device	Logitech K120	OK	OK
Mouse	Device	Logitech MISS	OK	OK
Laptop	Device	Acer 4750G	OK	OK
Smartphone	Device	Sony Xiscria E Sony Xparia Z1	OK	OK
IJSB Wive	Device	DT G2 BOB	OK	OK
External LIDD	Device	WD 500GB	OK	OK
Raspblan	Software	3.12.22+	OK	OK
Kodi	Software	14.10	OK	OK
Samba	Software	3.6.6	OK	OK
GAdmin-Samba	Software	0.2.9	OK	OK
Apache	Software	2.2.22 (Debian)	OK	OK
MySQL	Software	14.14, dist 5.5.41	OK	OK
PHP5	Software	5.4.39	OK	OK
Drecmweaver	Software	CS5.5 vl 1.5	OK	OK
Notepad++ (Win)	Software	6.5.5	OK	OK
Geany (Linux)	Software	1.22	OK	OK
DiffMerge	Software	4.2.0	OK	OK
WinRAR	Software	3.5.1	OK	OK
Mozilla Firefox	Software	36	OK	OK
Google Chrome	Software	41.0.2272.101 m	OK	OK
LIDMI cable	Cable	Standard HDMI	OK	OK
LAN cables	Cable	Standard LAN	OK	OK
HDMI to DVI	Adapter	ILM adapter	OK	OK
VGA rto VGA	Adopter	M-M adapter	OK	OK

Table 8
Unit testing results for system components and devices.

No	Page Name	Structure	Correctness	Functionality
1	index.php	OK	OK	OK
2	login.php	OK	OK	OK
3	register.php	OK	OK	OK
4	about.php	OK	OK	OK
5	contact.php	OK	OK	OK
6	home.php	OK	OK	OK
7	system.php	OK	OK	OK
8	system-config.php	OK	OK	OK
9	server.php	OK	OK	OK
10	server-shares.php	OK	OK	OK
11	server-config.php	OK	OK	OK
12	media.php	OK	OK	OK
13	media-music.php	OK	OK	OK
14	media-video.php	OK	OK	OK
15	media-pictures.pbp	OK	OK	OK
16	users-new.php	OK	OK	OK
17	users-manage.pbp	OK	OK	OK
18	comments.php	OK	OK	OK
19	comments-view.php	OK	OK	OK
20	profile.php	OK	OK	OK
21	account.php	OK	OK	OK
22	error-access.php	OK	OK	OK
23	error-permission.pbp	OK	OK	OK
24	index.html	OK	OK	OK
25	SambaKodiPi.css	OK	OK	OK
26	SambaKodiPi.js	OK	OK	OK

Table 9
Unit testing results for system components and devices.

Test ID	Test Case	Test Steps	Expected Result	Actual Result	Status	Remarks
1	Test user acceptance of system functionality	Testing user acceptance on file server function	Depends on user expectation	User is fully satisfied with the file server function	Pass	
		Testing user acceptance on the primary media center function (using Kodi program)	Depends on user expectation	User is fully satisfied with the Kodi program as the primary media center program	Pass	
		Testing user acceptance on the WebUI file server configuration functions	Depends on user expectation	User is fully satisfied with the WebUI file server configuration functions as they are working correctly	Pass	Custom parser function can still be developed to parse smb.conf
		Testing user acceptance on the WebUI media center functions	Depends on user expectation	User is fully satisfied with the WebUI media center functions as they are capable to play media files on web browser	Pass	Even though format support is limited, it may improve in the future
		Testing user acceptance on the WebUI management functions	Depends on user expectation	User is fully satisfied with WebUI management functions as they are working correctly	Pass	
2	Test user acceptance of WebUI design	Testing user acceptance on the WebUI design	Depends on user expectation	User is fully satisfied with the responsive WebUI design	Pass	

VI. Conclusion

The final system, which nicknamed as SambaKodiPi is essentially an integrated system of five main integrated components, namely: the RPi component, the Samba file server component, the Kodi media center component, the LAMP webserver component and its Web-based User Interface (WebUI) component. All of these components have passed component and integration testing and proved to be working together.

The system also passed system testing, which tested its functionality and reliability and finally acceptance testing by its intended user. Granted that the system may still be quite simple but there will always be rooms and time for improvements and addition of more functions in the future. And it is now properly equipped with all the required capability to realise all its original intent and purposes. Thus, the aimed to develop this system, as a whole, is a definitely a success.

Acknowledgment

The authors thank the anonymous reviewers of the previous version of this paper for their comments and suggestions. Acknowledgement also for the Faculty Computer Science and Information Technology (FCSIT), UNIMAS for providing the tools and devices for the project

References

[1] “Raspberry Pi FAQs,” 2016 [Online]. Available at http://www.raspberrypi.org/help/faqs/. [Accessed 9 Jan. 2016].

[2] R. Bruce, J. Brock and S. Reiser, “Make space for the Pi,” in *Proc. of the Southeast Conf.*, Fort Lauderdale, FL, USA, 2015.

[3] V. Beal, “Server,” [Online]. Available at http://www.webopedia.com/ TERM/S/server.html. [Accessed 9 Jan. 2015].

[4] “HTPC,” 2015 [Online]. Available at http://www.pcmag.com/ encyclopedia/term/44500/htpc. [Accessed 10 Jan. 2015].

[5] “Non-dedicated file server.,” 2015 [Online]. Available at http://www. thefreedictionary.com/non-dedicated+file+server. [Accessed: 12 Jan. 2015].

[6] I. Sommerville, *Software Engineering*, Boston, US: Addison-Wesley, 2011.

[7] M.-G. A. O. Programme, “Chapter 2: Testing throughout the Software Life Cycle,” in *Software Life Cycle*, Software Testing Foundations Certified Tester slide v1.0, 2011.

[8] G. Hanson, “What is a file server?,” 2015 [Online]. Available at http://www. wisegeek.com/what-is-a-file-server.htm. [Accessed: 1 Dec. 2015].

[9] “What is Raspberry Pi?,” 2016 [Online]. Available at http://www.raspberrypi. org/help/what-is-a-raspberry-pi/. [Accessed: 10 Jan. 2016].

[10] J. &. P. K. Fitzpatrick, “Which Media Center Is Right for You: Boxee, XBMC, and,” 2015. [Online]. Available at http://lifehacker. com/5462275/which- media-center-is-right-for-you-boxee-xbmc-and-windows-media-center-compared. [Accessed: 1 Dec. 2015].

[11] J. Layton, “How Media-center PCs Work,” 2015. [Online]. Available at http:// computer.howstuffworks.com/media-center-pc.htm. [Accessed: 12 Oct. 2015].