CONTENTS

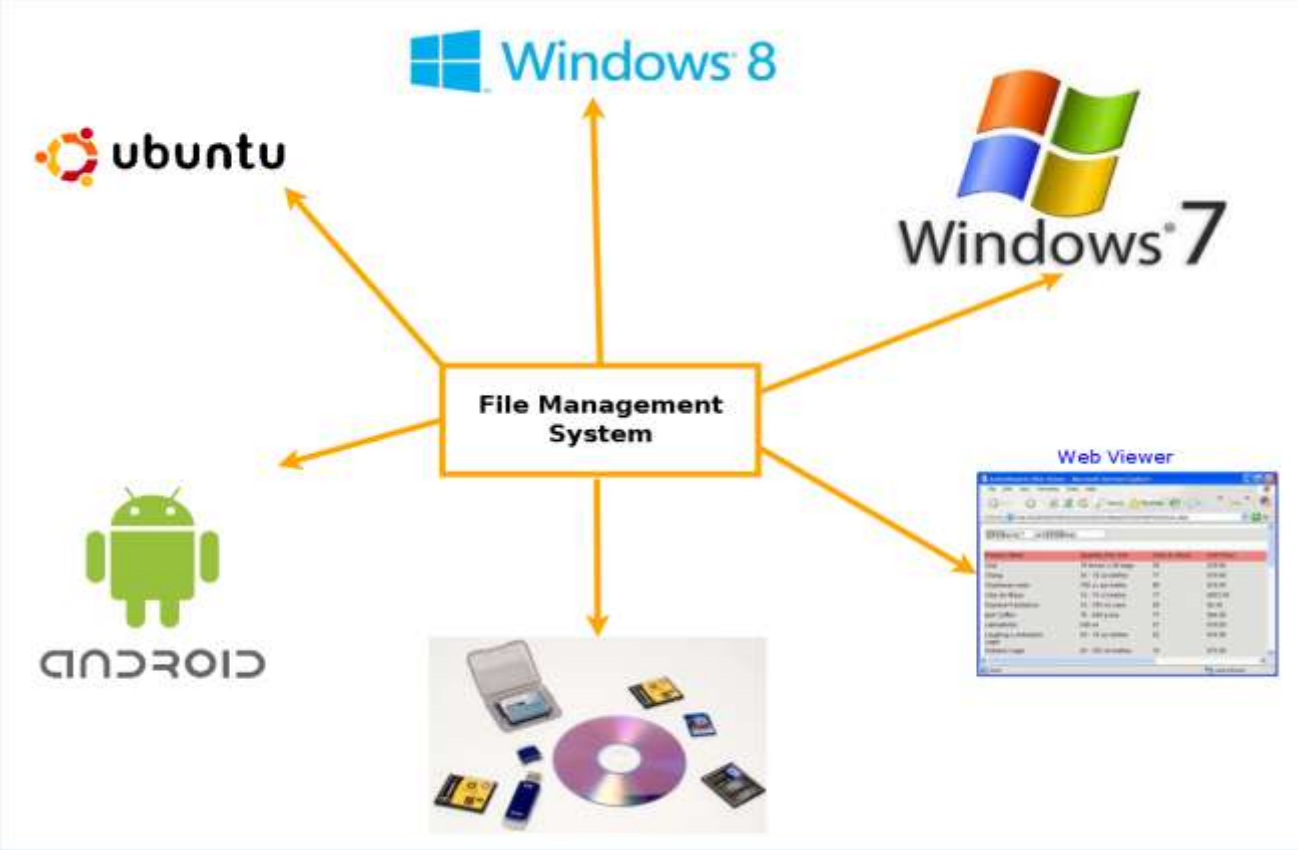
INTRODUCTION& OBJECTIVE

INTRODUCTION

External storage device is an eminently useful tool for all computer users. The tiny devices fit in your pocket, but can store and transport gigabytes of your data. A USB drive is useful for moving files and folders from one computer to another, as well as serving as a backup device for your important files.

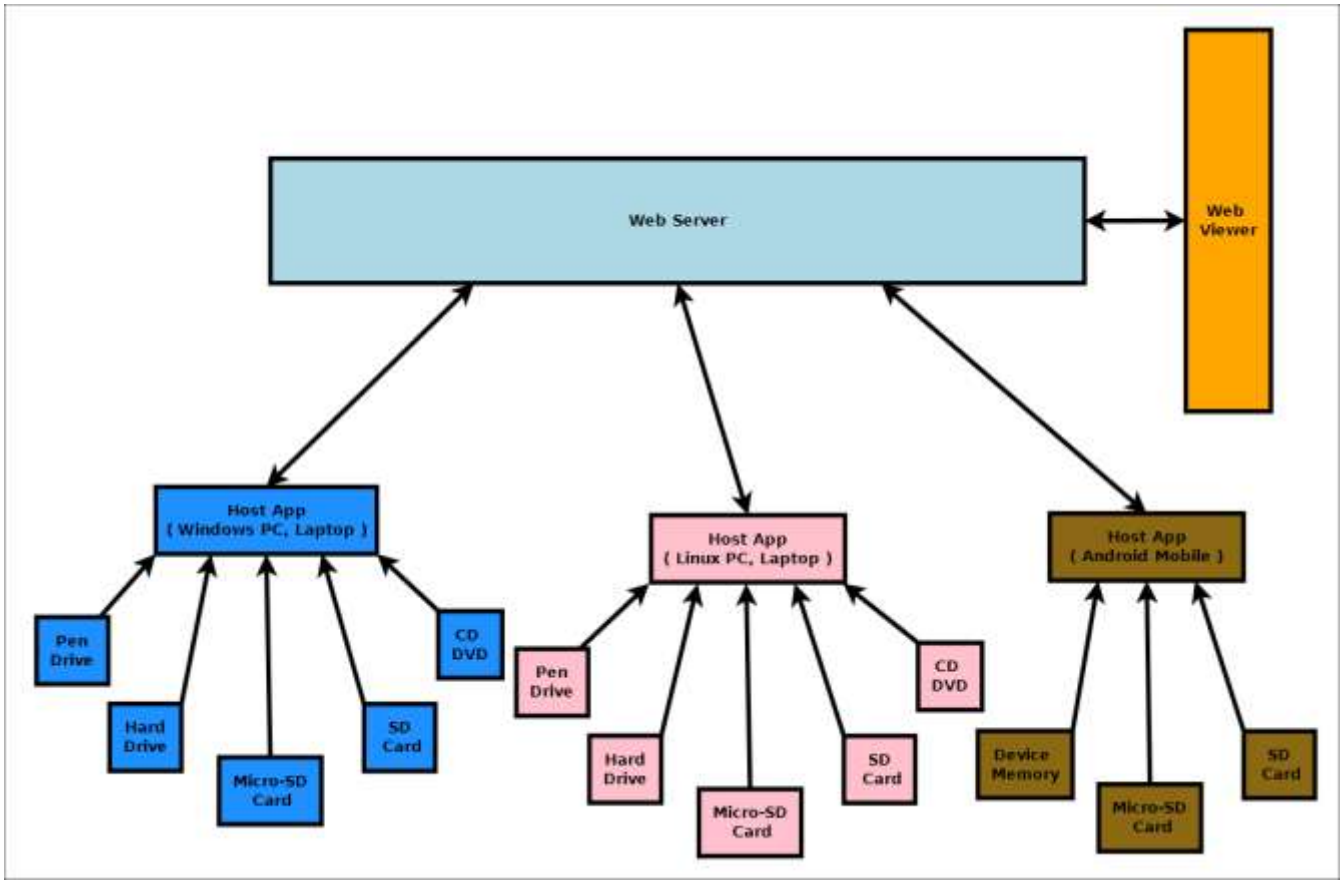
Let's count how many storage devices anyone owns now. It will be at least more than ten. Do you know which of your device contains what content? The answer is mostly no. We need a centralized solution for storing information about the files we have. Cloud based storage is costly. Why can't we turn our own storages devices into a connected personal cloud system? So we need to store information about the files we have. So we can search easily the file using the file information and access them whenever we need.

File Management System (will be referred as FMS in this document) is standalone software for managing files in various storage devices.



The main components of this project are:

- In this project there will be a web based server which will store information about files stored in user's machine and storage devices.
- The clients/host apps (windows, Linux, android) will analyze the file systems and upload the information to web server.
- There will be a web based viewer for viewing the information.



OBJECTIVE

Modern operating systems treat the storage devices as "Plug and Play" hardware, recognizing them as soon as they are plugged into a port and making them available for immediate use. Clients will work as host app which can detect storage devices like micro SD, SD card, pen drive and hard drives. Whenever storage devices are connected to host apps, host app will scan the file system of the storage device silently and update the file information in the server along with storage device unique Id.

The main features of the file management system are:

- Keep track of files stored in various storage device owned by the user in a single place.
- Easy to search for a particular file in the fms viewer and then retrieve file from specific device.
- Detect duplicate file stored in various devices.
- Stream file from any of the connected devices via web interface on demand
- Searching and sorting file information
- Share files via social networking sites and cloud services
- Backup important files

SYSTEM ANALYSIS

IDENTIFICATION OF NEED:

We use different types of storage medium for different types of devices. Here are some Advantages of external storage:

- External storage devices provide additional storage other than that available in computer.
- Data can be transported easily from one place to another.
- It is useful to store software and data that is not needed frequently.
- External storage also works as data backup.
- This back up may prove useful at times such as fire or theft because important data is not lost.

We need a centralized solution for storing information about the files we have. Why can't we turn our own storages devices into a connected personal cloud system?

The existing system allows user to share their personal data on the web through various cloud based applications but they do not allow us to manage all our devices' storages in one place. The existing ones, on the other hand, are not as user friendly as well. Sometimes they create confusion while logging in to the web server from other devices.

PRELIMINARY INVESTIGATION:

Using our application, the users can access information about the data of all his storage devices from one place with a very organized way. They can store and synchronize data of their desktop, laptop and/or smart phone. The main intension is to utilize personal storage devices, avoid duplicity of files in several devices, easy to find which storage device has the desired data.

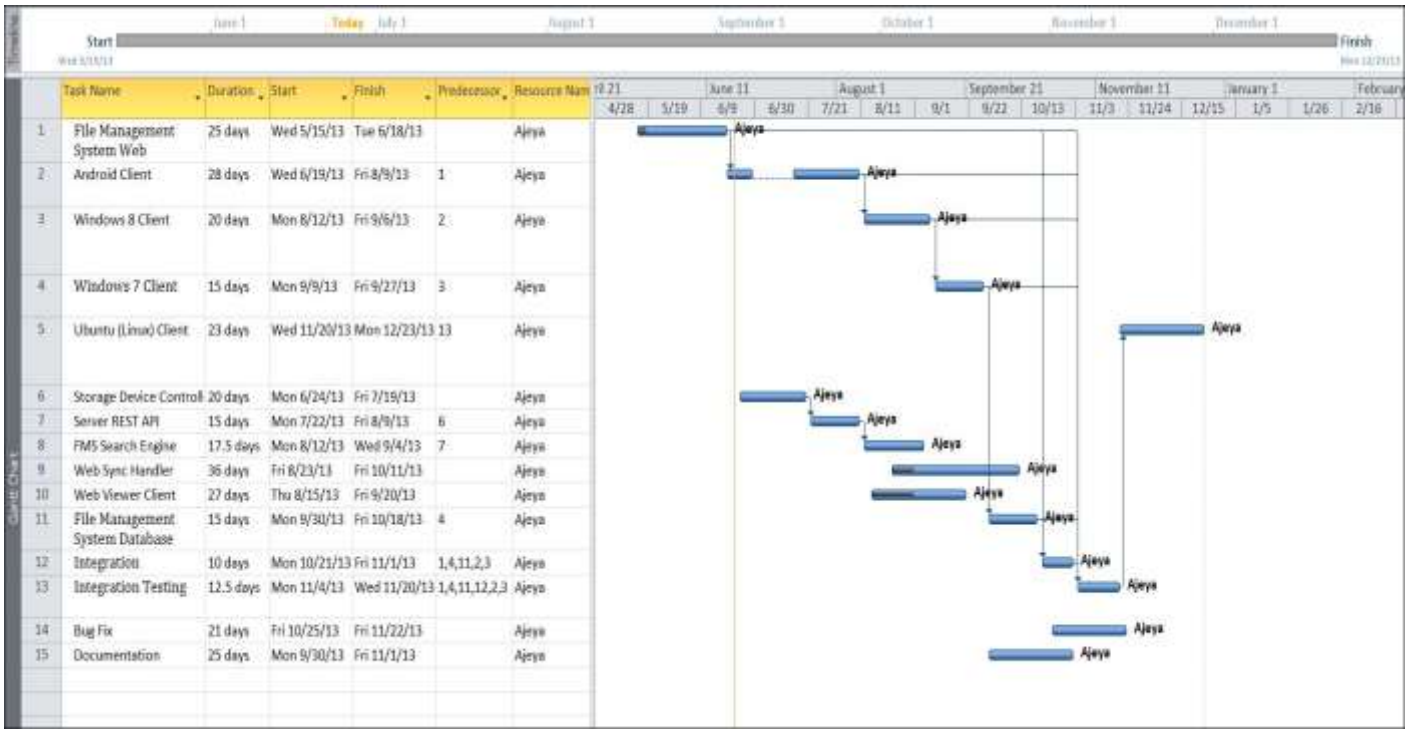
FEASIBILITY STUDY:

As per our requirement we have implemented several clients (Host App) and server as given below:

1. Windows Host App
2. Ubuntu Linux Host App
3. Android Host App
4. FMS Server / Controller
5. Web Viewer

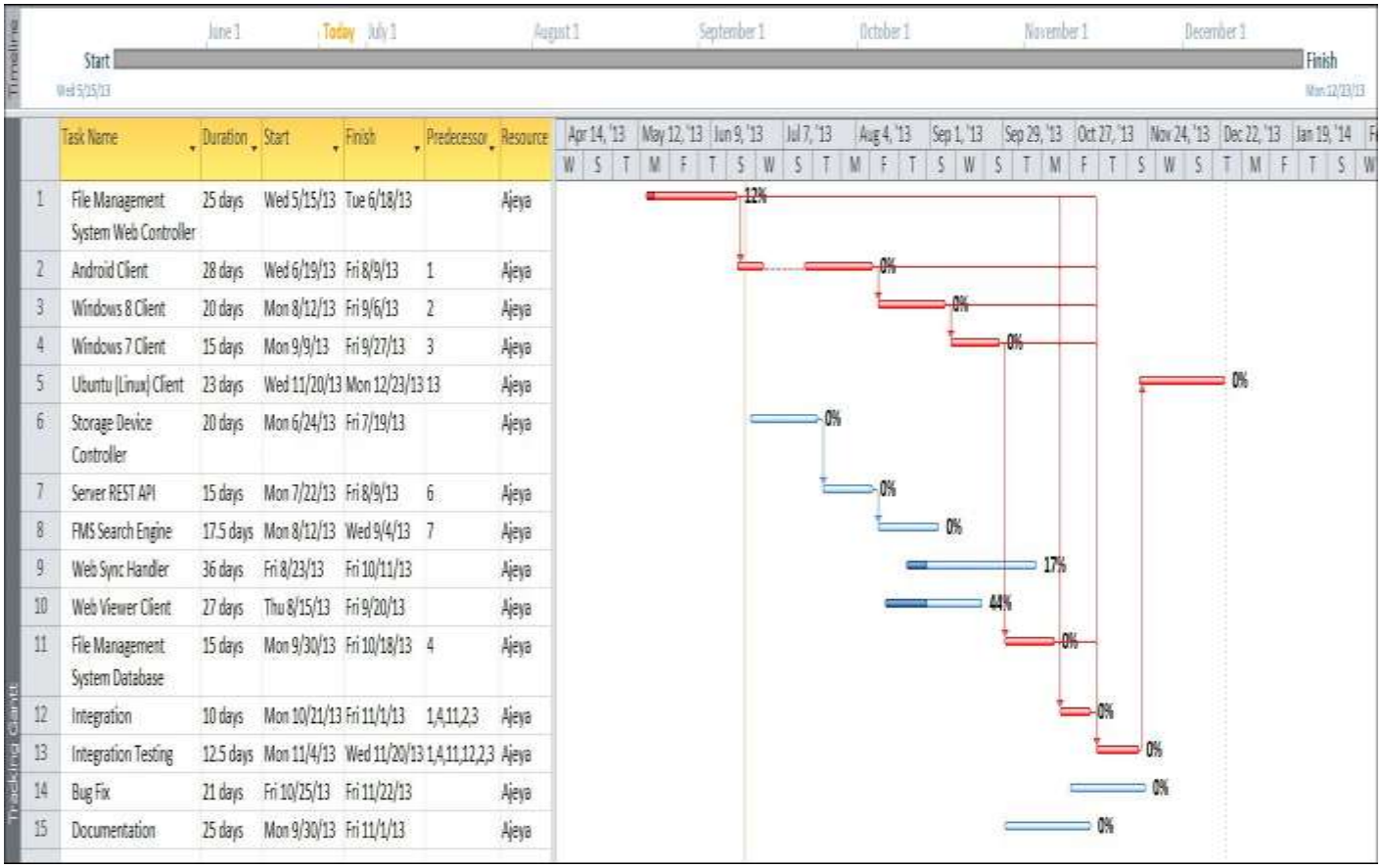
PROJECT PLANNING & SCHEDULING:

GANTT CHART



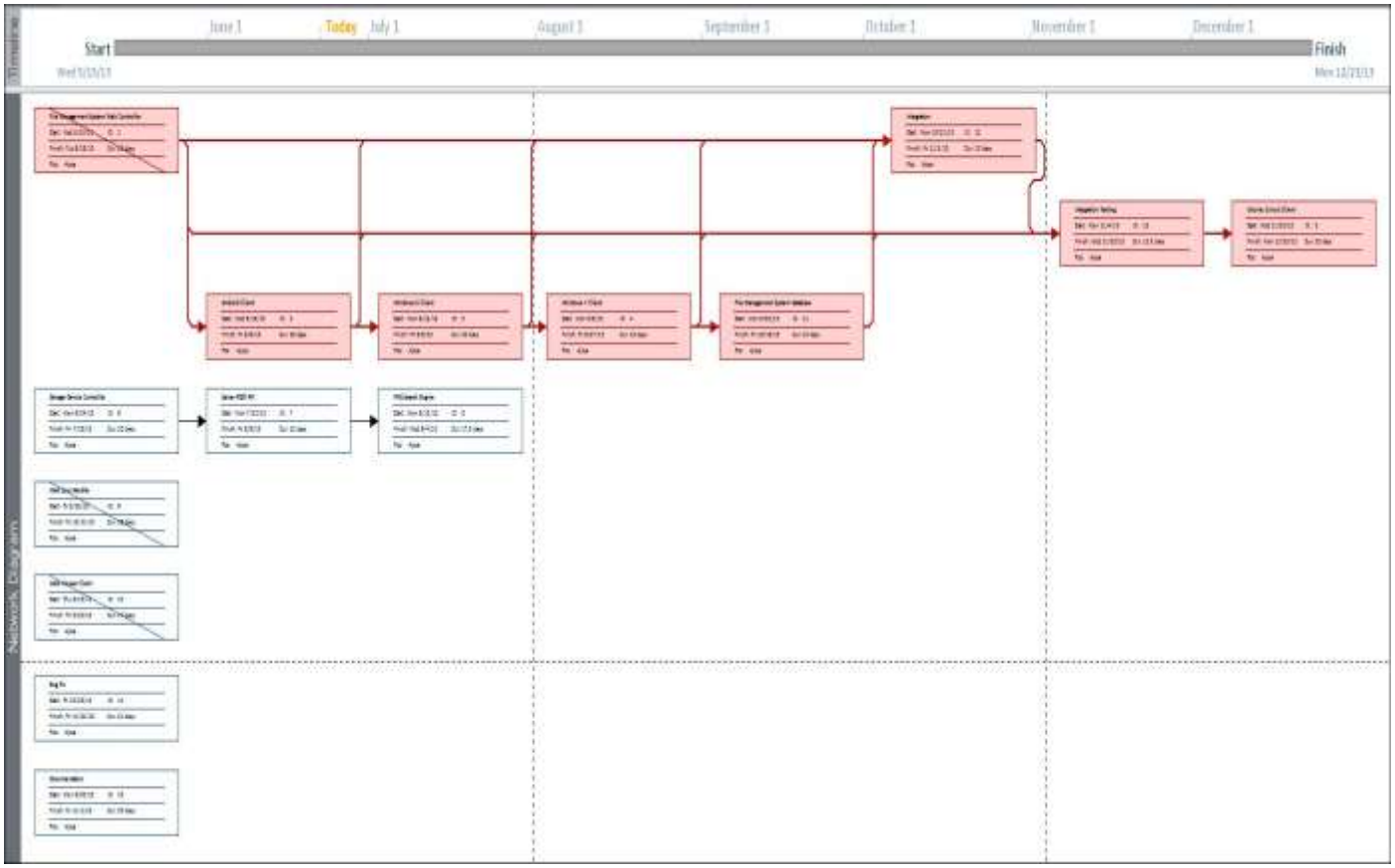
	Task Name	Duration	Start	Finish	Predecessor	Resource Name
1	File Management System Web	25 days	Wed 5/15/13	Tue 6/18/13		Ajeya
2	Android Client	28 days	Wed 6/19/13	Fri 8/9/13	1	Ajeya
3	Windows 8 Client	20 days	Mon 8/12/13	Fri 9/6/13	2	Ajeya
4	Windows 7 Client	15 days	Mon 9/9/13	Fri 9/27/13	3	Ajeya
5	Ubuntu (Linux) Client	23 days	Wed 11/20/13	Mon 12/23/13	13	Ajeya
6	Storage Device Controll	20 days	Mon 6/24/13	Fri 7/19/13		Ajeya
7	Server REST API	15 days	Mon 7/22/13	Fri 8/9/13	6	Ajeya
8	FMS Search Engine	17.5 days	Mon 8/12/13	Wed 9/4/13	7	Ajeya
9	Web Sync Handler	36 days	Fri 8/23/13	Fri 10/11/13		Ajeya
10	Web Viewer Client	27 days	Thu 8/15/13	Fri 9/20/13		Ajeya
11	File Management System Database	15 days	Mon 9/30/13	Fri 10/18/13	4	Ajeya
12	Integration	10 days	Mon 10/21/13	Fri 11/1/13	1,4,11,2,3	Ajeya
13	Integration Testing	12.5 days	Mon 11/4/13	Wed 11/20/13	1,4,11,12,2,3	Ajeya
14	Bug Fix	21 days	Fri 10/25/13	Fri 11/22/13		Ajeya
15	Documentation	25 days	Mon 9/30/13	Fri 11/1/13		Ajeya

TRACKING GANTT



	Task Name	Duration	Start	Finish	Predecessor	Resource
1	File Management System Web Controller	25 days	Wed 5/15/13	Tue 6/18/13		Ajeya
2	Android Client	28 days	Wed 6/19/13	Fri 8/9/13	1	Ajeya
3	Windows 8 Client	20 days	Mon 8/12/13	Fri 9/6/13	2	Ajeya
4	Windows 7 Client	15 days	Mon 9/9/13	Fri 9/27/13	3	Ajeya
5	Ubuntu (Linux) Client	23 days	Wed 11/20/13	Mon 12/23/13	13	Ajeya
6	Storage Device Controller	20 days	Mon 6/24/13	Fri 7/19/13		Ajeya
7	Server REST API	15 days	Mon 7/22/13	Fri 8/9/13	6	Ajeya
8	FMS Search Engine	17.5 days	Mon 8/12/13	Wed 9/4/13	7	Ajeya
9	Web Sync Handler	36 days	Fri 8/23/13	Fri 10/11/13		Ajeya
10	Web Viewer Client	27 days	Thu 8/15/13	Fri 9/20/13		Ajeya
11	File Management System Database	15 days	Mon 9/30/13	Fri 10/18/13	4	Ajeya
12	Integration	10 days	Mon 10/21/13	Fri 11/1/13	1,4,11,2,3	Ajeya
13	Integration Testing	12.5 days	Mon 11/4/13	Wed 11/20/13	1,4,11,12,2,3	Ajeya
14	Bug Fix	21 days	Fri 10/25/13	Fri 11/22/13		Ajeya
15	Documentation	25 days	Mon 9/30/13	Fri 11/1/13		Ajeya

PERT CHART (NETWORK DIAGRAM)



ESTIMATION

Timeline

Start Wed 5/15/13 | Today | September 1 | November 1 | Finish Mon 12/23/13

Task Usage

	Task Name	Work	Duration	Start	Finish
1	File Management System Web Controlle	200 hrs	25 days	Wed 5/15/13	Tue 6/18/13
	Ajeya	*200 hrs*		*Wed 5/15/13*	*Tue 6/18/13*
2	Android Client	224 hrs	28 days	Wed 6/19/13	Fri 8/9/13
	Ajeya	*224 hrs*		*Wed 6/19/13*	*Fri 8/9/13*
3	Windows 8 Client	160 hrs	20 days	Mon 8/12/13	Fri 9/6/13
	Ajeya	*160 hrs*		*Mon 8/12/13*	*Fri 9/6/13*
4	Windows 7 Client	120 hrs	15 days	Mon 9/9/13	Fri 9/27/13
	Ajeya	*120 hrs*		*Mon 9/9/13*	*Fri 9/27/13*
5	Ubuntu (Linux) Client	184 hrs	23 days	Wed 11/20/13	Mon 12/23/13
	Ajeya	*184 hrs*		*Wed 11/20/13*	*Mon 12/23/13*
6	Storage Device Controller	160 hrs	20 days	Mon 6/24/13	Fri 7/19/13
	Ajeya	*160 hrs*		*Mon 6/24/13*	*Fri 7/19/13*
7	Server REST API	120 hrs	15 days	Mon 7/22/13	Fri 8/9/13
	Ajeya	*120 hrs*		*Mon 7/22/13*	*Fri 8/9/13*
8	FMS Search Engine	140 hrs	17.5 days	Mon 8/12/13	Wed 9/4/13
	Ajeya	*140 hrs*		*Mon 8/12/13*	*Wed 9/4/13*
9	Web Sync Handler	288 hrs	36 days	Fri 8/23/13	Fri 10/11/13
	Ajeya	*288 hrs*		*Fri 8/23/13*	*Fri 10/11/13*
10	Web Viewer Client	216 hrs	27 days	Thu 8/15/13	Fri 9/20/13
	Ajeya	*216 hrs*		*Thu 8/15/13*	*Fri 9/20/13*
11	File Management System Database	120 hrs	15 days	Mon 9/30/13	Fri 10/18/13
	Ajeya	*120 hrs*		*Mon 9/30/13*	*Fri 10/18/13*
12	Integration	80 hrs	10 days	Mon 10/21/13	Fri 11/1/13
	Ajeya	*80 hrs*		*Mon 10/21/13*	*Fri 11/1/13*
13	Integration Testing	100 hrs	12.5 days	Mon 11/4/13	Wed 11/20/13
	Ajeya	*100 hrs*		*Mon 11/4/13*	*Wed 11/20/13*
14	Bug Fix	168 hrs	21 days	Fri 10/25/13	Fri 11/22/13
	Ajeya	*168 hrs*		*Fri 10/25/13*	*Fri 11/22/13*
15	Documentation	200 hrs	25 days	Mon 9/30/13	Fri 11/1/13
	Ajeya	*200 hrs*		*Mon 9/30/13*	*Fri 11/1/13*

SOFTWARE REQUIREMENT SPECIFICATIONS (SRS):

FUNCTIONAL REQUIREMENTS

ADD DEVICE

Introduction

The user can add their external storage device to file management system database.

Input

File management will take device ID.

Processing

The app will add the entire file attribute to file management system database.

Output

File management system database add the external storage device.

VIEW DEVICE

Introduction

The user can view information about their external storage device from file management system database.

Input

File management will recognize device ID and display external storage devices.

Processing

The FMS gathers all devices' ID and displays all external storage devices.

Output

The user can see all external storage devices.

BROWSE FILE INFORMATION

Introduction

The user can browse their file information using file management system.

Input

The required file name or any index keys

Processing

Get the information about the file and notify user that in which Device the file is located.

Output

The user can find all the files and folders from memory device.

SEARCH FILE

Introduction

Users can search file.

Input

Insert file name to search it.

Processing

Searches the file by file name and give it to the user.

Output

Display the information about required file.

SEARCH DUPLICATE FILE

Introduction

Users can search duplicate file.

Input

Insert file name or check the selected file's duplicate copy.

Processing

Searches for the same file name by using the file id.

Output

Display all duplicate file lists.

DELETE DUPLICATE FILE

Introduction

Users can delete duplicate file.

Input

Insert file name or check the selected file's duplicate copy.

Processing

Searches for the same file name by using the file id.

Output

Ask to delete duplicate copy from the user. Now user can delete the file.

SORT FILE INFORMATION

Introduction

Users can sort file information such as by Name, Size, Created time, last modified, Size.

Input

Instruction for sorting process according to user preference

Processing

Get the sorting instruction and sorting all the files according to the instruction.

Output

Get the entire sorted file list.

SHARE FILE

Introduction

Users can share any particular file in cloud storage or any website.

Input

File name, which is to be shared and give the link for the website or cloud storage.

Processing

Get the file by its name and shared it to the particular website or cloud storage.

Output

Get the file in cloud storage or particular website.

BACKUP FILE

Introduction

Users can back up their important file by coping in many devices.

Input

File name, which is to be backup and give destination of backup.

Processing

Get the file and copy it to the particular memory storage device.

Output

Get the same important file in multiple storage devices.

STREAM FILE

Introduction

Stream file from any of the connected devices via web interface on demand

Input

File name, which is to be stream

Processing

Get the file and stream the file via host app.

Output

Get the same important file in the web interface.

NON FUNCTIONAL REQUIREMENTS

- **Efficiency**:
  It will be efficient as it reduces manual labor and searching.
- **Backup**:
  Data could be stored to online storage.
- **Documentation**:
  The application will have users' manual pdf inside the help section.
- **Maintainability**:
  It is designed such a way that it can be maintained with minimal effort.
- **Performance**:
  The response time of file manager will be very fast. So it will be efficient enough to cater the user.
- **Privacy**:
  The data will be encrypted and the user data will not be shared with third party without proper authentication.
- **Looks & Feels**: Should have very attractive looks and feels to make the user happy even by opening it.
  ESCMS will use secure connection and enhanced security measures to protect data.
- **Usability**:
  - It will be very user friendly and usable by any person with minimal computer knowledge.

SOFTWARE ENGINEERING PARADIGM APPLIED

We have followed agile version of Model Driven Development (MDD). As the name implies, AMDD is the agile version of Model Driven Development (MDD). MDD is an approach to software development where extensive models are created before source code is written. A primary example of MDD is the Object Management Group (OMG)'s Model Driven Architecture (MDA) standard. With MDD a serial approach to development is often taken, MDD is quite popular with traditionalists, although as the RUP/EUP shows it is possible to take an iterative approach with MDD. The difference with AMDD is that instead of creating extensive models before writing source code you instead create agile models which are just barely good enough that drive your overall development efforts. AMDD is a

critical strategy for scaling agile software development beyond the small, co-located team approach that we saw during the first stage of agile adoption.

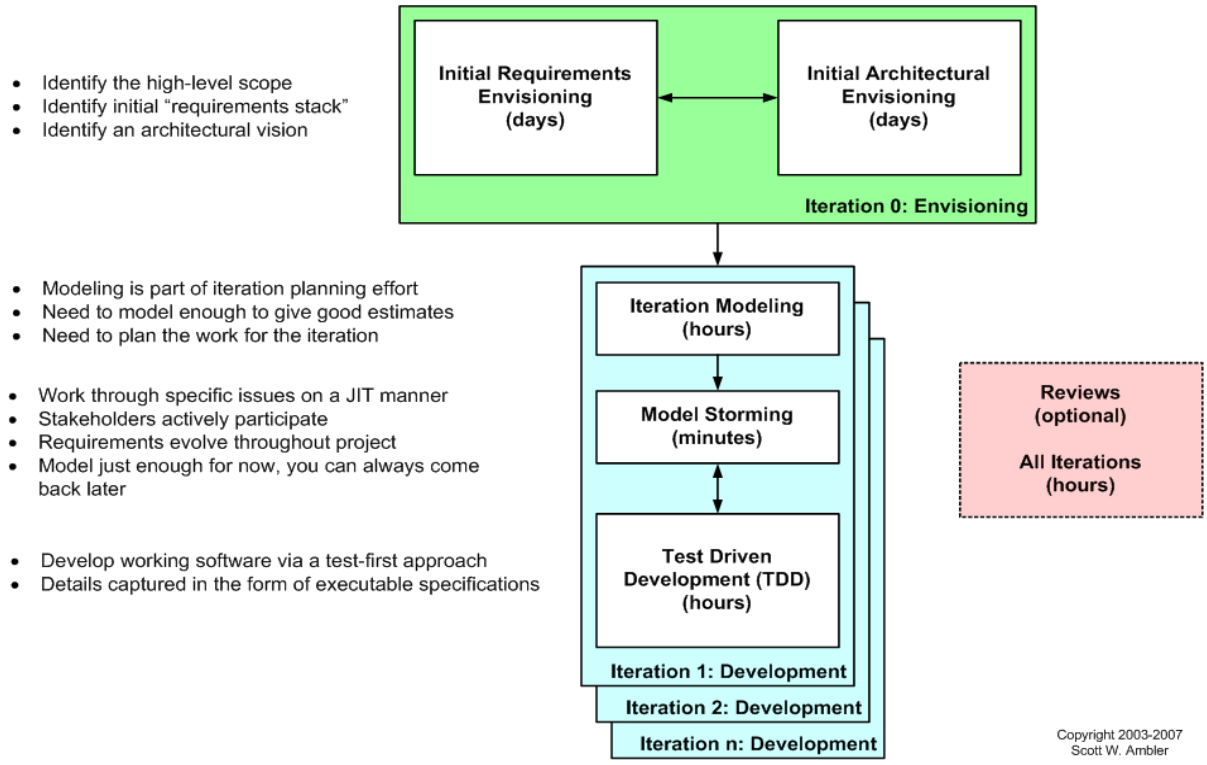


Figure 1: The AMDD lifecycle: Modeling activities throughout the lifecycle of a project

Above Figure depicts a high-level lifecycle for AMDD for the release of a system. First, let's start with how to read the diagram. Each box represents a development activity. The envisioning includes two main sub-activities, initial requirements envisioning and initial architecture envisioning. These are done during iteration 0, iteration being another term for cycle or sprint. "Iteration 0" is a common term for the first iteration before you start into development iterations, which are iterations one and beyond (for that release). The other activities – iteration modeling, model storming, reviews, and implementation – potentially occur during any iteration, including iteration 0. The time indicated in each box represents the length of an average session: perhaps you'll model for a few minutes then code for several hours. I'll discuss timing issues in more detail below..

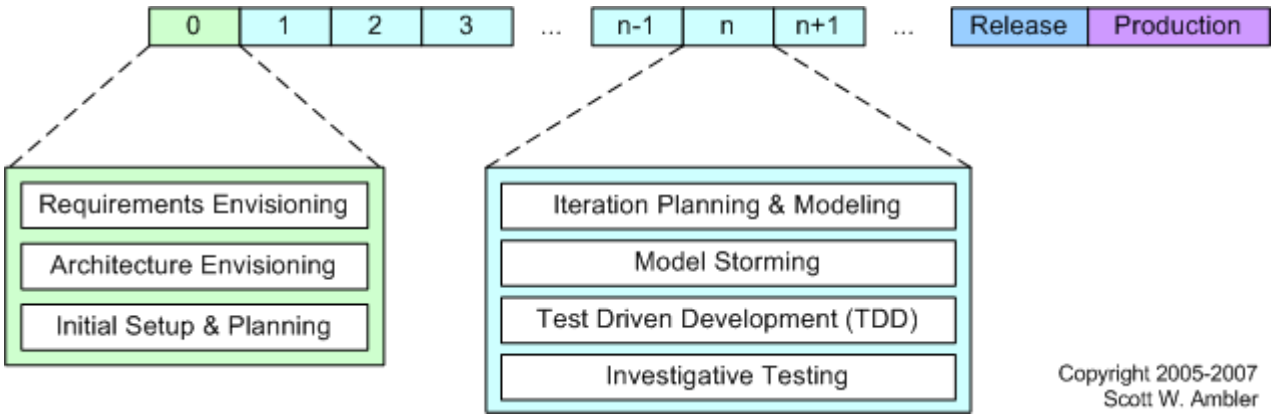


Figure 2AMDD Through the Agile Development Lifecycle.

Above Figure depicts how the AMDD activities fit into the various iterations of the agile software development lifecycle. It's simply another way to show that an agile project begins with some initial modelling and that modelling still occurs in each construction's iteration.

DATA MODELS

CONTEXT DIAGRAM

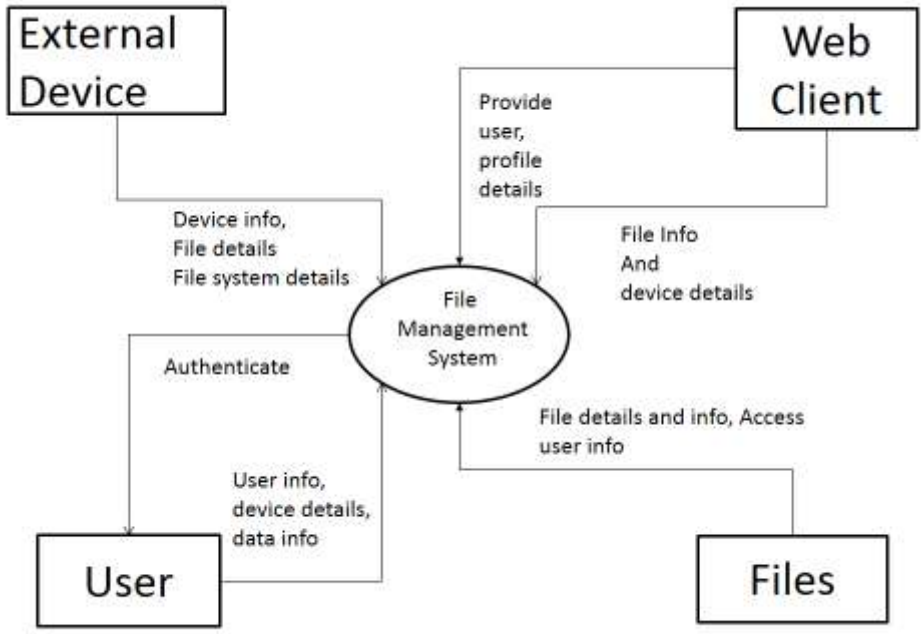


Figure 3:Context Diagram

DATA FLOW DIAGRAM

LEVEL 0 DFD

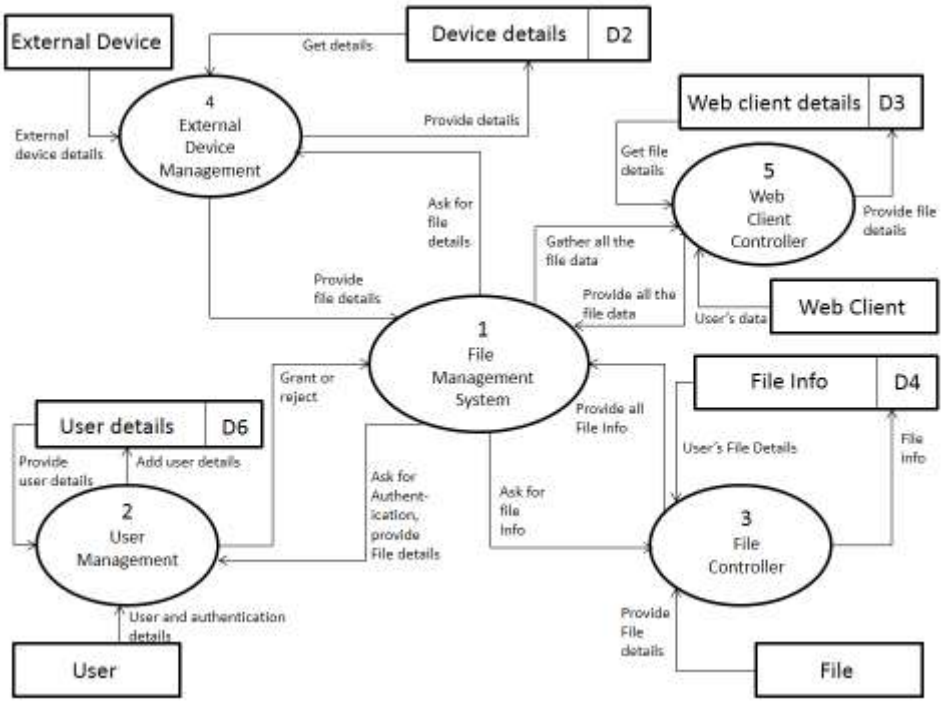


Figure 4:0 level DFD

LEVEL 1 DFD

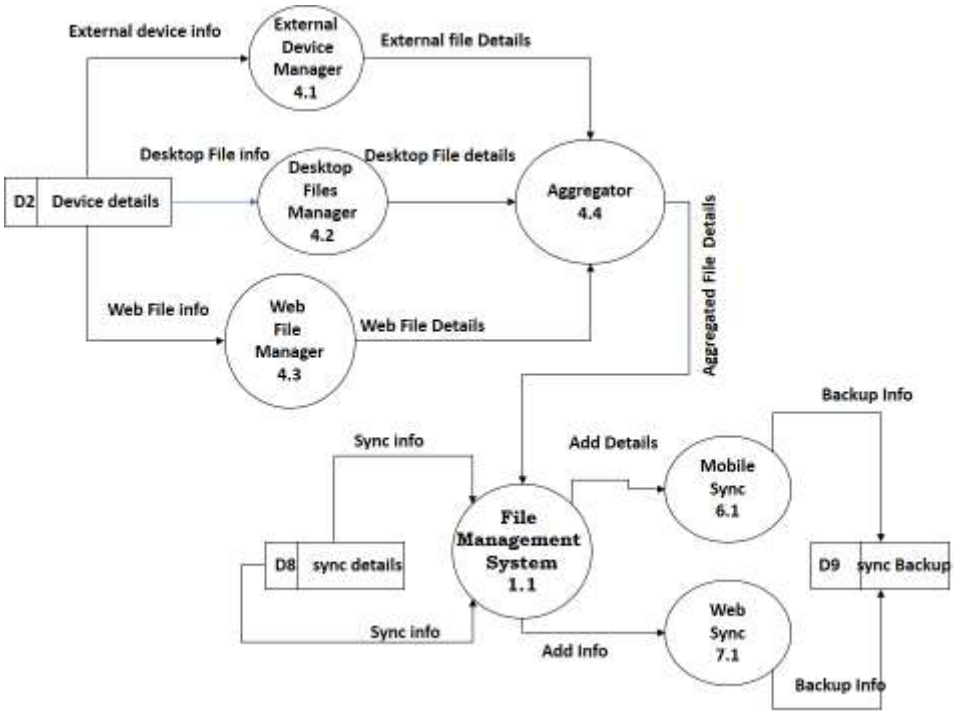


Figure 5:1 level DFD

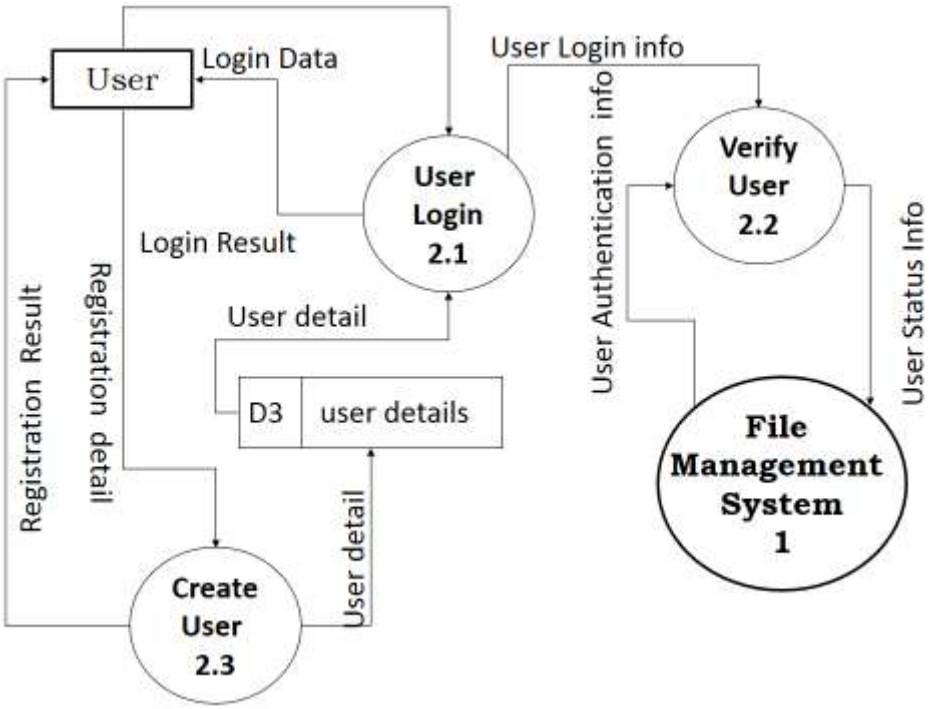


Figure 6:1 level DFD

LEVEL 2 DFD

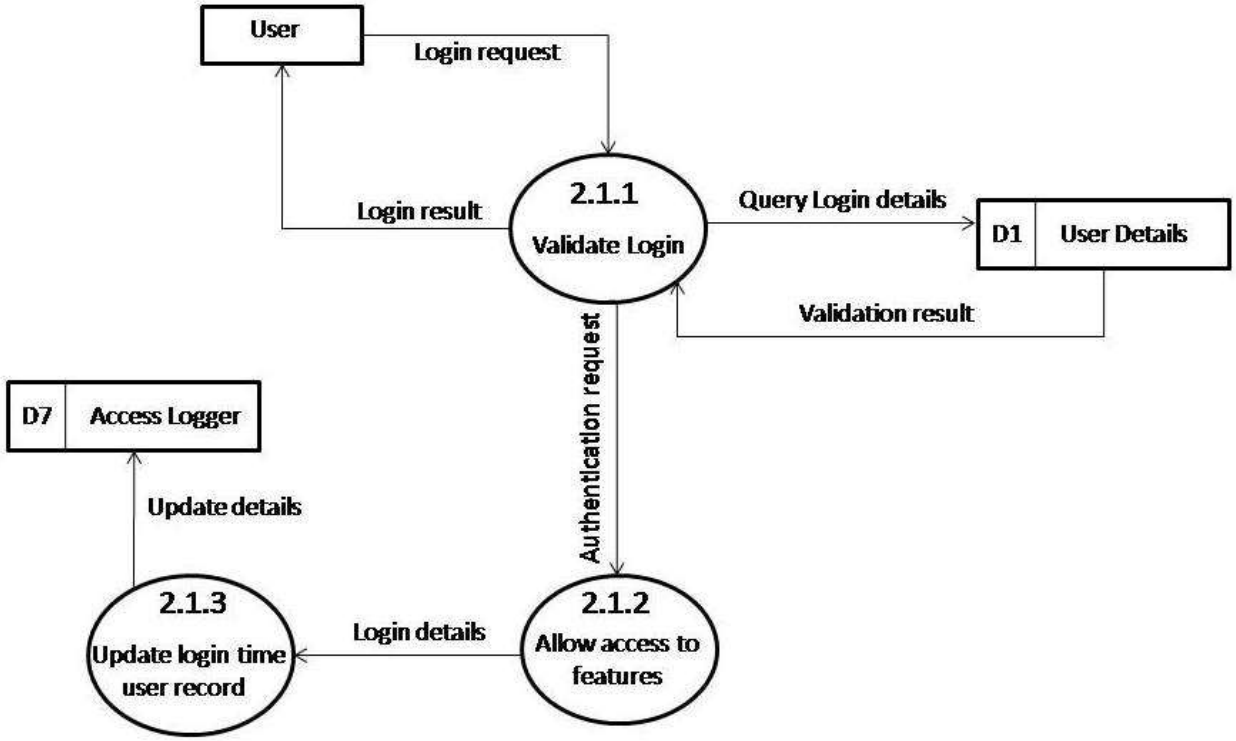
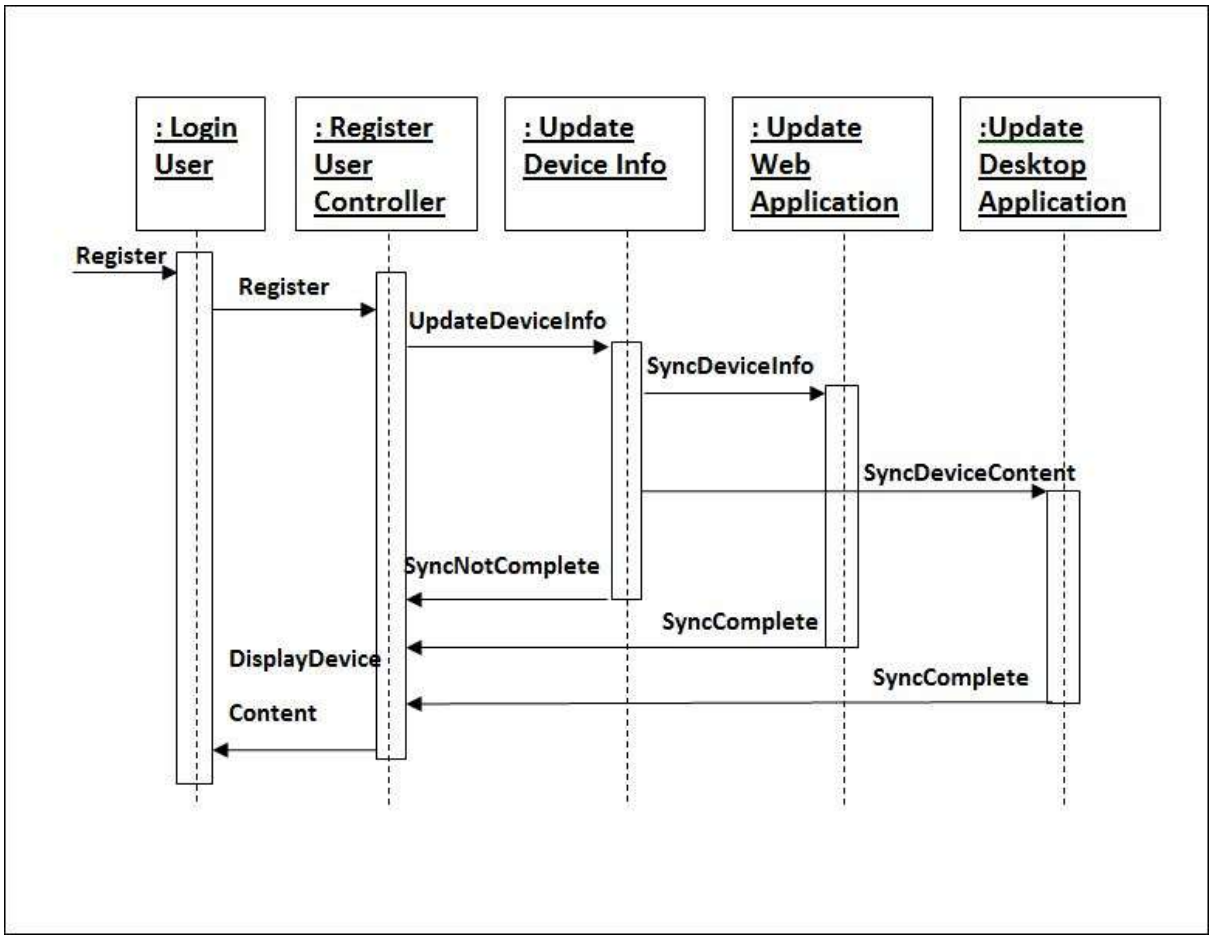
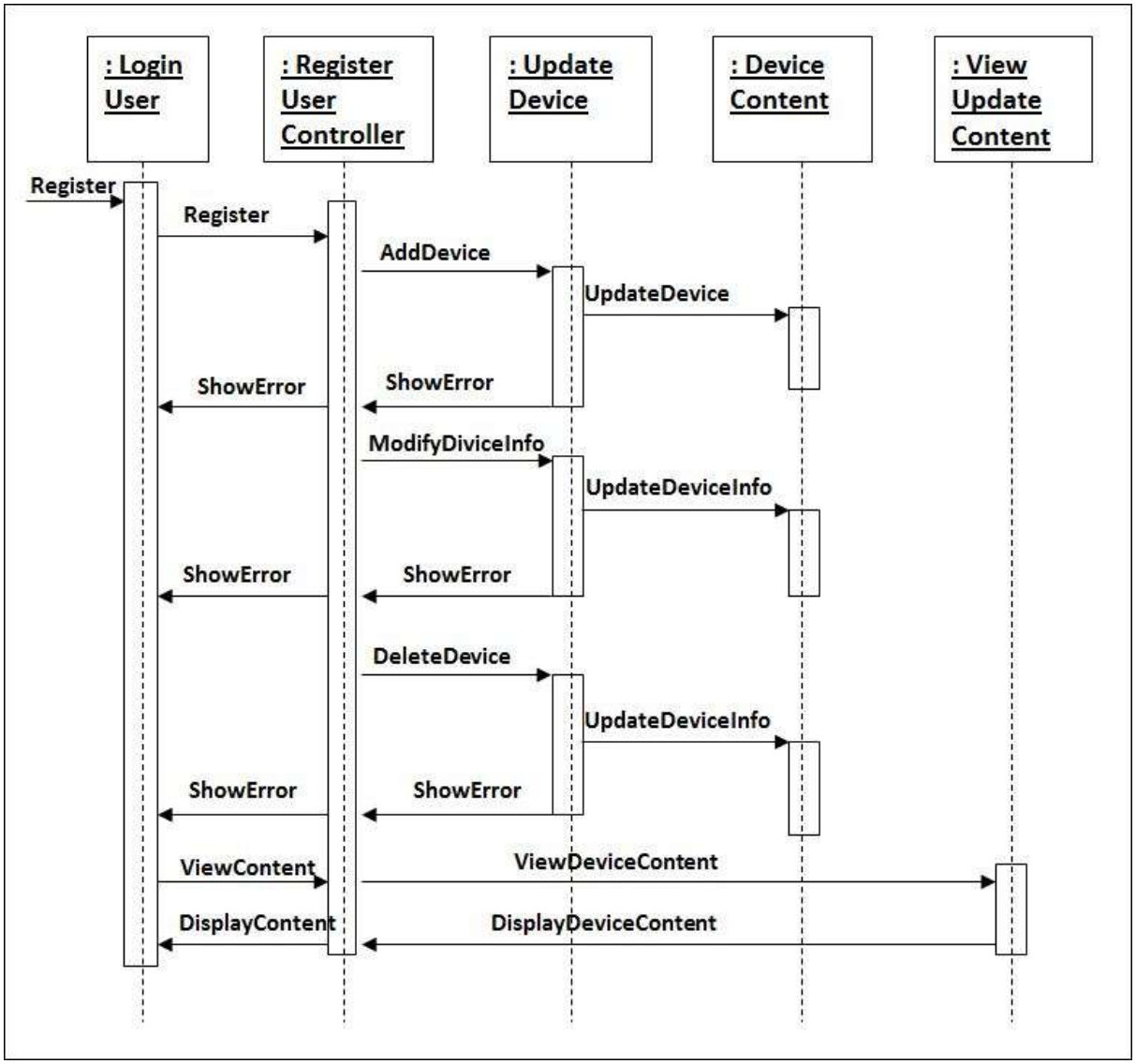


Figure 7:2 level DFD

SEQUENCE DIAGRAMS

REGISTER AND UPDATE NEW DEVICE




: Login User
: Register User Controller
: Update Device
: Device Content
: View Update Content
Register
Register
AddDevice
UpdateDevice
ShowError
ShowError
ModifyDiviceInfo
UpdateDeviceInfo
ShowError
ShowError
DeleteDevice
UpdateDeviceInfo
ShowError
ShowError
ViewContent
ViewDeviceContent
DisplayContent
DisplayDeviceContent

ENTITY RELATIONSHIP MODEL,

We will design a RDBMS for School Management System. The entities and their attributes are listed below. Attributes in Bold letter is the unique key.

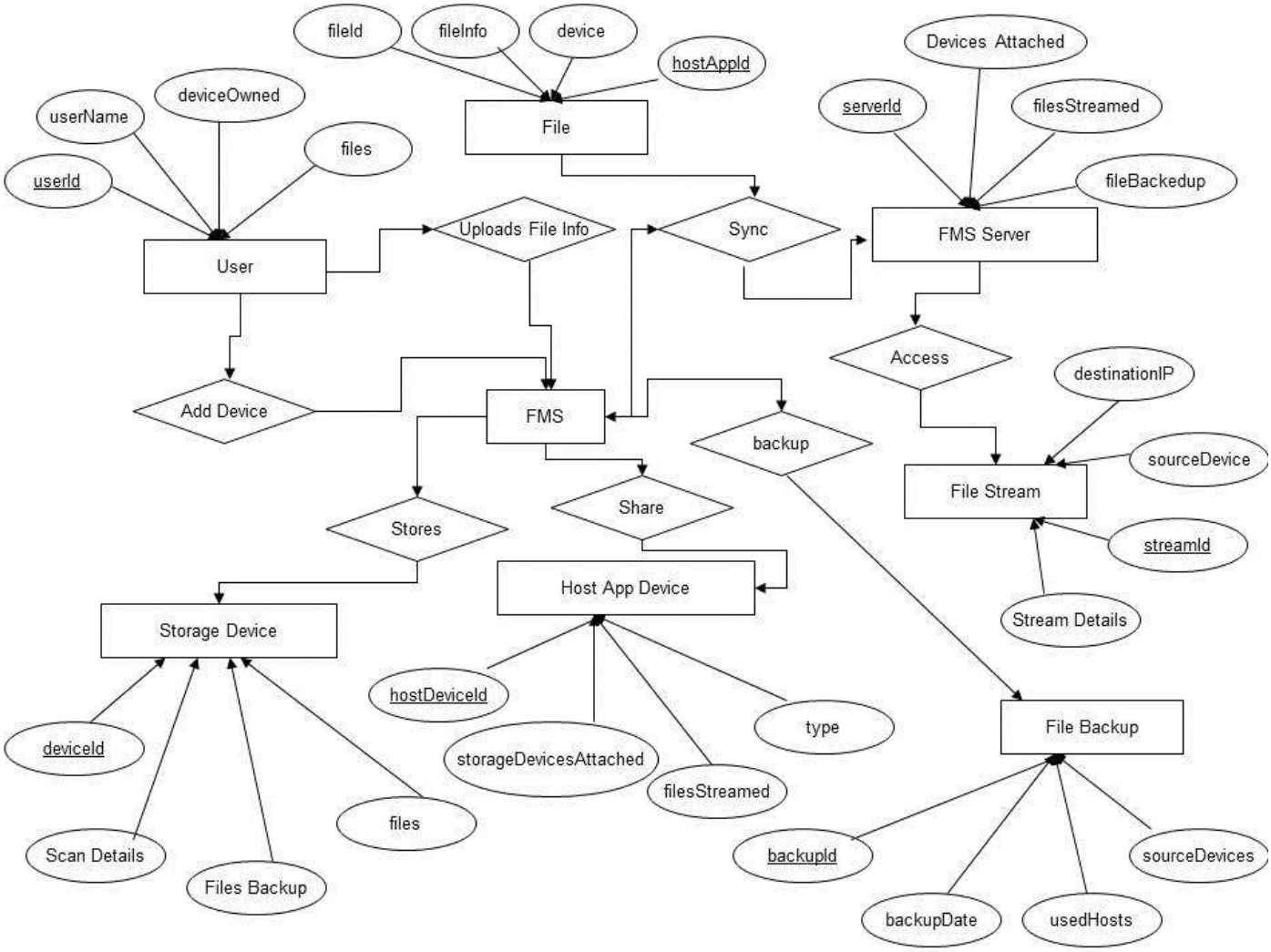
Entity	Attribute
UserInfo	userId; userName; deviceOwned; files;
FileBackupInfo	backupId; backupDate; usedHosts; sourceDevices; files;
FileStreamInfo	streamId; streamDate; usedHost; sourceDevice; destinationIP; files;
StorageDeviceInfo	deviceId; lastScannedDate; lastUsedHost; files; filesBackupHere;
HostAppDeviceInfo	hostDeviceId; storageDevicesAttached; filesStreamed; type;
FileInfo	fileId; fileName; filePath; device; hostAppId; fileSize; fileType; creationDate; modifyDate; backups; streams;
FMSServerInfo	serverId; storageDevicesAttached; hostDevicesAttached; filesStreamed; fileBackedup;

Relationship between Entities:

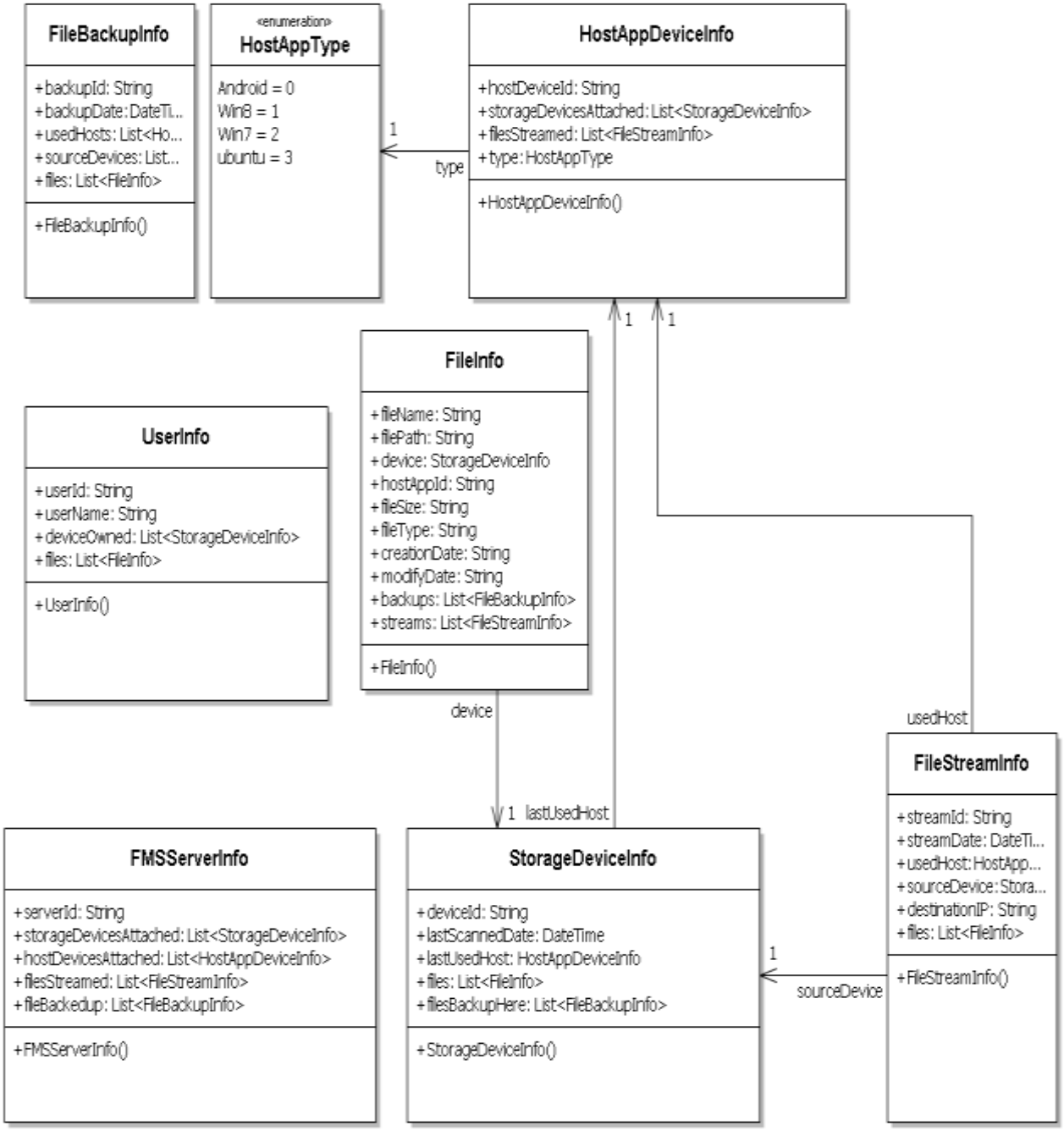
User has Files 1 : N

Devices has Files 1 : N

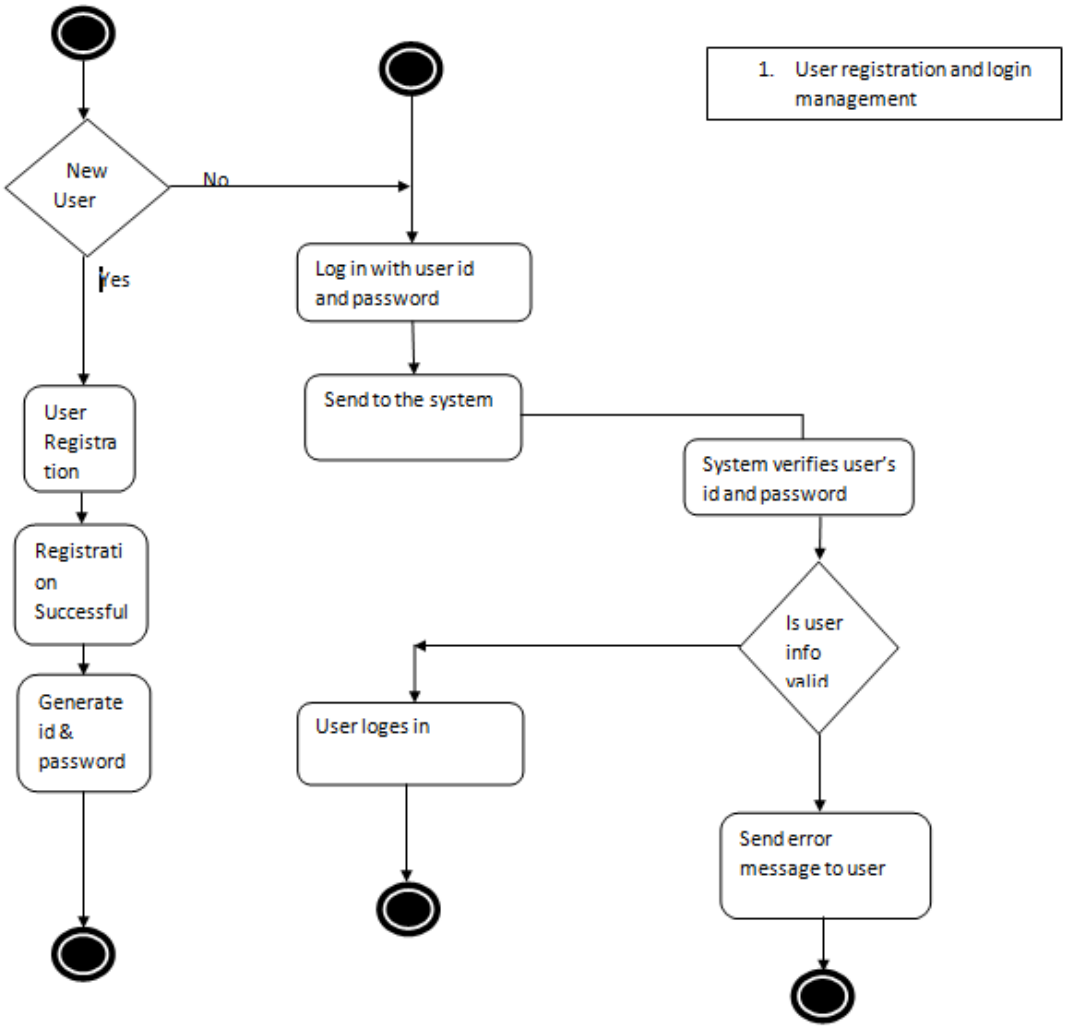
HostApp hosts Devices 1:N



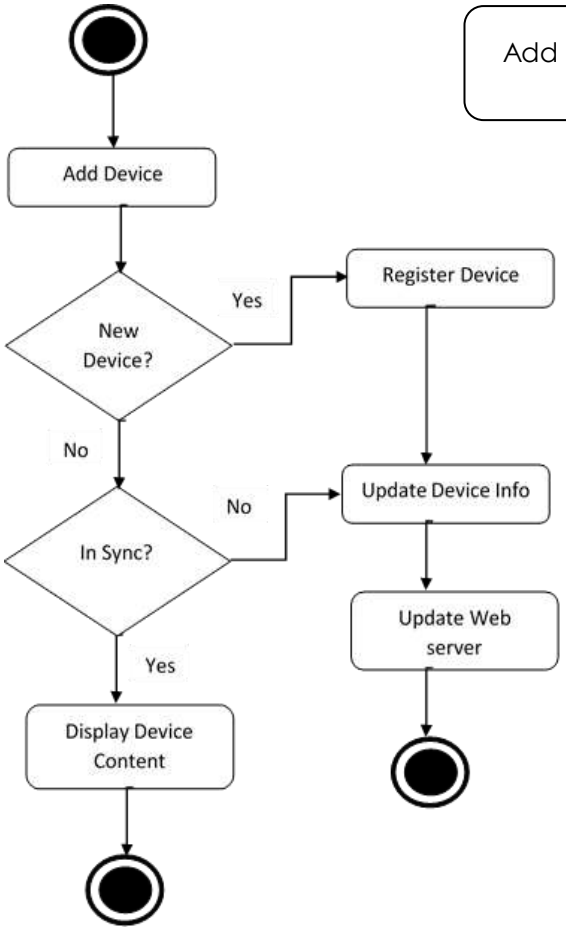
CLASS DIAGRAM



ACTIVITY DIAGRAMS


1. User registration and login management
New User
No
Yes
Log in with user id and password
Send to the system
System verifies user's id and password
Is user info valid
User loges in
Send error message to user
User Registra tion
Registrati on Successful
Generate id & password

Add New Device and Sync Web Server



SYSTEM DESIGN

MODULARISATION DETAILS

File Management System is divided in to following modules:

1. **File Management System Web Controller**: This is the controller or Server component which will aggregate all file information in one place and let users browse through the information they need.
2. **Android Client**: This is a host app for Android which will allow scanning storage devices like device memory, micro sd , sd card and upload the information to server.
3. **Windows 8 Client** : This is a host app for Windows 8 desktop and personal computers which will allow scanning storage devices like hard drive, pen drive , micro sd , sd card, CD, DVD and upload the information to server.
4. **Windows 7 Client**: This is a host app for Windows 7 desktop and personal computers which will allow scanning storage devices like hard drive, pen drive , micro sd , sd card, CD, DVD and upload the information to server.
5. **Ubuntu (Linux) Client** : This is a host app for Ubuntu desktop and personal computers which will allow scanning storage devices like hard drive, pen drive , micro sd , sd card, CD, DVD and upload the information to server.
6. **Storage Device Controller**: This will keep track of all storage devices available, access the devices unique id and tag the file information with device id for future use.
7. **Server REST API**: The Server REST component will facilitate interaction with online database from various devices.
8. **FMS Search Engine**: This will enable searching through file information and find the desired file.
9. **Web Sync Handler**: The host apps won't sync always. It will sync with server according to the user preferences.
10. **Web Viewer Client:** This client will help users to view desired information in a user friendly format.

DATABASE DESIGN

The database used for this software is called Fmsdb. Database tables and corresponding keys are shown in tabular form. It shows the tables and its columns. A key in Bold is the primary key.

Screenshots of table structures:

Table: UserInfo

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	userId	int(10)			No	*None*		
	userName	varchar(100)	latin1_swedish_ci		No	*None*		
	deviceOwned	varchar(100)	latin1_swedish_ci		No	*None*		
	files	varchar(100)	latin1_swedish_ci		No	*None*		

Table: FileBackupInfo

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	backupId	int(10)			No	None		
	backupData	varchar(100)	latin1_swedish_ci		No	None		
	usedHosts	varchar(100)	latin1_swedish_ci		No	None		
	sourceDevices	varchar(100)	latin1_swedish_ci		No	None		
	files	varchar(100)	latin1_swedish_ci		No	None		

Table: FileStreamInfo

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	streamId	int(10)			No	None		
	streamDate	datetime			No	None		
	usedHost	varchar(100)	latin1_swedish_ci		No	None		
	sourceDevice	varchar(100)	latin1_swedish_ci		No	None		
	destinationIP	varchar(100)	latin1_swedish_ci		No	None		
	files	varchar(100)	latin1_swedish_ci		No	None		

Table: StorageDeviceInfo

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	deviceId	int(10)			No	None		
	lastScannedDate	datetime			No	None		
	lastUsedHost	varchar(100)	latin1_swedish_ci		No	None		
	files	varchar(100)	latin1_swedish_ci		No	None		
	filesBackupHere	varchar(100)	latin1_swedish_ci		No	None		

Table: HostAppDeviceInfo

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	hostDeviceId	int(10)			No	None		
	storageDevicesAttached	varchar(100)	latin1_swedish_ci		No	None		
	filesStreamed	varchar(100)	latin1_swedish_ci		No	None		
	type	varchar(100)	latin1_swedish_ci		No	None		

Table: FileInfo

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	fileId	int(10)			No	None		
	fileName	varchar(100)	latin1_swedish_ci		No	None		
	filePath	varchar(100)	latin1_swedish_ci		No	None		
	device	varchar(100)	latin1_swedish_ci		No	None		
	hostAppId	int(10)			No	None		
	fileSize	float			No	None		
	fileType	varchar(100)	latin1_swedish_ci		No	None		
	creationDate	datetime			No	None		
	modifyDate	datetime			No	None		
	backups	varchar(100)	latin1_swedish_ci		No	None		
	streams	varchar(100)	latin1_swedish_ci		No	None		

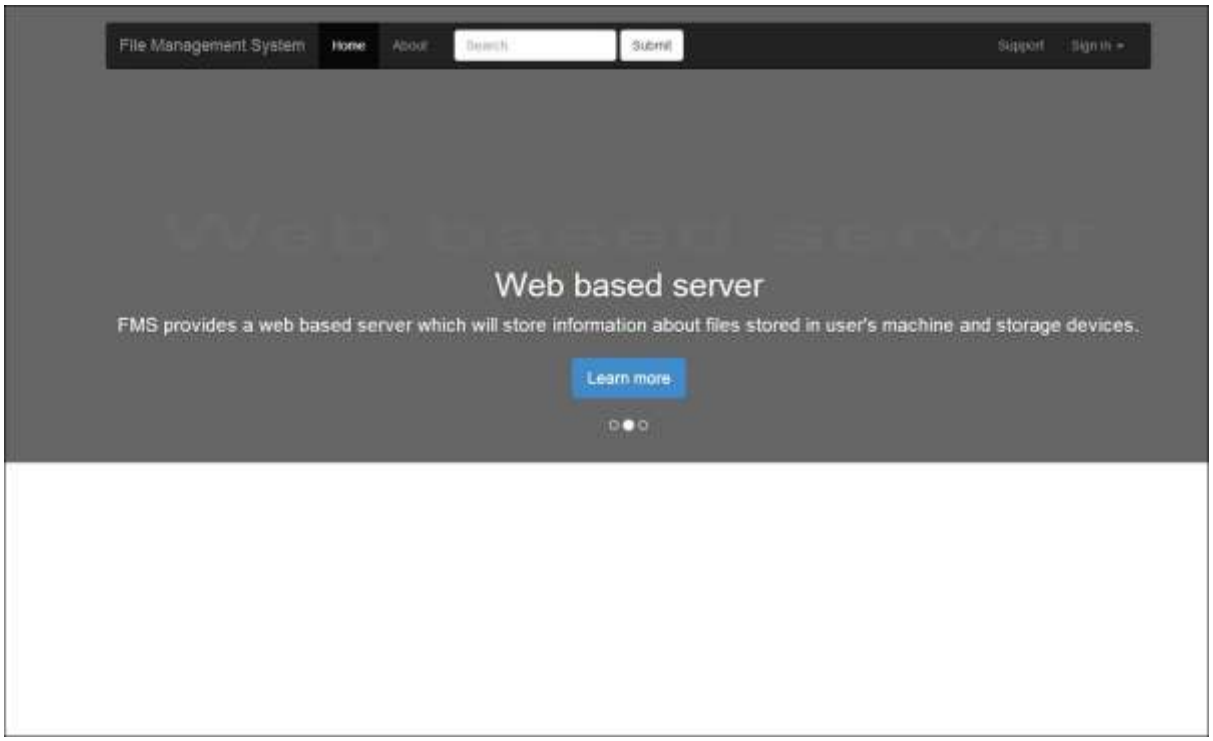
Table: FMSServerInfo

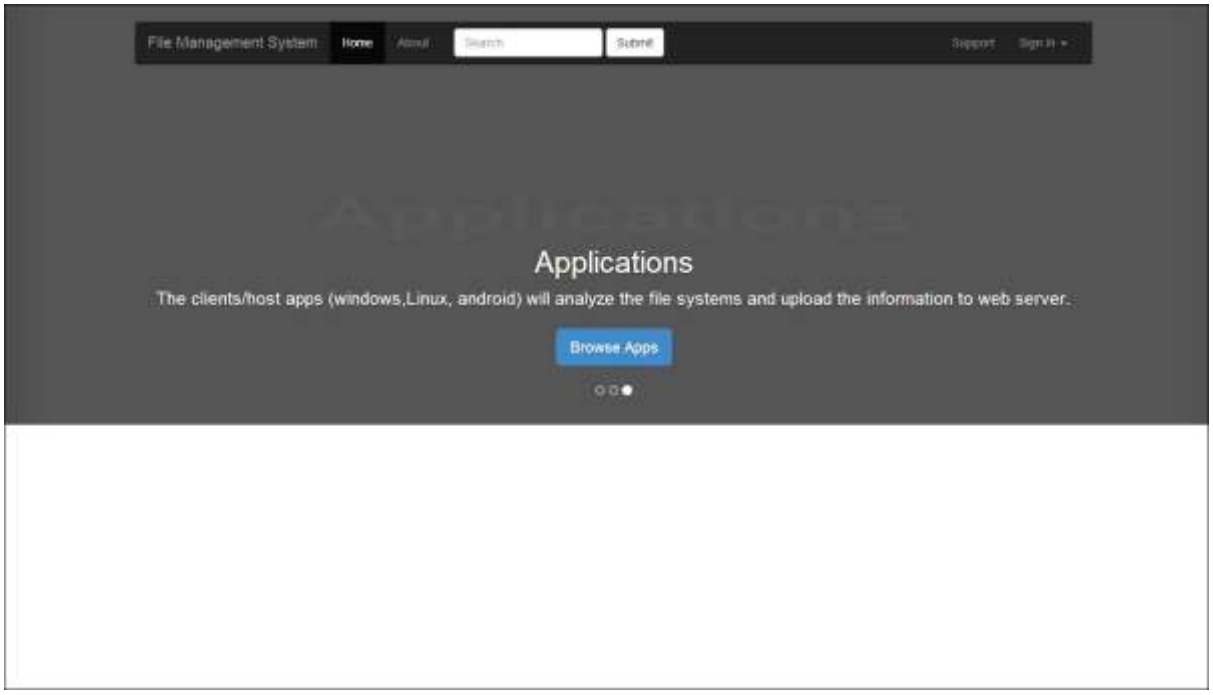
	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	serverId	int(10)			No	None		
☐	storageDevicesAttached	varchar(100)	latin1_swedish_ci		No	None		
☐	hostDevicesAttached	varchar(100)	latin1_swedish_ci		No	None		
☐	filesStreamed	varchar(100)	latin1_swedish_ci		No	None		
☐	fileBackedup	varchar(100)	latin1_swedish_ci		No	None		

Table	**Column**
UserInfo	**userId**; userName; deviceOwned; files;
FileBackupInfo	**backupId**; backupDate; usedHosts; sourceDevices; files;
FileStreamInfo	**streamId**; streamDate; usedHost; sourceDevice; destinationIP; files;
StorageDeviceInfo	**deviceId**; lastScannedDate; lastUsedHost; files; filesBackupHere;
HostAppDeviceInfo	**hostDeviceId**; storageDevicesAttached; filesStreamed; type;
FileInfo	**fileId**; fileName; filePath; device; hostAppId; fileSize; fileType; creationDate; modifyDate; backups; streams;
FMSServerInfo	**serverId**; storageDevicesAttached; hostDevicesAttached; filesStreamed; fileBackedup;

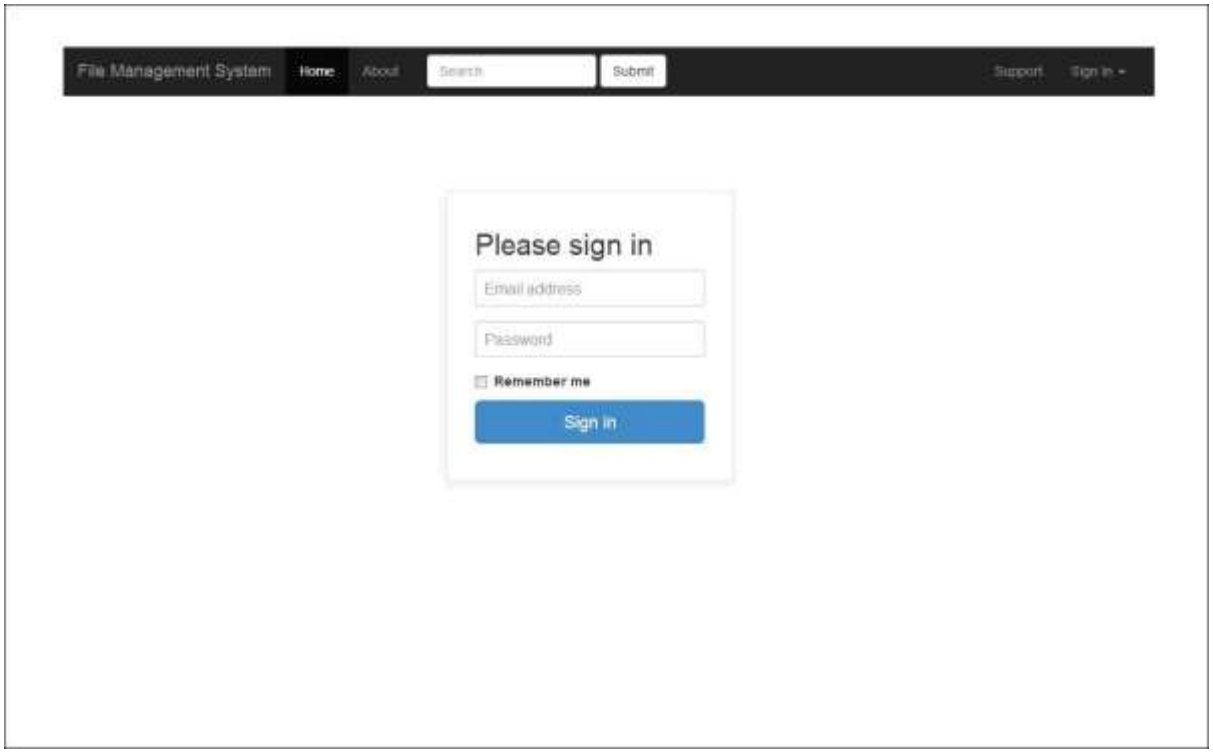
USER INTERFACE DESIGN

WEB HOME PAGE



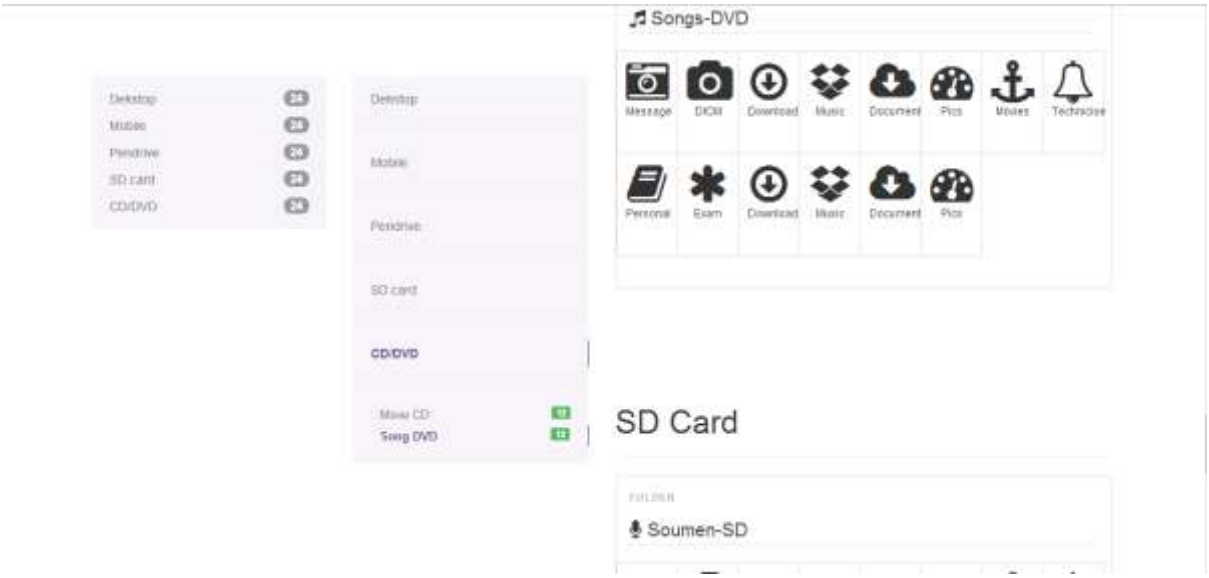


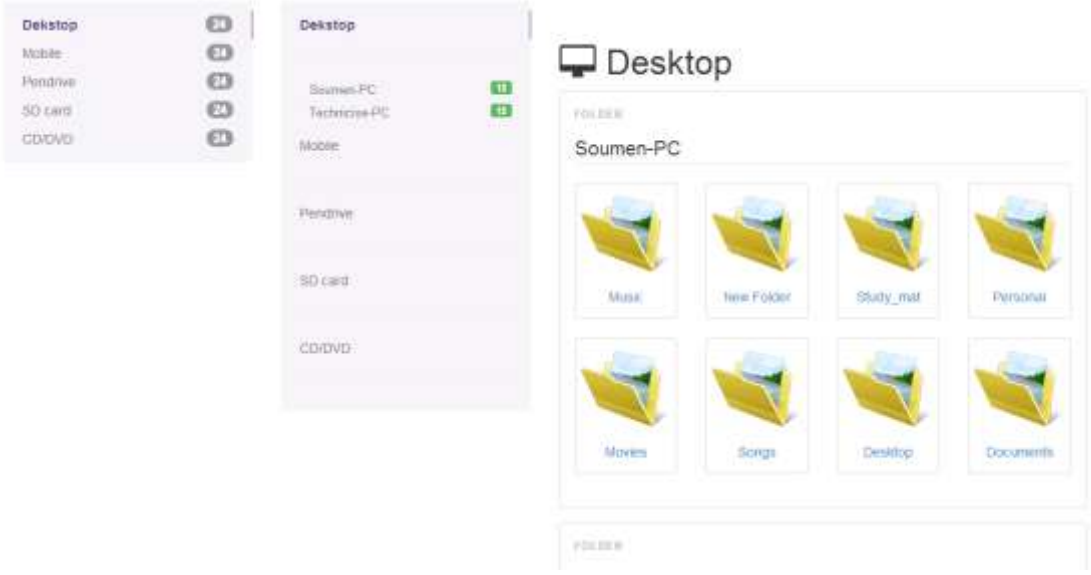
WEB LOGIN

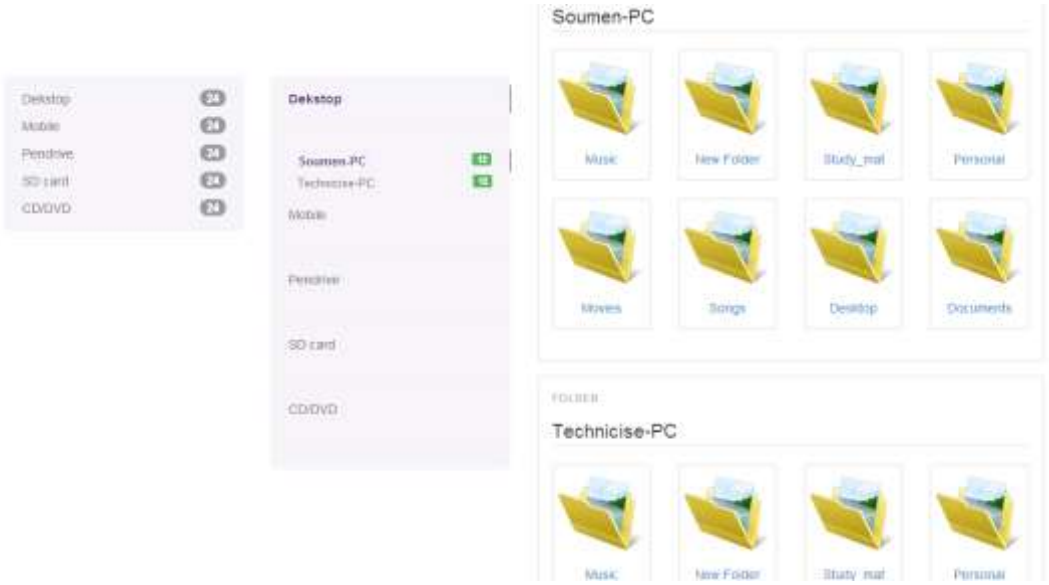


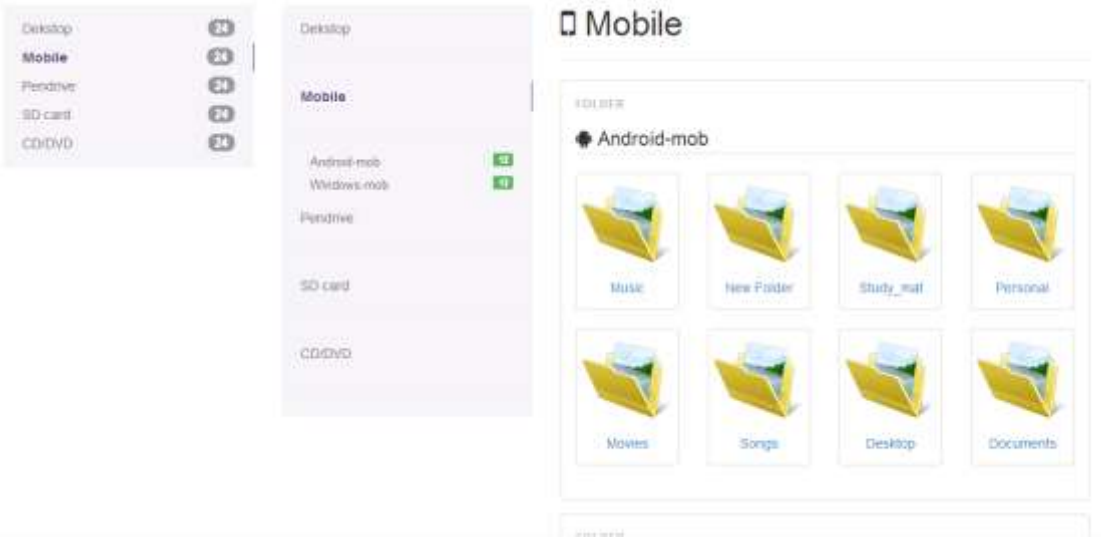
WEB DASHBOARD

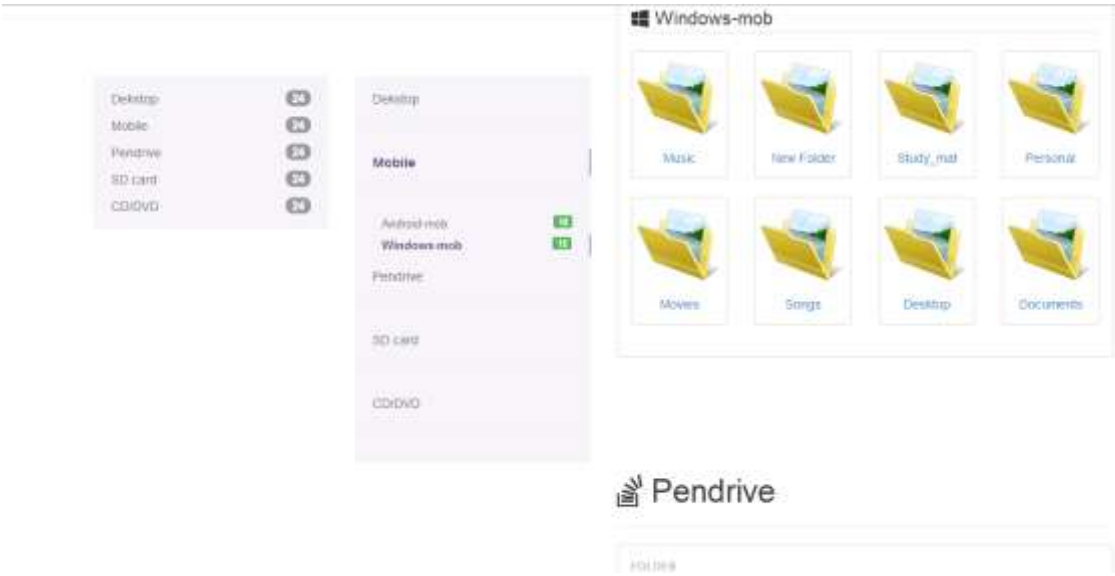

Dekstop 24
Mobile 24
Pendrive 24
SD card 24
CD/DVD 24
Dekstop
Mobile
Pendrive
SD card
CD/DVD
Movie CD 12
Song DVD 12
CD/DVD
Movie-CD
Message
DICM
Download
Music
Document
Pics
Movies
Technicise
Personal
Exam
Download
Music
Document
Pics
Songs-DVD


Songs-DVD
Dekstop 24
Mobile 24
Pendrive 24
SD card 24
CD/DVD 24
Dekstop
Mobile
Pendrive
SD card
CD/DVD
Movie CD 12
Song DVD 12
SD Card
Soumen-SD

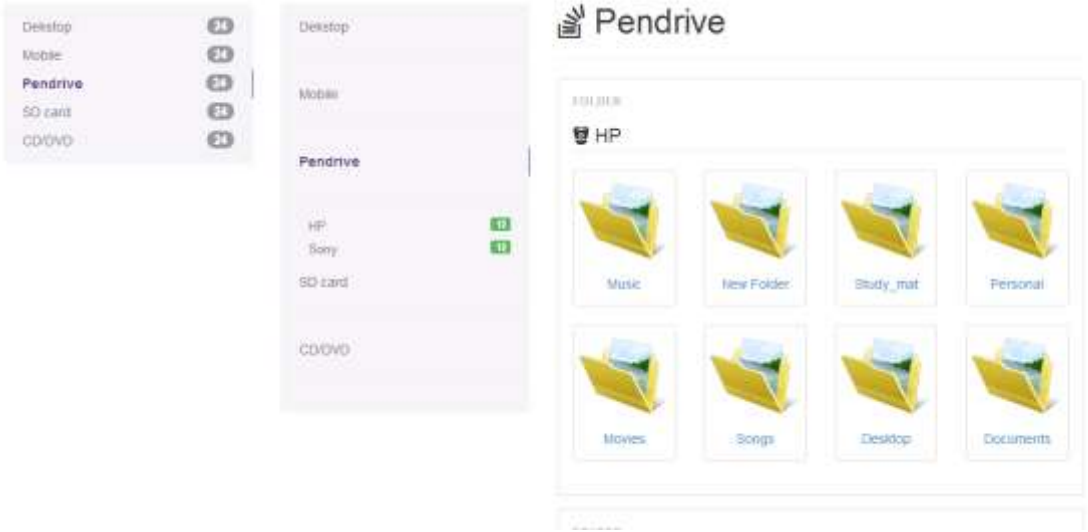

Dekstop 24
Mobile 24
Pendrive 24
SD card 24
CD/DVD 24
Dekstop
Soumen-PC 12
Technicise-PC 12
Mobile
Pendrive
SD card
CD/DVD
Desktop
Soumen-PC
Music
New Folder
Study_mat
Personal
Movies
Songs
Desktop
Documents

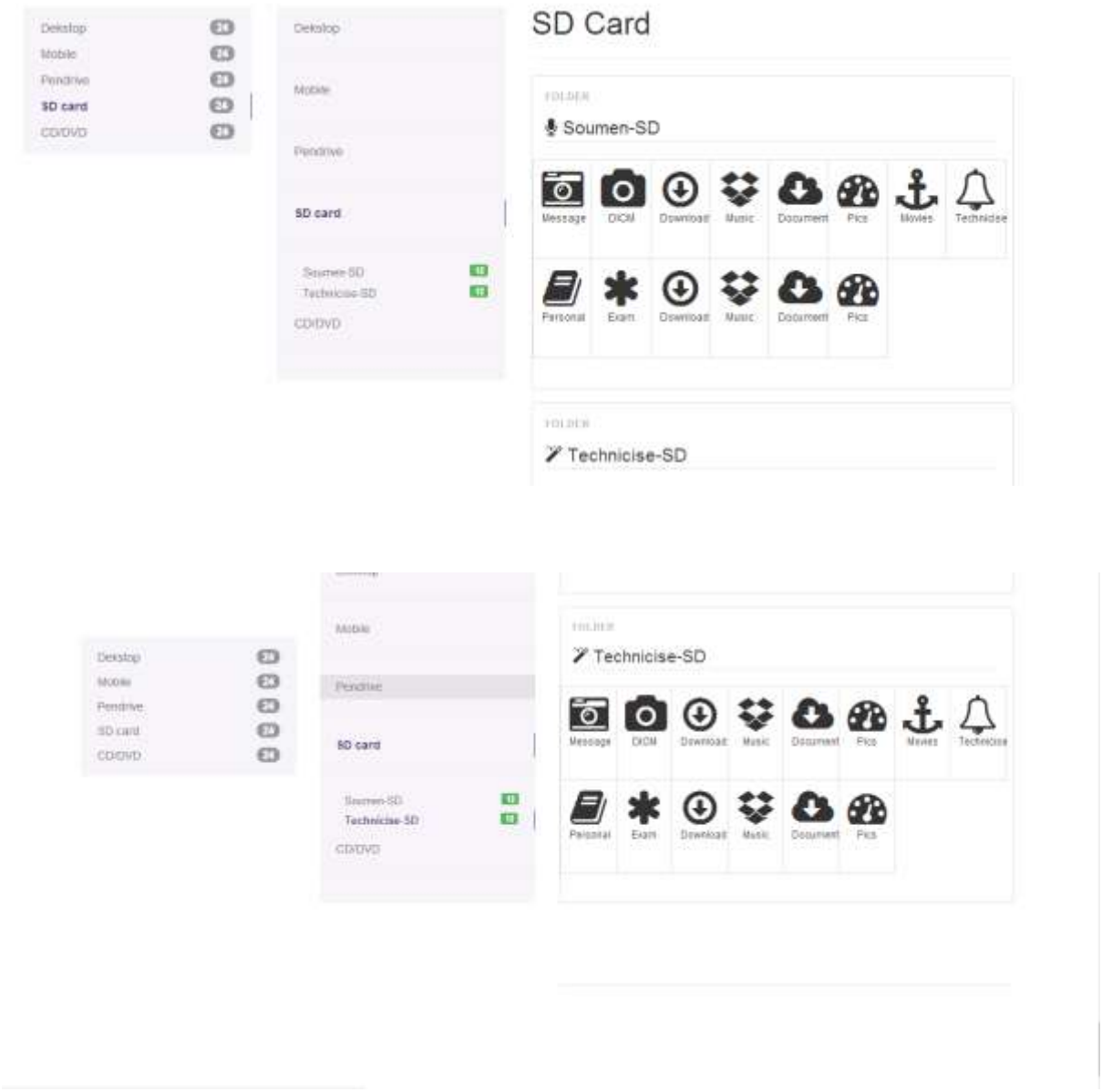

Desktop 24
Mobile 24
Pendrive 24
SD card 24
CD/DVD 24
Dekstop
Soumen-PC 42
Technicise-PC 42
Mobile
Pendrive
SD card
CD/DVD
Soumen-PC
Music
New Folder
Study_mat
Personal
Movies
Songs
Desktop
Documents
FOLDER
Technicise-PC
Music
New Folder
Study_mat
Personal


Desktop 24
Mobile 24
Pendrive 24
SD card 24
CD/DVD 24
Dekstop
Mobile
Android-mob 42
Windows-mob 42
Pendrive
SD card
CD/DVD
Mobile
FOLDER
Android-mob
Music
New Folder
Study_mat
Personal
Movies
Songs
Desktop
Documents

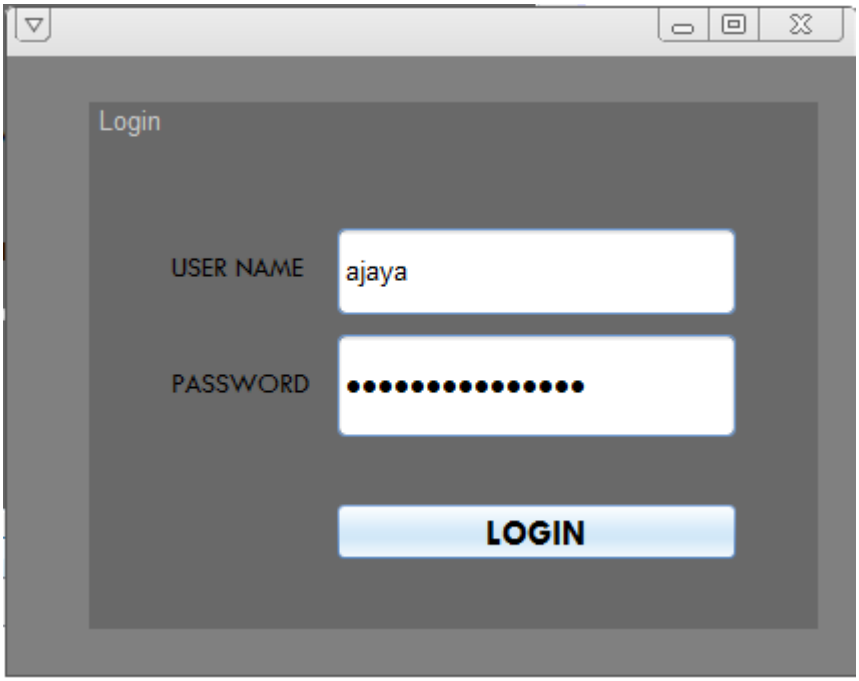

Windows-mob
Music
New Folder
Study_mat
Personal
Movies
Songs
Desktop
Documents
Pendrive

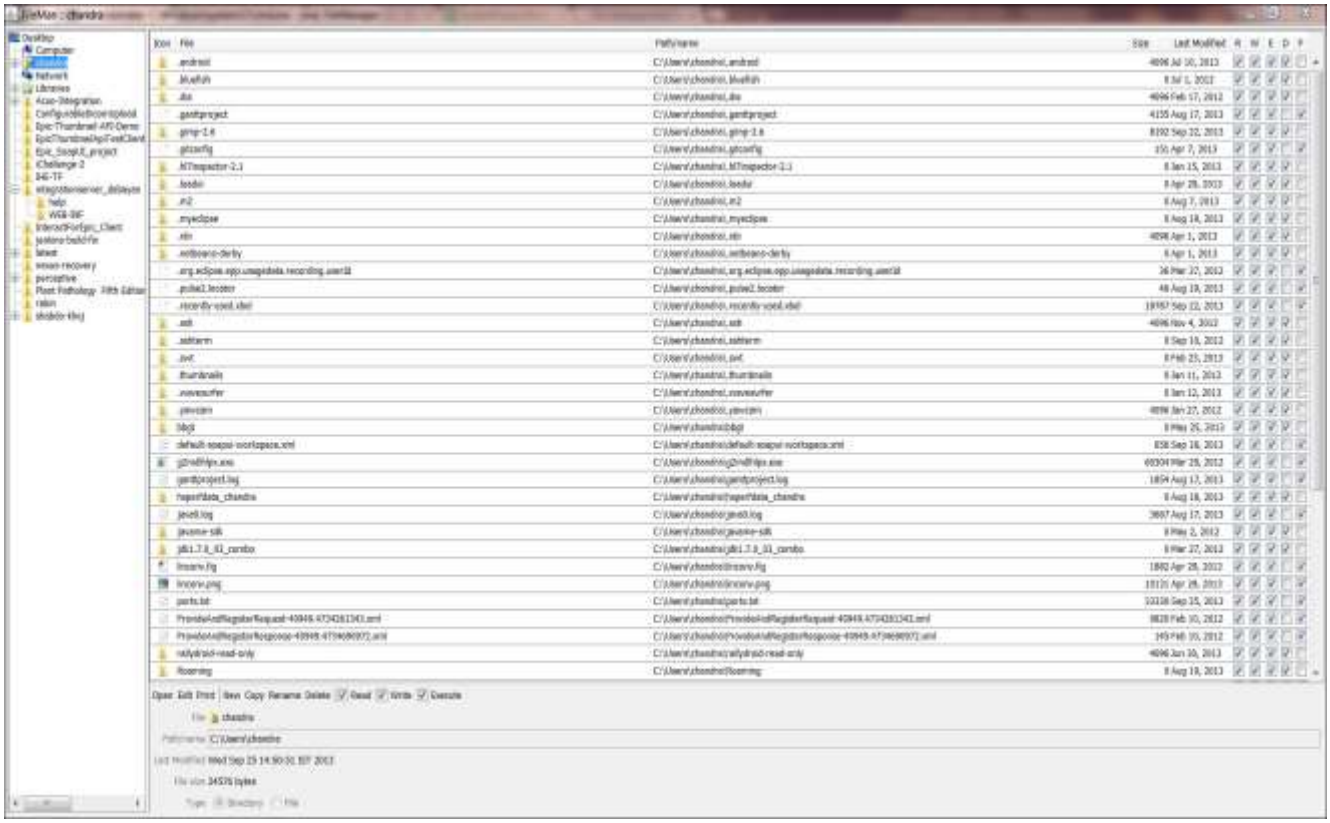

Sony
Music
New Folder
Study_mat
Personal
Movies
Songs
Desktop
Documents
CD/DVD

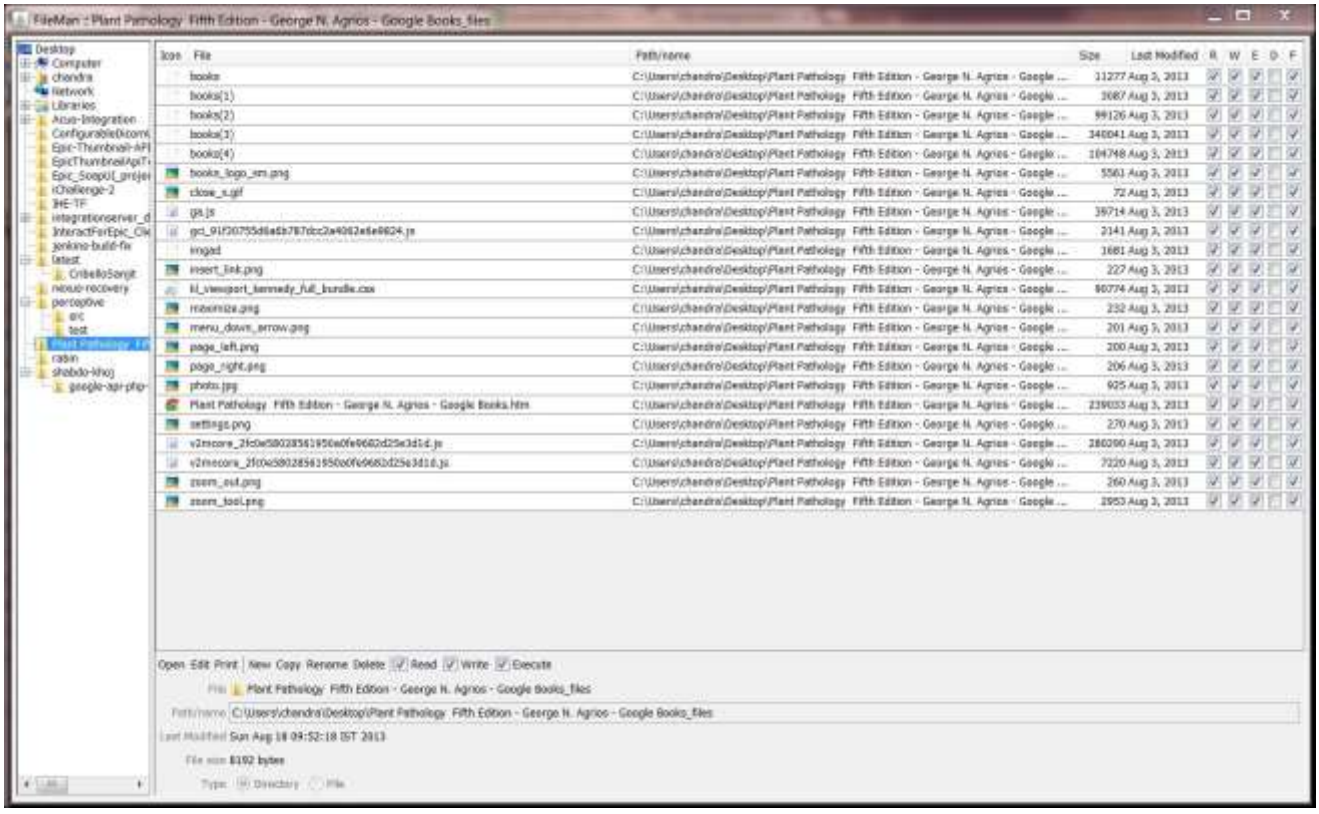

Pendrive
HP
Music
New Folder
Study_mat
Personal
Movies
Songs
Desktop
Documents

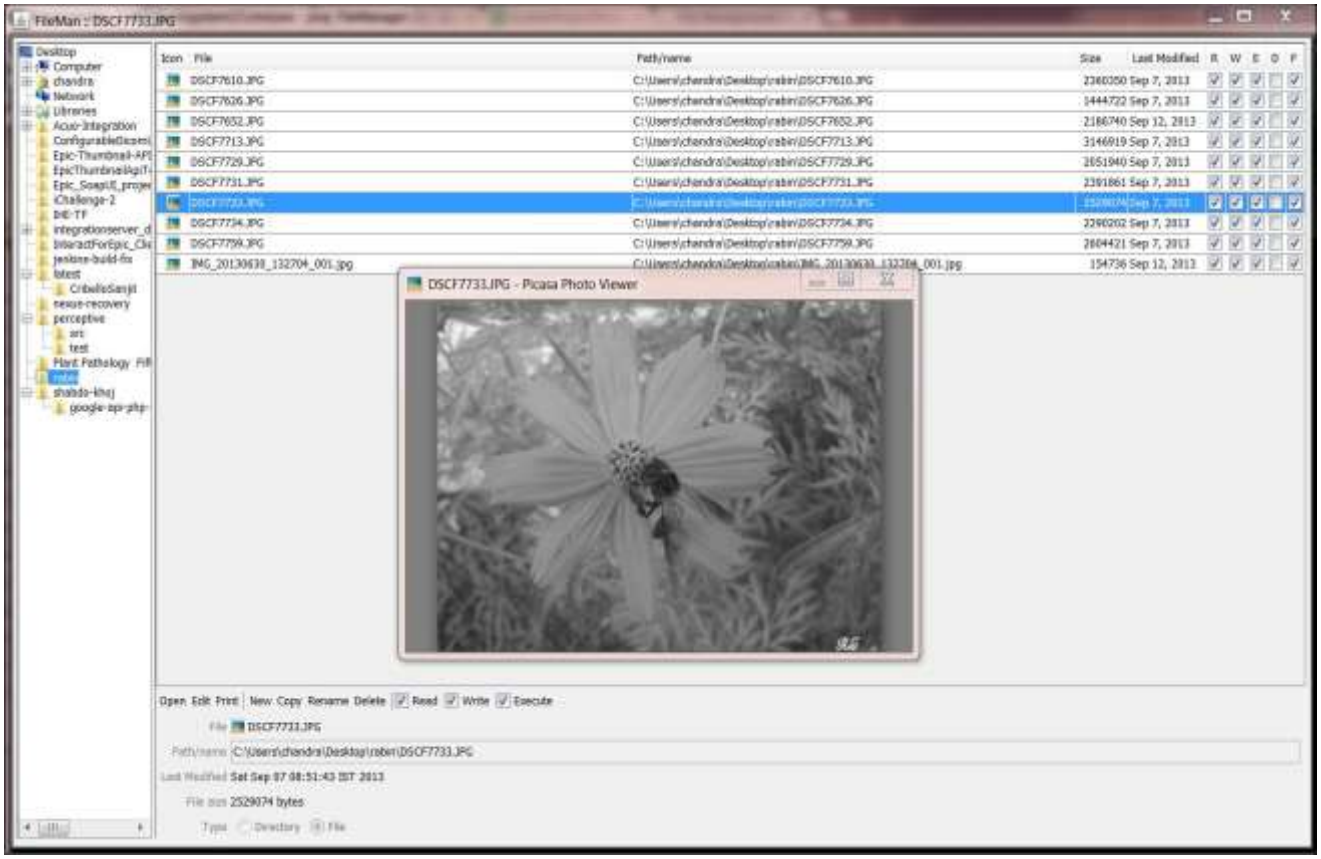

Dekstop
Mobile
Pendrive
SD card
CD/DVD
SD Card
Soumen-SD
Technicise-SD
Message
DICM
Download
Music
Document
Pics
Movies
Technicise
Personal
Exam

LINUX APP

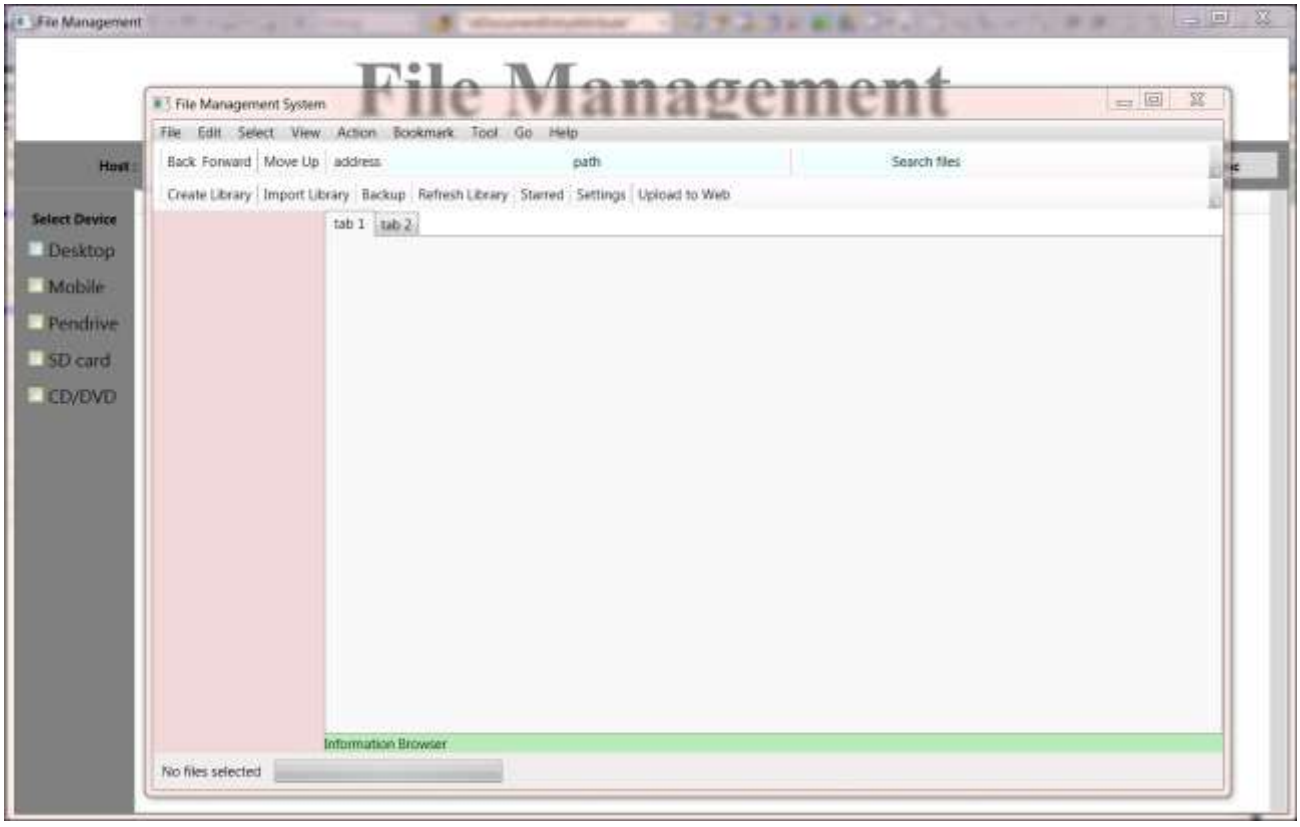

Login
USER NAME
ajaya
PASSWORD
LOGIN



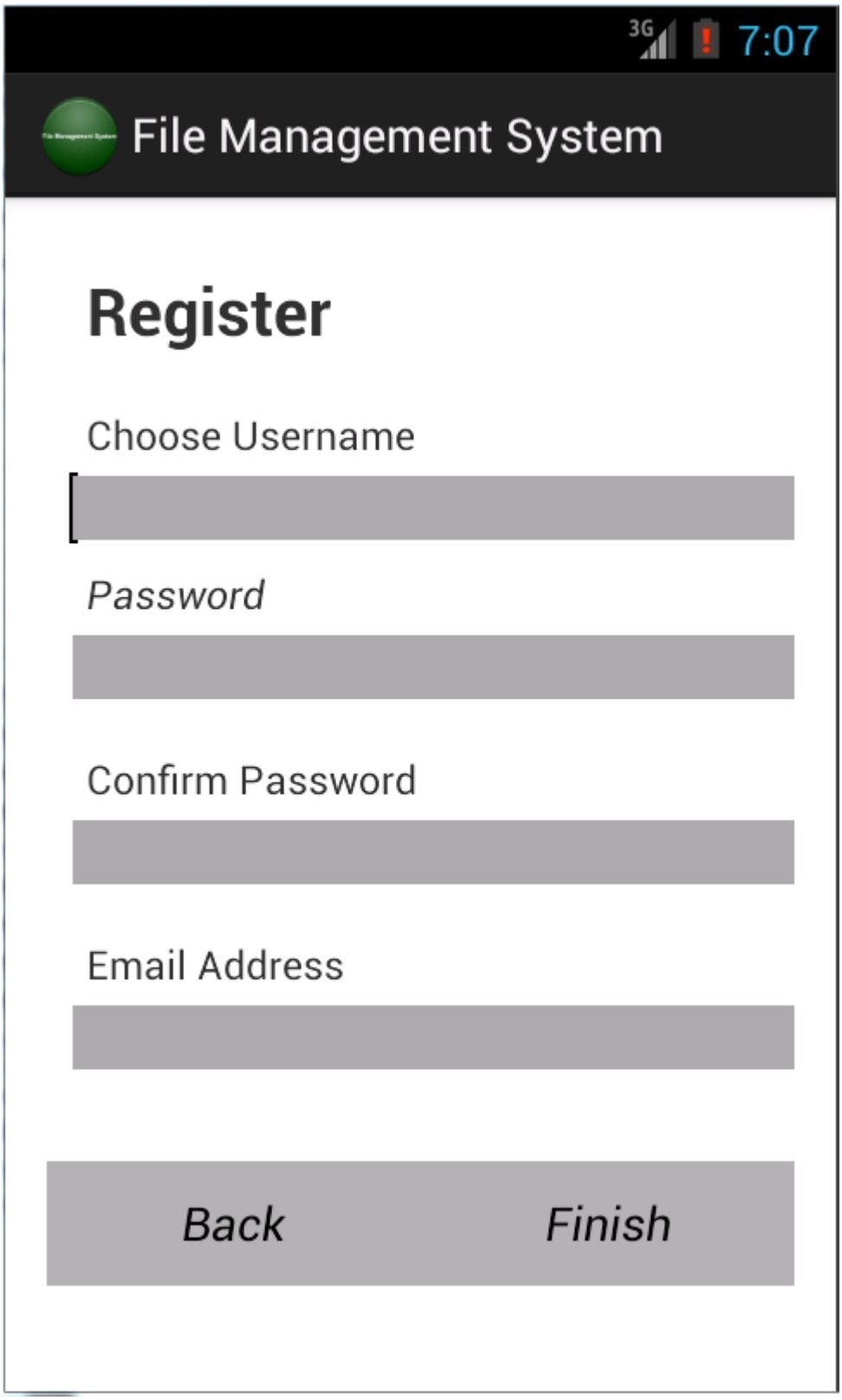

FileMan :: Plant Pathology Fifth Edition - George N. Agrios - Google Books_files


FileMan :: DSCF7731.JPG
DSCF7731.JPG - Picasa Photo Viewer

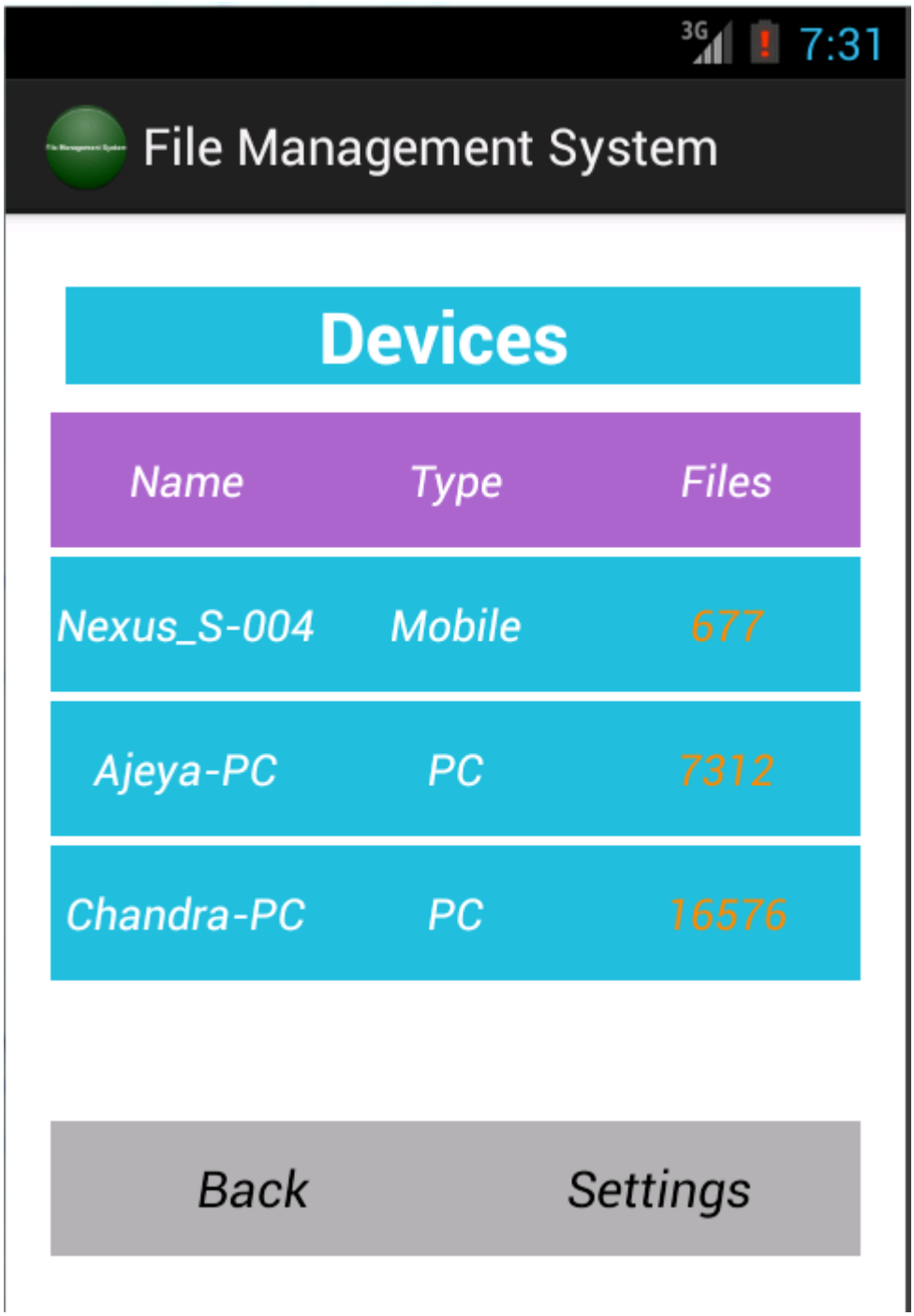
WINDOWS APP

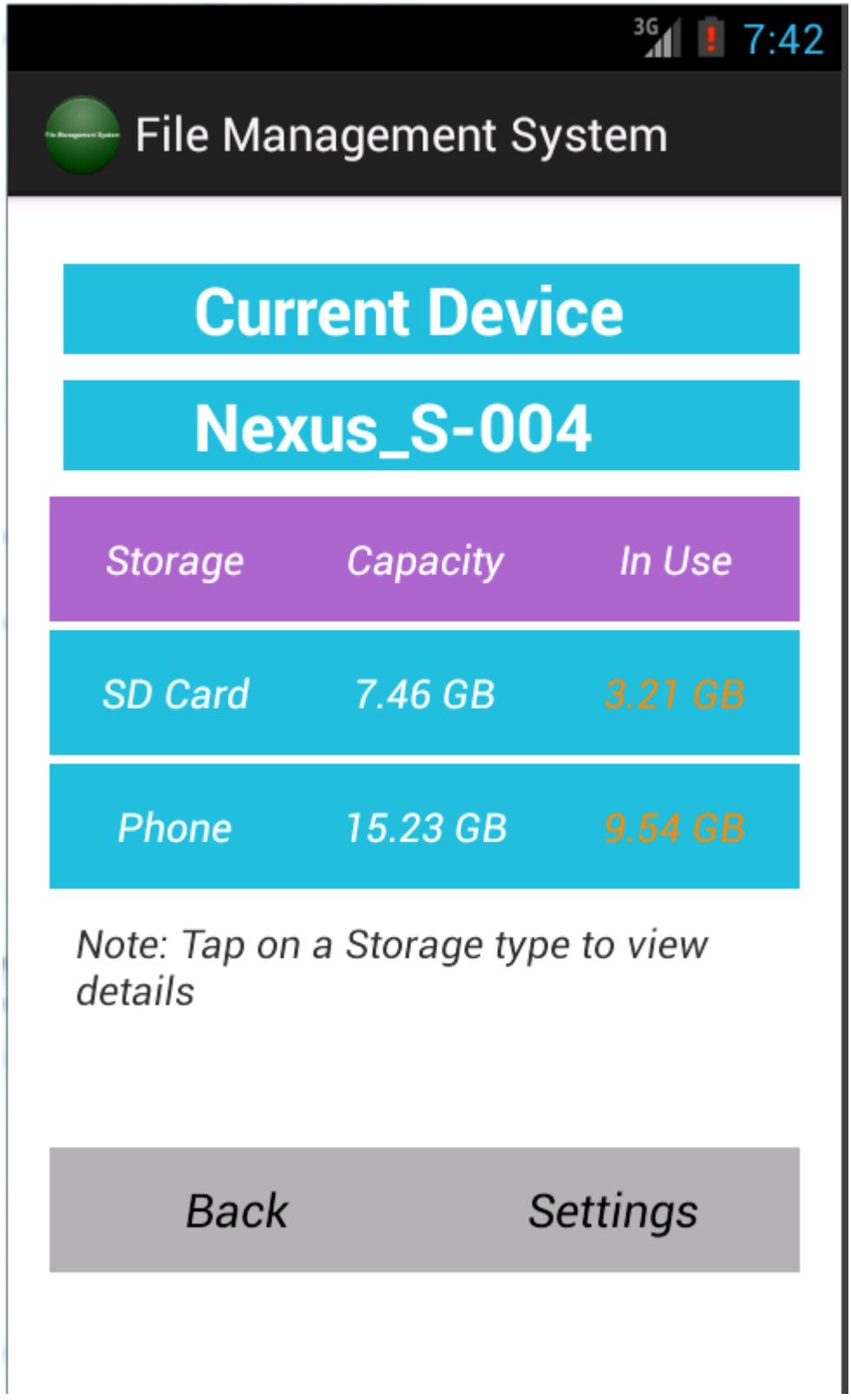


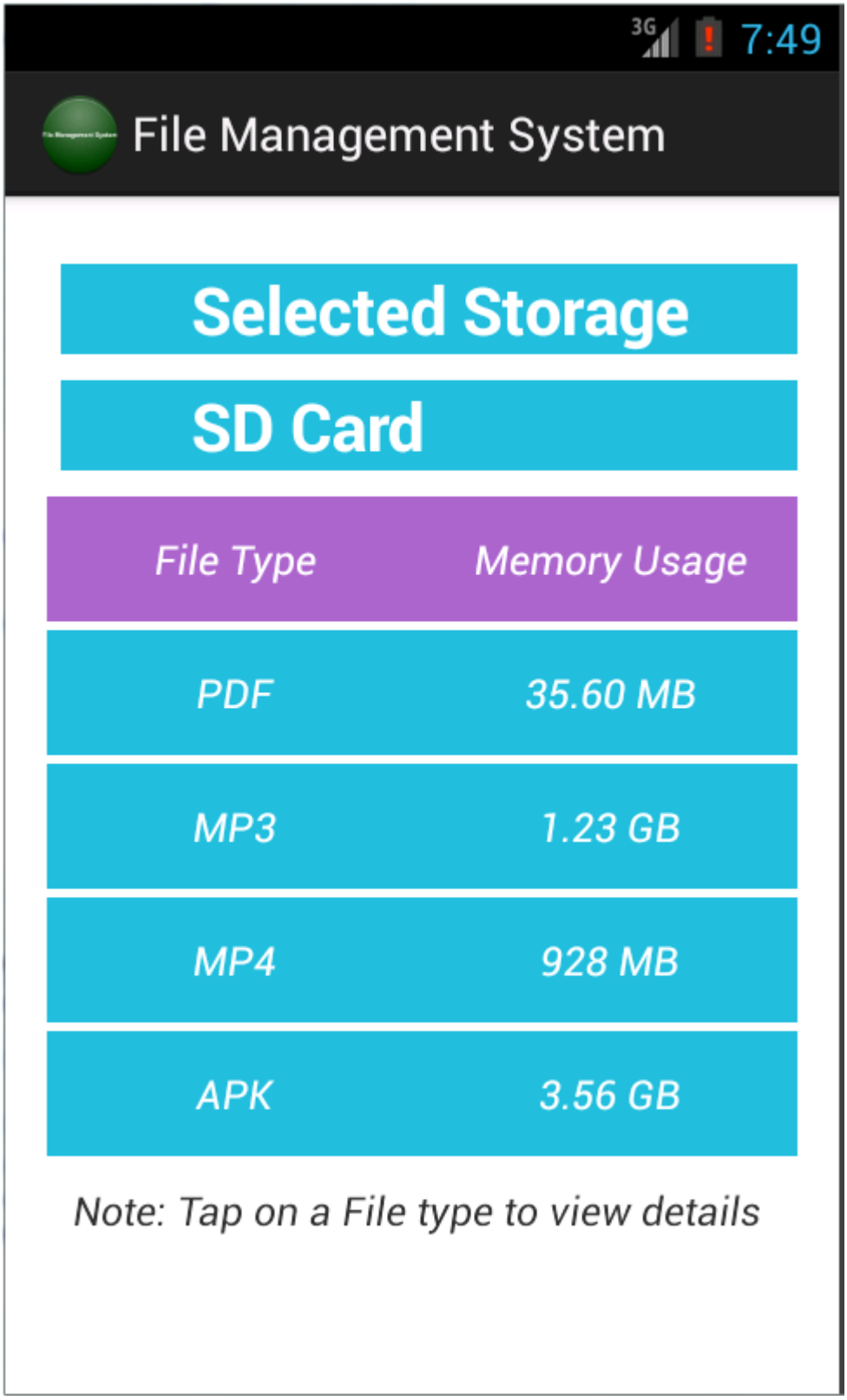
ANDROID APP


3G
7:07
File Management System
Register
Choose Username
Password
Confirm Password
Email Address
Back
Finish


3G
7:00
File Management System
Email/Phone Login
User Name
Required
Password
Required
Login
Remember
I forgot my password


3G
7:31
File Management System
Devices
Name
Type
Files
Nexus_S-004
Mobile
677
Ajeya-PC
PC
7312
Chandra-PC
PC
16576
Back
Settings


3G
7:42
File Management System
Current Device
Nexus_S-004
Storage
Capacity
In Use
SD Card
7.46 GB
3.21 GB
Phone
15.23 GB
9.54 GB
Note: Tap on a Storage type to view details
Back
Settings



TEST CASES (UNIT TEST CASES AND SYSTEM TEST CASES)

UNIT TEST CASES

Test Case Id	Type	Github ID	Subject	Test Name	Test Description	Step Name	Description	Expected Result
FMS-001	Manual	f3563be0a9c431104f52839039e86043cf640cf1	E:\DEVELOPERS_ZONE\GitHub\FMS\code	Check Successful Login for **FMS**	The purpose of this test is to verify that the User Id and Password of user is valid.	Step1	Insert wrong User Id and Password. And Click on Login Button.	**FMS** will display error message. And Failed to Login.
FMS-002						Step2	Insert Wrong User Id and valid Password. And Click on Login Button.	**FMS** will display error message. And Failed to Login.
FMS-003						Step3	Insert Valid User Id and Wrong Password. And Click on Login Button.	**FMS** will display error message. And Failed to Login.
FMS-004						Step4	Insert Nothing in User Id and Password fields. And Click on Login Button.	**FMS** will display error message. And Failed to Login.
FMS-005						Step5	Insert Nothing in User Id and insert Valid Password fields. And Click on Login Button.	**FMS** will display error message. And Failed to Login.

FMS-006						Step6	Insert Nothing in Password and insert Valid User Id fields. And Click on Login Button.	**FMS** will display error message. And Failed to Login.
FMS-007						Step7	Insert Nothing in User Id and insert invalid Password fields. And Click on Login Button.	**FMS** will display error message. And Failed to Login.
FMS-008						Step8	Insert Nothing in Password and insert invalid User Id fields. And Click on Login Button.	**FMS** will display error message. And Failed to Login.
FMS-009						Step9	Insert valid User Id and Password. And Click on Login Button.	Successfully login.
FMS-010	Manual	d01197ee4cd3bee9245874b5937ba740019fd131	E:\DEVELOPERS_ZONE\GitHub\FMS\code	Check Successful Registration for New **FMS** User.	The purpose of this test is to verify that the all new connection could be creating new Account By Registration.	Step1	Click on Registration link.	New Account creation area is opened.

FMS-011						Step2	Enter existing new User Id, Password, Retype Same password and Hints. And Click on Login Button.	**FMS** will display error message. And Failed to Registration.
FMS-012						Step3	Enter existing new User Id, Password, Retype Same password and Hints. And Click on Login Button.	**FMS** will display error message. And Failed to Registration.
FMS-013						Step4	Enter new User Id, Password, Retype Different password and Hints. And Click on Login Button.	**FMS** will display error message. And Failed to Registration.
FMS-014						Step5	Enter new User Id, Password, Retype Same password and enter nothing in Hints fields. And Click on Login Button.	**FMS** will display error message. And Failed to Registration.
FMS-015						Step6	Enter new User Id, Password, Hints and nothing in Retype password field. And Click on Login Button.	**FMS** will display error message. And Failed to Registration.

FMS-016						Step7	Enter nothing new User Id, Password, Retype password and Hints Fields. And Click on Login Button.	**FMS** will display error message. And Failed to Registration.
FMS-017						Step8	Enter Proper new User Id, Password, Retype Same password and Hints Fields. And Click on Login Button.	Successful Registration is done and this area is closed and come to login area.
FMS-018	Manual	f0657bbdf47e26ec481fa172b0fa76f9becb2681	E:\DEVELOPERS_ZONE\GitHub\FMS\code	Check Successful addition of device.	The purpose of this test is to check whether different types of devices can be added or not.	Step1	Login from desktop application, in Windows OS and click on add device info button.	The device gets added, corresponding notification is displayed.
FMS-019						Step2	Login from desktop application, in Linux OS and click on add device info button.	The device gets added, corresponding notification is displayed.
FMS-020						Step3	Login from an Android mobile phone and click on add device info button.	The device gets added, corresponding notification is displayed.

FMS-021						Step4	Login from Windows OS web browser using the app's website URL and click on add button.	Display error message. You can not perform that action from a web browser. Please login from an Android phone app, Windows/Linux desktop app.
FMS-022						Step5	Login from Android web browser and try to add a new phone	Display error message. You can not perform that action from a web browser. Please login from an Android phone app, Windows/Linux desktop app.
FMS-023						Step6	Login from Windows OS web browser using the app's website URL and click on add button.	Display error message. You can not perform that action from a web browser. Please login from an Android phone app, Windows/Linux desktop app.
FMS-024						Step7	Loogin from a web brwser other than the Android, Linux, Windows OS and try to add device.	Displays error message, the device can not be added from the web browser.

FMS-025	Manual	e22bd0e470f145f3db336ed9e28d474d8f4637d7	E:\DEVELOPERS_ZONE\GitHub\FMS\code	Check Successful fetching of device info that has been added.	The purpose of this test is to verify that the added devices are being displayed or not.	Step1	Login to Android app of file manager and click on show devices.	Show all the previously added devices.
FMS-026						Step2	Login to Windows app of file manager and click on show devices.	Shows all the previously added devices.
FMS-027						Step3	Login to Linux app of file manager and click on show devices.	Shows all the previously added devices.
FMS-028						Step4	Login to the file manager website from any web browser in the world and click on show devices.	Shows all the previously added devices.
FMS-029	Manual	90330b92328d862892fc7743653908 1cc2b7f70d	E:\DEVELOPERS_ZONE\GitHub\FMS\code	Check successful synchronization of devices file structure.	The purpose of this test is to verify whether a user can successfully synchronize a particular device's file structure or not.	Step1	Login to Android OS app, go to the synchronization option and sync current device.	Displays message, device has been synchronized successfully.

FMS-030						Step2	Login to Windows OS app, go to the synchronization option and sync current device.	Displays message, device has been synchronized successfully.
FMS-031						Step3	Login to Linux OS app, go to the synchronization option and sync current device.	Displays message, device has been synchronized successfully.
FMS-032						Step4	Login to Web browser and try to sync.	No sync option is there as that feature is not available for web version.
FMS-033	Manual	47fb570f63ffec837a49e235f629c49cc55a70f0	E:\DEVELOPERS_ZONE\GitHub\FMS\code	Check each device file structure information is available or not.	It is to check each device file structure information is available or not.	Step1	Login from Android app and see all devices info from there.	Show the file structure of all added machines.
FMS-034						Step2	Login from Windows app and see all devices info from there.	Show the file structure of all added machines.
FMS-035						Step3	Login from Linux app and see all devices info from there.	Show the file structure of all added machines.

FMS-036						Step4	Login from any web browser and see all devices info from there.	Show the file structure of all added machines.
FMS-037						Step5	Check whether the right file structure is showing or not from an application.	Shows exactly same file structure that is visible in the de vice storage.
FMS-038						Step6	Check whether the right file structure is showing or not from a web browser.	Shows exactly same file structure that is visible in the de vice storage.
FMS-039	Manual	b8a65899fa408a4d99a8ee8cabbe2ac0b54226b0	E:\DEVELOPERS_ZONE\GitHub\FMS\code	Check successful deleting of a device	It is to check that we can remove a device from the list or not.	Step1	Login to an application and try to remove a device.	The device info gets removed and that is not visible anywhere in the app.
FMS-040						Step2	Login through a browser and try to remove a device.	The device info gets removed and that is not visible anywhere in the app.

Test Case Id	Type	Github ID	Subject	Test Name	Test Description	Step Name	Description	Expected Result
FMS-041	Manual	f3563be0a9c4311004f528390 39e86043cf640cf1	E:\DEVELOPERS_ZONE\GitHub\FMS\code	Check Log in.	It is to check that Login works properly.	Step 1	Click on Login button after inserting invalid User id and password from **FMS**.	Login failed to **FMS**. And can't able to use the feature.
FMS-042						Step 2	Click on Login button after inserting valid User id and password from **FMS**.	Successfully Login to **FMS**. And can able to use the feature.
FMS-043	Manual	d01197ee4cd3bee92458 74b5937ba740019fd131	E:\DEVELOPERS_ZONE\GitHub\FMS\code	Check Successful Registration for New **FMS** User.	The purpose of this test is to verify that the all new connection could be creating new Account By Registration.	Step 1	Click on Registration link.	New Account creation area is opened.
FMS-044						Step 2	Click on Registration button after inserting invalid information from **FMS**.	Registration failed to **FMS**. And can't able to use the feature.
FMS-045						Step 3	Click on Registration button after	Registration Successfully done to **FMS**.

							inserting valid information from **FMS**.	
FMS-046						Step 4	Click on Login button after inserting newly created valid User id and password from **FMS**.	Successfully Login to by new User Id And password **FMS**. And can able to use the feature.
FMS-047	Manual	f0657bbdf47e26ec481fa172b0fa76f9becb2681	E:\DEVELOPERS_ZONE\GitHub\FMS\code	Check Successful addition of a new device	The purpose of this test is to check a machine can be added or not.	Step 1	Add a device by logging in from an Android device.	The device gets added and it can be seen in the devices tab along with all information.
FMS-048						Step 2	Add a device by logging in from a Windows device.	The device gets added and it can be seen in the devices tab along with all information.
FMS-049						Step 3	Add a device by logging in from a Windows device.	The device gets added and it can be seen in the devices tab along with all information.
FMS-050						Step 4	Add a device by logging in from a web browser.	Shows error that a device can not be added from a web browser. It shows previously added devices.

CODING

COMPLETE PROJECT CODING

WINDOWS APP

FMS GUI DESIGN CODING

MainWindow.xaml

```
<Window x:Class="FileManagement.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="File Management" Height="357" Width="765">
    <Window.Resources>
        <ResourceDictionary>
            <ResourceDictionary.MergedDictionaries>
                <ResourceDictionary
Source="/FileManagementStyle;component/ControlStyle.xaml" />
            </ResourceDictionary.MergedDictionaries>
        </ResourceDictionary>
    </Window.Resources>


    <DockPanel LastChildFill="True">
        <UniformGrid DockPanel.Dock="Top">
            <Label Name="lable1" VerticalContentAlignment="Center"
HorizontalContentAlignment="Center" FontFamily="Times New Roman" FontSize="70"
Foreground="#848484"
              Content="File Management" FontWeight="Bold"></Label>
        </UniformGrid>

        <UniformGrid DockPanel.Dock="Top" Rows="1" Background="Gray">
            <Label Style="{StaticResource LblStyle}">Host : </Label>
            <TextBox Style="{StaticResource commonTBtyle}"></TextBox>
            <Label Style="{StaticResource LblStyle}">User ID : </Label>
            <TextBox Style="{StaticResource commonTBtyle}"></TextBox>
            <Label Style="{StaticResource LblStyle}">Password : </Label>
            <PasswordBox Style="{StaticResource commonPBtyle}"></PasswordBox>
            <Label Style="{StaticResource LblStyle}">Port : </Label>
            <TextBox Style="{StaticResource commonTBtyle}"></TextBox>
            <Button Style="{StaticResource PlainBtnStyle}">Connect</Button>
            <Button Style="{StaticResource PlainBtnStyle}">Sync</Button>

        </UniformGrid>

        <DockPanel DockPanel.Dock="Top" LastChildFill="True" >
            <DockPanel DockPanel.Dock="Left" LastChildFill="True" Background="Gray">
                <UniformGrid DockPanel.Dock="Top" Columns="1" Margin="10">
                    <Label Style="{StaticResource LblStyle}">Select Device</Label>
                    <CheckBox  VerticalAlignment="Center" Style="{StaticResource
CheckBtyle}">Desktop</CheckBox>
                    <CheckBox  VerticalAlignment="Center" Style="{StaticResource
CheckBtyle}">Mobile</CheckBox>
                    <CheckBox  VerticalAlignment="Center" Style="{StaticResource
CheckBtyle}">Pendrive</CheckBox>
                    <CheckBox VerticalAlignment="Center" Style="{StaticResource
CheckBtyle}">SD card</CheckBox>
```

```
                    <CheckBox  VerticalAlignment="Center" Style="{StaticResource
CheckBtyle}">CD/DVD</CheckBox>
                </UniformGrid>
                <Label DockPanel.Dock="Bottom"></Label>
            </DockPanel>
            <UniformGrid>
                <ListView Name="productsView"  HorizontalAlignment="Stretch"
ScrollViewer.VerticalScrollBarVisibility="Visible" ItemsSource="{Binding
productsCollection}">


                    <ListView.View>
                        <GridView>
                            <GridViewColumn Width="200" Header="File Name"   />
                            <GridViewColumn Width="70" Header="File Size"   />
                            <GridViewColumn Width="100" Header="File Type"  />
                            <GridViewColumn Width="110" Header="Last Modified"  />
                            <GridViewColumn Width="370" Header="Permissions"  />
                            <GridViewColumn Width="370" Header="Owner/Group"  />

                        </GridView>
                    </ListView.View>
                </ListView>

            </UniformGrid>

        </DockPanel>


    </DockPanel>


</Window>
```

BrowserWindow.xaml

```
<Window x:Class="FileManagement.BrowserWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="File Management System" Height="600" Width="800">
    <Grid>
        <DockPanel LastChildFill="True">
            <Menu DockPanel.Dock="Top">
                <MenuItem Header="File">
                    <MenuItem Header="New Tab"></MenuItem>
                    <MenuItem Header="Close Tab"></MenuItem>
                    <Separator></Separator>
                    <MenuItem Header="Show Command Prompt"></MenuItem>
                    <MenuItem Header="Copy Folder Path"></MenuItem>
                    <MenuItem Header="Copy File Path"></MenuItem>
                    <Separator></Separator>
                    <MenuItem Header="Set File Attribute"></MenuItem>
                    <MenuItem Header="Delete"></MenuItem>
                    <MenuItem Header="Delete Permannently"></MenuItem>
                    <MenuItem Header="Rename"></MenuItem>
                    <MenuItem Header="Properties"></MenuItem>
                    <Separator></Separator>
                    <MenuItem Header="Exit"></MenuItem>
```

```
            </MenuItem> <!--End Of FILE-->

            <MenuItem Header="Edit">
                <MenuItem Header="Undo"></MenuItem>
                <MenuItem Header="Redo"></MenuItem>
                <Separator></Separator>
                <MenuItem Header="Cut"></MenuItem>
                <MenuItem Header="Copy"></MenuItem>
                <MenuItem Header="Paste"></MenuItem>
                <MenuItem Header="Paste Shortcut"></MenuItem>
                <Separator></Separator>
                <MenuItem Header="Copy To Folder"></MenuItem>
                <MenuItem Header="Move To Folder"></MenuItem>
            </MenuItem> <!--End of Menu-->
            <MenuItem Header="Select">
                <MenuItem Header="Select All"></MenuItem>
                <MenuItem Header="Select None"></MenuItem>
                <MenuItem Header="Invert Selection"></MenuItem>
                <MenuItem Header="Select All Of Same Type"></MenuItem>
                <MenuItem Header="Wildcard Select"></MenuItem>
                <MenuItem Header="Wildcard Deselect"></MenuItem>
            </MenuItem> <!--ENd Of Selct-->
            <MenuItem Header="View">
                <MenuItem Header="Information Bar"></MenuItem>
                <MenuItem Header="Folders"></MenuItem>
                <MenuItem Header="Toolbar"></MenuItem>
                <Separator></Separator>
                <MenuItem Header="Extra Large Icon"></MenuItem>
                <MenuItem Header="Large Icon"></MenuItem>
                <MenuItem Header="Medium Icons"></MenuItem>
                <MenuItem Header="Small Icons"></MenuItem>
                <MenuItem Header="List"></MenuItem>
                <MenuItem Header="Tiles"></MenuItem>
                <MenuItem Header="Details"></MenuItem>
                <MenuItem Header="Thumbnails"></MenuItem>
                <Separator></Separator>
                <MenuItem Header="Auto Arrange"></MenuItem>
                <Separator></Separator>
                <MenuItem Header="Group By"></MenuItem>
                <MenuItem Header="Short By"></MenuItem>
                <MenuItem Header="Refresh"></MenuItem>
                <Separator></Separator>
                <MenuItem Header="Show Hidden File"></MenuItem>
                <Separator></Separator>
                <MenuItem Header="Filter"></MenuItem>
            </MenuItem><!--End Of View-->
            <MenuItem Header="Action">

                <MenuItem Header="New Folder"></MenuItem>
                <MenuItem Header="Create Shortcut"></MenuItem>
                <Separator></Separator>
                <MenuItem Header="Split File"></MenuItem>
                <MenuItem Header="Merge File"></MenuItem>
                <MenuItem Header="Destroy File"></MenuItem>
                <MenuItem Header="Upload File"></MenuItem>
                <Separator></Separator>
                <MenuItem Header="Send File">
                    <MenuItem Header="To Removable Disc"></MenuItem>
                    <MenuItem Header="Writable Disc"></MenuItem>
                    <MenuItem Header="Bloothooth Device"></MenuItem>
                    <MenuItem Header="Mail Recipient"></MenuItem>
                    <MenuItem Header="Desktop"></MenuItem>
```

```
                        <MenuItem Header="Zip Folder"></MenuItem>
                    </MenuItem>
                </MenuItem><!--End Of Action-->
                <MenuItem Header="Bookmark">
                    <MenuItem Header="Bookmark This Tab"></MenuItem>
                    <MenuItem Header="Organize Bookmarks"></MenuItem>
                    <Separator></Separator>
                </MenuItem><!--End Of Bookmark-->
                <MenuItem Header="Tool">
                    <MenuItem Header="Search"></MenuItem>
                    <Separator></Separator>
                    <MenuItem Header="Duplicate Finder"></MenuItem>
                    <MenuItem Header="Account Manager"></MenuItem>
                    <MenuItem Header="Customize Folders"></MenuItem>
                    <MenuItem Header="Options"></MenuItem>
                </MenuItem><!--End Of Tool-->
                <MenuItem Header="Go">
                    <MenuItem Header="Back"></MenuItem>

                    <MenuItem Header="Forward"></MenuItem>
                    <MenuItem Header="Up One Level"></MenuItem>
                    <Separator></Separator>
                    <MenuItem Header="My Computer"></MenuItem>
                    <MenuItem Header="Desktop"></MenuItem>
                    <MenuItem Header="My Documents"></MenuItem>
                    <MenuItem Header="My Music"></MenuItem>
                    <MenuItem Header="My Picture"></MenuItem>
                    <Separator></Separator>
                    <MenuItem Header="Control Panel"></MenuItem>

                    <MenuItem Header="Printers"></MenuItem>
                    <MenuItem Header="Network"></MenuItem>
                    <Separator></Separator>
                    <MenuItem Header="Recycle Bin"></MenuItem>
                    <Separator></Separator>
                    <MenuItem Header="Downloads"></MenuItem>
                    <MenuItem Header="Temporary Folder"></MenuItem>
                    <MenuItem Header="Temporary Brun Folder"></MenuItem>
                </MenuItem><!--End Of Go-->
                <MenuItem Header="Help">
                    <MenuItem Header="Help"></MenuItem>
                    <Separator></Separator>
                    <MenuItem Header="About File Manager++"></MenuItem>
                </MenuItem><!--End Of Help-->

            </Menu>
            <ToolBar DockPanel.Dock="Top">
                <Button >Back</Button>
                <Button >Forward</Button>
                <Separator></Separator>
                <Button >Move Up</Button>
                <Separator></Separator>
                <Label>address</Label>
                <TextBox Background="Azure" Width="350"
HorizontalContentAlignment="Center">path</TextBox>
                <Separator></Separator>
                <TextBox Background="Azure" Width="150"
HorizontalContentAlignment="Right">Search files</TextBox>

            </ToolBar>
```

```
            <ToolBar DockPanel.Dock="Top">
                <Button >Create Library</Button>
                <Separator></Separator>
                <Button Name="importLibBtn" >Import Library</Button>
                <Separator></Separator>
                <Button Name="backupBtn" >Backup</Button>
                <Separator></Separator>
                <Button>Refresh Library</Button>
                <Separator></Separator>
                <Button>Starred</Button>
                <Separator></Separator>
                <Button>Settings</Button>
                <Separator></Separator>
                <Button>Upload to Web</Button>
            </ToolBar>

            <StatusBar DockPanel.Dock="Bottom">
                <Label>No files selected</Label>
                <ProgressBar Width="200" Height="20"></ProgressBar>
            </StatusBar>
            <DockPanel DockPanel.Dock="Bottom" LastChildFill="True">
                <DockPanel Name="navigatorSideBar" Width="150" DockPanel.Dock="Left"
Background="#FFF2D9D9"></DockPanel>
                <TextBlock DockPanel.Dock="Bottom" Name="informationBrowserTB"
Background="#FFB8EBB8">Information Browser</TextBlock>
                <!--<ListView Name="mainView" DockPanel.Dock="Top">

                </ListView>-->
                <TabControl DockPanel.Dock="Top">
                    <TabItem Header="tab 1"></TabItem>
                    <TabItem Header="tab 2"></TabItem>
                </TabControl>
            </DockPanel>



        </DockPanel>

    </Grid>
</Window>
```

GUI STYLE: FMSSTYLES

ControlStyle.xaml

```
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

    <Style x:Key="PlainBtnStyle" TargetType="Button">
        <Setter Property="Background" Value="#D8D8D8" />
        <Setter Property="Foreground" Value="Black" />
        <Setter Property="FontWeight" Value="Bold" />
        <Setter Property="BorderBrush" Value="#FFC4C458" />
        <Setter Property="Margin" Value="10"/>
        <Setter Property="OpacityMask" Value="White" />
        <Setter Property="Padding" Value="7" />
        <Setter Property="Template">
```

```
            <Setter.Value>
                <ControlTemplate TargetType="Button">
                    <Border Name="border"
                            BorderThickness="1"
                            Padding="4,2"
                            BorderBrush="White"
                            CornerRadius="0"
                            Background="{TemplateBinding Background}">
                        <ContentPresenter HorizontalAlignment="Center"
VerticalAlignment="Center" />
                    </Border>
                    <ControlTemplate.Triggers>
                        <Trigger Property="IsMouseOver" Value="True">
                            <Setter TargetName="border" Property="BorderBrush"
Value="#2E9AFE" />
                            <Setter Property="Button.Background" Value="#2E9AFE" />
                            <Setter Property="Button.Foreground" Value="White" />
                        </Trigger>
                        <Trigger Property="IsPressed" Value="True">
                            <Setter TargetName="border" Property="BorderBrush"
Value="White" />
                            <Setter Property="Button.Background" Value="#D8D8D8" />
                        </Trigger>

                    </ControlTemplate.Triggers>

                </ControlTemplate>
            </Setter.Value>
        </Setter>
    </Style>

    <Style x:Key="TItemStyle" TargetType="TabItem">
        <Setter Property="Width" Value="115" />
        <Setter Property="Height" Value="33" />
        <Setter Property="Background" Value="#D8D8D8" />
        <Setter Property="BorderBrush" Value="white" />

        <Setter Property="Template">
            <Setter.Value>
                <ControlTemplate TargetType="{x:Type TabItem}">
                    <Grid>
                        <Border Name="Border" Margin="0,0,0,0" Background="#D8D8D8"
                  BorderBrush="#D8D8D8" BorderThickness="1,1,1,1" CornerRadius="0">
                            <ContentPresenter x:Name="ContentSite"
VerticalAlignment="Center"
                              HorizontalAlignment="Center"
                              ContentSource="Header" Margin="12,2,12,2"
                              RecognizesAccessKey="True">
                                <ContentPresenter.LayoutTransform>
                                    <RotateTransform Angle="0" />
                                </ContentPresenter.LayoutTransform>
                            </ContentPresenter>
                        </Border>
                    </Grid>
                    <ControlTemplate.Triggers>
                        <Trigger Property="IsSelected" Value="True">
                            <Setter TargetName="Border" Property="Background"
Value="#2E9AFE" />
                            <Setter Property="Button.Foreground" Value="White" />
                        </Trigger>
                        <Trigger Property="IsMouseOver" Value="True">
                            <Setter TargetName="Border" Property="BorderBrush"
```

```
Value="#2E9AFE" />
                            <Setter TargetName="Border" Property="Background"
Value="#2E9AFE" />
                            <Setter Property="Button.Foreground" Value="White" />
                        </Trigger>
                    </ControlTemplate.Triggers>
                </ControlTemplate>
            </Setter.Value>
        </Setter>
        <Setter Property="HeaderTemplate">
            <Setter.Value>
                <DataTemplate>
                    <Border x:Name="grid" >
                        <ContentPresenter>
                            <ContentPresenter.Content>
                                <TextBlock Margin="4" FontSize="15"
Text="{TemplateBinding Content}"/>
                            </ContentPresenter.Content>
                        </ContentPresenter>

                    </Border>

                </DataTemplate>
            </Setter.Value>
        </Setter>
    </Style>

    <Style x:Key="CheckBtyle" TargetType="CheckBox">
        <Setter Property="Height" Value="32" />
        <Setter Property="FontSize" Value="16" />
        <Setter Property="Width" Value="80" />
        <Setter Property="Background" Value="#FFF9F9EC" />
        <Setter Property="BorderBrush" Value="#FFC4C458" />
        <Setter Property="Foreground" Value="#000000" />
    </Style>

    <Style x:Key="pBstyle" TargetType="PasswordBox">
        <Setter Property="Control.Template" >
            <Setter.Value>
                <ControlTemplate TargetType="{x:Type PasswordBox}">
                    <Border x:Name="border" BorderBrush="{TemplateBinding
BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}"
Background="{TemplateBinding Background}" SnapsToDevicePixels="True">
                        <Grid>
                            <ScrollViewer x:Name="PART_ContentHost" Focusable="False"
HorizontalScrollBarVisibility="Hidden" VerticalScrollBarVisibility="Hidden"/>

                            <TextBlock x:Name="InternalWatermarkLabel"
                       Text="{TemplateBinding Tag}"
                       Visibility="Collapsed" Focusable="False"
                       VerticalAlignment="Top" Margin=" 5 1 0 0"
                       Foreground="Silver"
                       Background="Transparent"/>
                        </Grid>
                    </Border>
                    <ControlTemplate.Triggers>
                        <MultiTrigger>
                            <MultiTrigger.Conditions>
                                <Condition Property="IsFocused" Value="False" />
                            </MultiTrigger.Conditions>
                            <MultiTrigger.Setters>
                                <Setter Property="Visibility"
```

```
TargetName="InternalWatermarkLabel"
                        Value="Visible" />
                            </MultiTrigger.Setters>
                        </MultiTrigger>
                        <Trigger Property="IsEnabled" Value="False">
                            <Setter Property="Opacity" TargetName="border"
Value="0.56"/>
                        </Trigger>
                        <Trigger Property="IsMouseOver" Value="True">
                            <Setter Property="BorderBrush" TargetName="border"
Value="#FF7EB4EA"/>
                        </Trigger>
                        <Trigger Property="IsKeyboardFocused" Value="True">
                            <Setter Property="BorderBrush" TargetName="border"
Value="#FF569DE5"/>
                        </Trigger>
                    </ControlTemplate.Triggers>
                </ControlTemplate>
            </Setter.Value>
        </Setter>
    </Style>

    <Style x:Key="TitlebarBTN" TargetType="Button">
        <Setter Property="OverridesDefaultStyle" Value="True"/>
        <Setter Property="Margin" Value="0"/>
        <Setter Property="FontWeight" Value="Bold" />
        <Setter Property="Template">
            <Setter.Value>
                <ControlTemplate TargetType="Button">
                    <Border Name="border"
                            BorderThickness="1"
                            Padding="4,2"
                            BorderBrush="White"
                            CornerRadius="0"
                            Background="{TemplateBinding Background}">
                        <ContentPresenter HorizontalAlignment="Center"
VerticalAlignment="Center" />
                    </Border>
                    <ControlTemplate.Triggers>
                        <Trigger Property="IsMouseOver" Value="True">
                            <Setter TargetName="border" Property="BorderBrush"
Value="White" />
                            <Setter Property="Button.Background" Value="#D8D8D8" />
                        </Trigger>
                        <Trigger Property="IsPressed" Value="True">
                            <Setter TargetName="border" Property="BorderBrush"
Value="White" />
                            <Setter Property="Button.Background" Value="#2E9AFE" />
                        </Trigger>
                    </ControlTemplate.Triggers>
                </ControlTemplate>
            </Setter.Value>
        </Setter>
    </Style>

    <Style x:Key="{x:Type ToolTip}" TargetType="{x:Type ToolTip}">
        <Setter Property="Background" Value="Black"/>
        <Setter Property="BorderBrush" Value="#D8D8D8"/>
        <Setter Property="Foreground" Value="White" />
        <Setter Property="Padding" Value="7" />
        <Setter Property="FontWeight" Value="Bold" />
```

```
    </Style>

    <Style x:Key="commonPBtyle" TargetType="PasswordBox">
        <Setter Property="VerticalContentAlignment" Value="Center" />
        <Setter Property="HorizontalContentAlignment" Value="Left" />
        <Setter Property="FontWeight" Value="Bold" />
    </Style>

    <Style x:Key="commonTBtyle" TargetType="TextBox">
        <Setter Property="VerticalContentAlignment" Value="Center" />
        <Setter Property="HorizontalContentAlignment" Value="Left" />
        <Setter Property="FontWeight" Value="Bold" />

        <Setter Property="Control.Template" >
            <Setter.Value>
                <ControlTemplate TargetType="{x:Type TextBox}">
                    <Border x:Name="border" BorderBrush="{TemplateBinding
BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}"
Background="{TemplateBinding Background}" SnapsToDevicePixels="True">
                        <Grid>
                            <ScrollViewer x:Name="PART_ContentHost" Focusable="False"
HorizontalScrollBarVisibility="Hidden" VerticalScrollBarVisibility="Hidden"/>

                            <TextBlock x:Name="InternalWatermarkLabel"
                       Text="{TemplateBinding Tag}"
                       Visibility="Collapsed" Focusable="False"
                       VerticalAlignment="Top" Margin=" 5 1 0 0"
                       Foreground="Silver"
                       Background="Transparent"/>
                        </Grid>
                    </Border>
                    <ControlTemplate.Triggers>
                        <MultiTrigger>
                            <MultiTrigger.Conditions>
                                <Condition Property="IsFocused" Value="False" />
                                <Condition Property="Text" Value="" />
                            </MultiTrigger.Conditions>
                            <MultiTrigger.Setters>
                                <Setter Property="Visibility"
TargetName="InternalWatermarkLabel"
                        Value="Visible" />
                            </MultiTrigger.Setters>
                        </MultiTrigger>
                        <Trigger Property="IsEnabled" Value="False">
                            <Setter Property="Opacity" TargetName="border"
Value="0.56"/>
                        </Trigger>
                        <Trigger Property="IsMouseOver" Value="True">
                            <Setter Property="BorderBrush" TargetName="border"
Value="#FF7EB4EA"/>

                        </Trigger>
                        <Trigger Property="IsKeyboardFocused" Value="True">
                            <Setter Property="BorderBrush" TargetName="border"
Value="#FF569DE5"/>
                        </Trigger>
                    </ControlTemplate.Triggers>
                </ControlTemplate>
            </Setter.Value>
        </Setter>

    </Style>
```

```
    <Style TargetType="{x:Type ScrollBar}">
        <Setter Property="Stylus.IsFlicksEnabled" Value="True" />
        <Setter Property="Background" Value="#D8D8D8" />
        <Setter Property="Width" Value="15"/>
        <Setter Property="MinWidth" Value="8" />
        <Setter Property="Template">

            <Setter.Value>
                <ControlTemplate TargetType="{x:Type ScrollBar}">
                    <Grid x:Name="GridRoot" Width="19" Background="{TemplateBinding
Background}">
                        <Grid.RowDefinitions>
                            <RowDefinition Height="0.00001*" />
                        </Grid.RowDefinitions>

                        <Track x:Name="PART_Track" Grid.Row="0"
IsDirectionReversed="true" Focusable="false">
                            <Track.Thumb>
                                <Thumb x:Name="Thumb" Background="Gray"
Style="{DynamicResource ScrollThumbs}" />
                            </Track.Thumb>
                            <Track.IncreaseRepeatButton>
                                <RepeatButton x:Name="PageUp"
Command="ScrollBar.PageDownCommand" Opacity="0" Focusable="false" />
                            </Track.IncreaseRepeatButton>
                            <Track.DecreaseRepeatButton>
                                <RepeatButton x:Name="PageDown"
Command="ScrollBar.PageUpCommand" Opacity="0" Focusable="false" />
                            </Track.DecreaseRepeatButton>
                        </Track>
                    </Grid>

                    <ControlTemplate.Triggers>
                        <Trigger SourceName="Thumb" Property="IsMouseOver"
Value="true">
                            <Setter Value="#2E9AFE" TargetName="Thumb"
Property="Background" />
                        </Trigger>
                        <Trigger SourceName="Thumb" Property="IsDragging"
Value="true">
                            <Setter Value="#D8D8D8" TargetName="Thumb"
Property="Background" />
                        </Trigger>

                        <Trigger Property="IsEnabled" Value="false">
                            <Setter TargetName="Thumb" Property="Visibility"
Value="Collapsed" />
                        </Trigger>
                        <Trigger Property="Orientation" Value="Horizontal">
                            <Setter TargetName="GridRoot" Property="LayoutTransform">
                                <Setter.Value>
                                    <RotateTransform Angle="-90" />
                                </Setter.Value>
                            </Setter>
                            <Setter TargetName="PART_Track"
Property="LayoutTransform">
                                <Setter.Value>
                                    <RotateTransform Angle="-90" />
                                </Setter.Value>
                            </Setter>
                            <Setter Property="Width" Value="Auto" />
```

```
                            <Setter Property="Height" Value="15" />
                            <Setter TargetName="Thumb" Property="Tag"
Value="Horizontal" />
                            <Setter TargetName="PageDown" Property="Command"
Value="ScrollBar.PageLeftCommand" />
                            <Setter TargetName="PageUp" Property="Command"
Value="ScrollBar.PageRightCommand" />
                        </Trigger>
                    </ControlTemplate.Triggers>
                </ControlTemplate>
            </Setter.Value>
        </Setter>
    </Style>

    <Style x:Key="LblStyle" TargetType="Label">
        <Setter Property="VerticalContentAlignment" Value="Center" />
        <Setter Property="HorizontalContentAlignment" Value="Right" />
        <Setter Property="Foreground" Value="Black" />
        <Setter Property="FontWeight" Value="Bold" />

    </Style>
    <Style x:Key="TxtblkStyle" TargetType="TextBlock">
        <Setter Property="VerticalAlignment" Value="Center" />
        <Setter Property="HorizontalAlignment" Value="Center" />
        <Setter Property="Foreground" Value="Black" />
        <Setter Property="FontWeight" Value="Bold" />

    </Style>

    <Style x:Key="DatepkrStyle" TargetType="DatePicker">
        <Setter Property="VerticalAlignment" Value="Center" />
        <Setter Property="HorizontalAlignment" Value="Left" />
        <Setter Property="Foreground" Value="Black" />
        <Setter Property="FontWeight" Value="Bold" />

    </Style>


</ResourceDictionary>
```

DATA CLASSES : FMSDATA

UserInfo

```
public class UserInfo
    {
        public string userId;
        public string userName;
        public List< StorageDeviceInfo> deviceOwned;
        public List<FileInfo> files;
    }
```

FileBackupInfo

```
public class FileBackupInfo
    {
        public string backupId;
        public DateTime backupDate;
        public List< HostAppDeviceInfo > usedHosts;
        public List< StorageDeviceInfo >sourceDevices;
        public List<FileInfo> files;
    }
```

FileStreamInfo

```
public class FileStreamInfo
    {
        public string streamId;
        public DateTime streamDate;
        public HostAppDeviceInfo usedHost;
        public StorageDeviceInfo sourceDevice;
        public string destinationIP;
        public List<FileInfo> files;
    }
```

StorageDeviceInfo

```
public class StorageDeviceInfo
    {
        public string deviceId;
        public DateTime lastScannedDate;
        public HostAppDeviceInfo lastUsedHost;
        public List<FileInfo> files;
        public List<FileBackupInfo> filesBackupHere;
    }
```

HostAppType

```
public enum HostAppType
    {
        Android,
        Win8,
        Win7,
```

```
        ubuntu
    }
```

HostAppDeviceInfo

```
public class HostAppDeviceInfo
    {
        public string hostDeviceId;
        public List<StorageDeviceInfo> storageDevicesAttached;
        public List<FileStreamInfo> filesStreamed;
        public HostAppType type;
    }
```

FileInfo

```
public class FileInfo
    {
        public string fileName;
        public string filePath;
        public StorageDeviceInfo device;
        public string hostAppId;
        public string fileSize;
        public string fileType;
        public string creationDate;
        public string modifyDate;

        public List<FileBackupInfo> backups;
        public List<FileStreamInfo> streams;
    }
```

FMSServerInfo

```
public class FMSServerInfo
    {
        public string serverId;
        public List<StorageDeviceInfo> storageDevicesAttached;
        public List<HostAppDeviceInfo> hostDevicesAttached;
        public List<FileStreamInfo> filesStreamed;
        public List<FileBackupInfo> fileBackedup;
```

```
    }
```

DATABASE CONNECTOR: FMSDB

DbInteraction.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace FMSClassLibrary
{

    public class DbInteraction
    {
        static string passwordCurrent = "technicise";
        static string dbmsCurrent = "FMSdb";

        private static MySql.Data.MySqlClient.MySqlConnection OpenDbConnection()
        {
            MySql.Data.MySqlClient.MySqlConnection msqlConnection = null;

            msqlConnection = new
MySql.Data.MySqlClient.MySqlConnection("server=localhost;user id=root;Password=" +
passwordCurrent + ";database=" + dbmsCurrent + ";persist security info=False");

            //open the connection
            if (msqlConnection.State != System.Data.ConnectionState.Open)
                msqlConnection.Open();

            return msqlConnection;
        }

        #region User

        public static int DoRegisterNewUser(UserInfo NewUser)
        {
            return DoRegisterNewuserindb(NewUser);
        }

        private static int DoRegisterNewuserindb(UserInfo NewUser)
        {
            int returnVal = 0;
            MySql.Data.MySqlClient.MySqlConnection msqlConnection =
OpenDbConnection();

            try
            {
                //define the command reference
                MySql.Data.MySqlClient.MySqlCommand msqlCommand = new
MySql.Data.MySqlClient.MySqlCommand();

                //define the connection used by the command object
                msqlCommand.Connection = msqlConnection;

                msqlCommand.CommandText = "INSERT INTO user(id,userid,passwrd,hints) "
```

```
+ "VALUES(@id,@userid,@passwrd,@hints)";

                msqlCommand.Parameters.AddWithValue("@id", NewUser.userId);
                msqlCommand.Parameters.AddWithValue("@userid", NewUser.userId);
                msqlCommand.Parameters.AddWithValue("@passwrd", NewUser.userName);
                msqlCommand.Parameters.AddWithValue("@hints", NewUser.deviceOwned);


                msqlCommand.ExecuteNonQuery();

                returnVal = 1;
            }
            catch (Exception er)
            {
                returnVal = 0;
            }
            finally
            {
                //always close the connection
                msqlConnection.Close();
            }
            return returnVal;
        }

        #endregion

        #region ID password

        public static string FetcheId()
        {

            string idStr = string.Empty;

            int returnVal = 0;
            MySql.Data.MySqlClient.MySqlConnection msqlConnection =
OpenDbConnection();

            try
            {


                //define the command reference
                MySql.Data.MySqlClient.MySqlCommand msqlCommand = new
MySql.Data.MySqlClient.MySqlCommand();

                //define the connection used by the command object
                msqlCommand.Connection = msqlConnection;


                msqlCommand.CommandText = "Select userid from user;";
                MySql.Data.MySqlClient.MySqlDataReader msqlReader =
msqlCommand.ExecuteReader();

                msqlReader.Read();

                idStr = msqlReader.GetString("userid");

            }
            catch (Exception er)
            {
                //Assert//.Show(er.Message);
            }
```

```
            finally
            {
                //always close the connection
                msqlConnection.Close();
            }

            return idStr;
        }

        public static string FetchePassword()
        {

            string passwordStr = string.Empty;

            int returnVal = 0;
            MySql.Data.MySqlClient.MySqlConnection msqlConnection =
OpenDbConnection();

            try
            {


                //define the command reference
                MySql.Data.MySqlClient.MySqlCommand msqlCommand = new
MySql.Data.MySqlClient.MySqlCommand();

                //define the connection used by the command object
                msqlCommand.Connection = msqlConnection;


                msqlCommand.CommandText = "Select passwrd from user;";
                MySql.Data.MySqlClient.MySqlDataReader msqlReader =
msqlCommand.ExecuteReader();

                msqlReader.Read();

                passwordStr = msqlReader.GetString("passwrd");

            }
            catch (Exception er)
            {
                //Assert//.Show(er.Message);
            }
            finally
            {
                //always close the connection
                msqlConnection.Close();
            }

            return passwordStr;
        }
        #endregion

        #region Device


        public static int DoRegisterNewDevice(StorageDeviceInfo DeviceDetails)
        {
            int returnVal = 0;
            MySql.Data.MySqlClient.MySqlConnection msqlConnection =
OpenDbConnection();
```

```
            try
            {
                //define the command reference
                MySql.Data.MySqlClient.MySqlCommand msqlCommand = new
MySql.Data.MySqlClient.MySqlCommand();

                //define the connection used by the command object
                msqlCommand.Connection = msqlConnection;

                msqlCommand.CommandText = "INSERT INTO
Device(DeviceId,lastScannedDate,lastUsedHost,files,remark) "
                                                        +
"VALUES(@DeviceId,@lastScannedDate,@lastUsedHost,@files,@filesBackupHere,@remark)";

                msqlCommand.Parameters.AddWithValue("@DeviceId",
DeviceDetails.deviceId);
                msqlCommand.Parameters.AddWithValue("@files", DeviceDetails.files);
                msqlCommand.Parameters.AddWithValue("@lastScannedDate",
DeviceDetails.lastScannedDate);
                msqlCommand.Parameters.AddWithValue("@lastUsedHost",
DeviceDetails.lastUsedHost);

                msqlCommand.ExecuteNonQuery();

                returnVal = 1;
            }
            catch (Exception er)
            {
                returnVal = 0;
            }
            finally
            {
                //always close the connection
                msqlConnection.Close();
            }
            return returnVal;
        }


        public static List<StorageDeviceInfo> GetAllDeviceList()
        {
            return QueryAllDeviceList();
        }
        private static List<StorageDeviceInfo> QueryAllDeviceList()
        {
            List<StorageDeviceInfo> DeviceList = new List<StorageDeviceInfo>();

            MySql.Data.MySqlClient.MySqlConnection msqlConnection =
OpenDbConnection();

            try
            {   //define the command reference
                MySql.Data.MySqlClient.MySqlCommand msqlCommand = new
MySql.Data.MySqlClient.MySqlCommand();
                msqlCommand.Connection = msqlConnection;

                msqlCommand.CommandText = "Select * From Device ;";
                MySql.Data.MySqlClient.MySqlDataReader msqlReader =
msqlCommand.ExecuteReader();

                while (msqlReader.Read())
                {
```

```
                    StorageDeviceInfo Device = new StorageDeviceInfo();

                    /*
                    Device.deviceId = msqlReader.GetString("DeviceId");
                    Device.lastScannedDate = msqlReader.GetString("lastScannedDate");
                    Device.lastUsedHost = msqlReader.GetString("lastUsedHost");
                    Device.lastUsedHost = msqlReader.GetString("lastUsedHost");
                    Device.files = msqlReader.GetString("files");
                    */

                    DeviceList.Add(Device);
                }

            }
            catch (Exception er)
            {
            }
            finally
            {
                //always close the connection
                msqlConnection.Close();
            }

            return DeviceList;
        }

        #endregion


    }
}
```

LINUX APP

FileManager

```
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.Desktop;
import java.awt.Dimension;
import java.awt.Container;
import java.awt.Component;
import java.awt.Graphics;
import java.awt.Image;
import java.awt.event.*;
import java.awt.image.*;

import javax.swing.*;
import javax.swing.border.*;
```

```
import javax.swing.event.*;
import javax.swing.tree.*;
import javax.swing.table.*;
import javax.swing.filechooser.FileSystemView;

import javax.imageio.ImageIO;

import java.util.Date;
import java.util.List;
import java.util.ArrayList;

import java.io.*;
import java.nio.channels.FileChannel;

import java.net.URL;

/**
A basic File Manager.  Requires 1.6+ for the Desktop & SwingWorker
classes, amongst other minor things.

Includes support classes FileTableModel & FileTreeCellRenderer.

@TODO Bugs
<li>Still throws occasional AIOOBEs and NPEs, so some update on
the EDT must have been missed.
<li>Fix keyboard focus issues - especially when functions like
rename/delete etc. are called that update nodes & file lists.
<li>Needs more testing in general.

@TODO Functionality
<li>Implement Read/Write/Execute checkboxes
<li>Implement Copy
<li>Extra prompt for directory delete (camickr suggestion)
<li>Add File/Directory fields to FileTableModel
<li>Double clicking a directory in the table, should update the tree
<li>Move progress bar?
<li>Add other file display modes (besides table) in CardLayout?
<li>Menus + other cruft?
<li>Implement history/back
<li>Allow multiple selection
```

```
<li>Add file search


*/
class FileManager {

    /** Title of the application */
    public static final String APP_TITLE = "FileMan";
    /** Used to open/edit/print files. */
    private Desktop desktop;
    /** Provides nice icons and names for files. */
    private FileSystemView fileSystemView;

    /** currently selected File. */
    private File currentFile;

    /** Main GUI container */
    private JPanel gui;

    /** File-system tree. Built Lazily */
    private JTree tree;
    private DefaultTreeModel treeModel;

    /** Directory listing */
    private JTable table;
    private JProgressBar progressBar;
    /** Table model for File[]. */
    private FileTableModel fileTableModel;
    private ListSelectionListener listSelectionListener;
    private boolean cellSizesSet = false;
    private int rowIconPadding = 6;

    /* File controls. */
    private JButton openFile;
    private JButton printFile;
    private JButton editFile;
    private JButton deleteFile;
    private JButton newFile;
    private JButton copyFile;
    /* File details. */
```

```
private JLabel fileName;
private JTextField path;
private JLabel date;
private JLabel size;
private JCheckBox readable;
private JCheckBox writable;
private JCheckBox executable;
private JRadioButton isDirectory;
private JRadioButton isFile;

/* GUI options/containers for new File/Directory creation.  Created lazily. */
private JPanel newFilePanel;
private JRadioButton newTypeFile;
private JTextField name;

public Container getGui() {
   if (gui==null) {
      gui = new JPanel(new BorderLayout(3,3));
      gui.setBorder(new EmptyBorder(5,5,5,5));

      fileSystemView = FileSystemView.getFileSystemView();
      desktop = Desktop.getDesktop();

      JPanel detailView = new JPanel(new BorderLayout(3,3));
      //fileTableModel = new FileTableModel();

      table = new JTable();
      table.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
      table.setAutoCreateRowSorter(true);
      table.setShowVerticalLines(false);

      listSelectionListener = new ListSelectionListener() {
         @Override
         public void valueChanged(ListSelectionEvent lse) {
            int row = table.getSelectionModel().getLeadSelectionIndex();
            setFileDetails( ((FileTableModel)table.getModel()).getFile(row) );
         }
      };
      table.getSelectionModel().addListSelectionListener(listSelectionListener);
      JScrollPane tableScroll = new JScrollPane(table);
```

```
Dimension d = tableScroll.getPreferredSize();
tableScroll.setPreferredSize(new Dimension((int)d.getWidth(), (int)d.getHeight()/2));
detailView.add(tableScroll, BorderLayout.CENTER);

// the File tree
DefaultMutableTreeNode root = new DefaultMutableTreeNode();
treeModel = new DefaultTreeModel(root);

TreeSelectionListener treeSelectionListener = new TreeSelectionListener() {
   public void valueChanged(TreeSelectionEvent tse){
      DefaultMutableTreeNode node =
         (DefaultMutableTreeNode)tse.getPath().getLastPathComponent();
      showChildren(node);
      setFileDetails((File)node.getUserObject());
   }
};

// show the file system roots.
File[] roots = fileSystemView.getRoots();
for (File fileSystemRoot : roots) {
   DefaultMutableTreeNode node = new DefaultMutableTreeNode(fileSystemRoot);
   root.add( node );
   //showChildren(node);
   //
   File[] files = fileSystemView.getFiles(fileSystemRoot, true);
   for (File file : files) {
      if (file.isDirectory()) {
         node.add(new DefaultMutableTreeNode(file));
      }
   }
   //
}

tree = new JTree(treeModel);
tree.setRootVisible(false);
tree.addTreeSelectionListener(treeSelectionListener);
tree.setCellRenderer(new FileTreeCellRenderer());
tree.expandRow(0);
JScrollPane treeScroll = new JScrollPane(tree);
```

```
// as per trashgod tip
tree.setVisibleRowCount(15);

Dimension preferredSize = treeScroll.getPreferredSize();
Dimension widePreferred = new Dimension(
   200,
   (int)preferredSize.getHeight());
treeScroll.setPreferredSize( widePreferred );

// details for a File
JPanel fileMainDetails = new JPanel(new BorderLayout(4,2));
fileMainDetails.setBorder(new EmptyBorder(0,6,0,6));

JPanel fileDetailsLabels = new JPanel(new GridLayout(0,1,2,2));
fileMainDetails.add(fileDetailsLabels, BorderLayout.WEST);

JPanel fileDetailsValues = new JPanel(new GridLayout(0,1,2,2));
fileMainDetails.add(fileDetailsValues, BorderLayout.CENTER);

fileDetailsLabels.add(new JLabel("File", JLabel.TRAILING));
fileName = new JLabel();
fileDetailsValues.add(fileName);
fileDetailsLabels.add(new JLabel("Path/name", JLabel.TRAILING));
path = new JTextField(5);
path.setEditable(false);
fileDetailsValues.add(path);
fileDetailsLabels.add(new JLabel("Last Modified", JLabel.TRAILING));
date = new JLabel();
fileDetailsValues.add(date);
fileDetailsLabels.add(new JLabel("File size", JLabel.TRAILING));
size = new JLabel();
fileDetailsValues.add(size);
fileDetailsLabels.add(new JLabel("Type", JLabel.TRAILING));

JPanel flags = new JPanel(new FlowLayout(FlowLayout.LEADING,4,0));
isDirectory = new JRadioButton("Directory");
isDirectory.setEnabled(false);
flags.add(isDirectory);

isFile = new JRadioButton("File");
```

```
isFile.setEnabled(false);
flags.add(isFile);
fileDetailsValues.add(flags);

int count = fileDetailsLabels.getComponentCount();
for (int ii=0; ii<count; ii++) {
   fileDetailsLabels.getComponent(ii).setEnabled(false);
}

JToolBar toolBar = new JToolBar();
// mnemonics stop working in a floated toolbar
toolBar.setFloatable(false);

openFile = new JButton("Open");
openFile.setMnemonic('o');

openFile.addActionListener(new ActionListener(){
   public void actionPerformed(ActionEvent ae) {
      try {
         desktop.open(currentFile);
      } catch(Throwable t) {
         showThrowable(t);
      }
      gui.repaint();
   }
});
toolBar.add(openFile);

editFile = new JButton("Edit");
editFile.setMnemonic('e');
editFile.addActionListener(new ActionListener(){
   public void actionPerformed(ActionEvent ae) {
      try {
         desktop.edit(currentFile);
      } catch(Throwable t) {
         showThrowable(t);
      }
   }
});
toolBar.add(editFile);
```

```
printFile = new JButton("Print");
printFile.setMnemonic('p');
printFile.addActionListener(new ActionListener(){
   public void actionPerformed(ActionEvent ae) {
      try {
         desktop.print(currentFile);
      } catch(Throwable t) {
         showThrowable(t);
      }
   }
});
toolBar.add(printFile);

// Check the actions are supported on this platform!
openFile.setEnabled(desktop.isSupported(Desktop.Action.OPEN));
editFile.setEnabled(desktop.isSupported(Desktop.Action.EDIT));
printFile.setEnabled(desktop.isSupported(Desktop.Action.PRINT));

toolBar.addSeparator();

newFile = new JButton("New");
newFile.setMnemonic('n');
newFile.addActionListener(new ActionListener(){
   public void actionPerformed(ActionEvent ae) {
      newFile();
   }
});
toolBar.add(newFile);

copyFile = new JButton("Copy");
copyFile.setMnemonic('c');
copyFile.addActionListener(new ActionListener(){
   public void actionPerformed(ActionEvent ae) {
      showErrorMessage("'Copy' not implemented.", "Not implemented.");
   }
});
toolBar.add(copyFile);

JButton renameFile = new JButton("Rename");
```

```
renameFile.setMnemonic('r');
renameFile.addActionListener(new ActionListener(){
   public void actionPerformed(ActionEvent ae) {
      renameFile();
   }
});
toolBar.add(renameFile);

deleteFile = new JButton("Delete");
deleteFile.setMnemonic('d');
deleteFile.addActionListener(new ActionListener(){
   public void actionPerformed(ActionEvent ae) {
      deleteFile();
   }
});
toolBar.add(deleteFile);

toolBar.addSeparator();

readable = new JCheckBox("Read  ");
readable.setMnemonic('a');
//readable.setEnabled(false);
toolBar.add(readable);

writable = new JCheckBox("Write  ");
writable.setMnemonic('w');
//writable.setEnabled(false);
toolBar.add(writable);

executable = new JCheckBox("Execute");
executable.setMnemonic('x');
//executable.setEnabled(false);
toolBar.add(executable);

JPanel fileView = new JPanel(new BorderLayout(3,3));

fileView.add(toolBar,BorderLayout.NORTH);
fileView.add(fileMainDetails,BorderLayout.CENTER);

detailView.add(fileView, BorderLayout.SOUTH);
```

```
        JSplitPane splitPane = new JSplitPane(
            JSplitPane.HORIZONTAL_SPLIT,
            treeScroll,
            detailView);
        gui.add(splitPane, BorderLayout.CENTER);

        JPanel simpleOutput = new JPanel(new BorderLayout(3,3));
        progressBar = new JProgressBar();
        simpleOutput.add(progressBar, BorderLayout.EAST);
        progressBar.setVisible(false);

        gui.add(simpleOutput, BorderLayout.SOUTH);

    }
    return gui;
}

public void showRootFile() {
    // ensure the main files are displayed
    tree.setSelectionInterval(0,0);
}

private TreePath findTreePath(File find) {
    for (int ii=0; ii<tree.getRowCount(); ii++) {
        TreePath treePath = tree.getPathForRow(ii);
        Object object = treePath.getLastPathComponent();
        DefaultMutableTreeNode node = (DefaultMutableTreeNode)object;
        File nodeFile = (File)node.getUserObject();

        if (nodeFile==find) {
            return treePath;
        }
    }
    // not found!
    return null;
}

private void renameFile() {
    if (currentFile==null) {
```

```
        showErrorMessage("No file selected to rename.","Select File");
        return;
    }

    String renameTo = JOptionPane.showInputDialog(gui, "New Name");
    if (renameTo!=null) {
        try {
            boolean directory = currentFile.isDirectory();
            TreePath parentPath = findTreePath(currentFile.getParentFile());
            DefaultMutableTreeNode parentNode =
                (DefaultMutableTreeNode)parentPath.getLastPathComponent();

            boolean renamed = currentFile.renameTo(new File(
                currentFile.getParentFile(), renameTo));
            if (renamed) {
                if (directory) {
                    // rename the node..

                    // delete the current node..
                    TreePath currentPath = findTreePath(currentFile);
                    System.out.println(currentPath);
                    DefaultMutableTreeNode currentNode =
                        (DefaultMutableTreeNode)currentPath.getLastPathComponent();

                    treeModel.removeNodeFromParent(currentNode);

                    // add a new node..
                }

                showChildren(parentNode);
            } else {
                String msg = "The file '" +
                    currentFile +
                    "' could not be renamed.";
                showErrorMessage(msg,"Rename Failed");
            }
        } catch(Throwable t) {
            showThrowable(t);
        }
    }
```

```
    gui.repaint();
}

private void deleteFile() {
    if (currentFile==null) {
        showErrorMessage("No file selected for deletion.","Select File");
        return;
    }

    int result = JOptionPane.showConfirmDialog(
        gui,
        "Are you sure you want to delete this file?",
        "Delete File",
        JOptionPane.ERROR_MESSAGE
        );
    if (result==JOptionPane.OK_OPTION) {
        try {
            System.out.println("currentFile: " + currentFile);
            TreePath parentPath = findTreePath(currentFile.getParentFile());
            System.out.println("parentPath: " + parentPath);
            DefaultMutableTreeNode parentNode =
                (DefaultMutableTreeNode)parentPath.getLastPathComponent();
            System.out.println("parentNode: " + parentNode);

            boolean directory = currentFile.isDirectory();
            boolean deleted = currentFile.delete();
            if (deleted) {
                if (directory) {
                    // delete the node..
                    TreePath currentPath = findTreePath(currentFile);
                    System.out.println(currentPath);
                    DefaultMutableTreeNode currentNode =
                        (DefaultMutableTreeNode)currentPath.getLastPathComponent();

                    treeModel.removeNodeFromParent(currentNode);
                }

                showChildren(parentNode);
            } else {
                String msg = "The file '" +
```

```
                    currentFile +
                    '" could not be deleted.";
                showErrorMessage(msg,"Delete Failed");
            }
        } catch(Throwable t) {
            showThrowable(t);
        }
    }
    gui.repaint();
}

private void newFile() {
    if (currentFile==null) {
        showErrorMessage("No location selected for new file.","Select Location");
        return;
    }

    if (newFilePanel==null) {
        newFilePanel = new JPanel(new BorderLayout(3,3));

        JPanel southRadio = new JPanel(new GridLayout(1,0,2,2));
        newTypeFile = new JRadioButton("File", true);
        JRadioButton newTypeDirectory = new JRadioButton("Directory");
        ButtonGroup bg = new ButtonGroup();
        bg.add(newTypeFile);
        bg.add(newTypeDirectory);
        southRadio.add( newTypeFile );
        southRadio.add( newTypeDirectory );

        name = new JTextField(15);

        newFilePanel.add( new JLabel("Name"), BorderLayout.WEST );
        newFilePanel.add( name );
        newFilePanel.add( southRadio, BorderLayout.SOUTH );
    }

    int result = JOptionPane.showConfirmDialog(
        gui,
        newFilePanel,
        "Create File",
```

```
            JOptionPane.OK_CANCEL_OPTION);
        if (result==JOptionPane.OK_OPTION) {
            try {
                boolean created;
                File parentFile = currentFile;
                if (!parentFile.isDirectory()) {
                    parentFile = parentFile.getParentFile();
                }
                File file = new File( parentFile, name.getText() );
                if (newTypeFile.isSelected()) {
                    created = file.createNewFile();
                } else {
                    created = file.mkdir();
                }
                if (created) {

                    TreePath parentPath = findTreePath(parentFile);
                    DefaultMutableTreeNode parentNode =
                        (DefaultMutableTreeNode)parentPath.getLastPathComponent();

                    if (file.isDirectory()) {
                        // add the new node..
                        DefaultMutableTreeNode newNode = new DefaultMutableTreeNode(file);

                        TreePath currentPath = findTreePath(currentFile);
                        DefaultMutableTreeNode currentNode =
                            (DefaultMutableTreeNode)currentPath.getLastPathComponent();

                        treeModel.insertNodeInto(newNode, parentNode,
parentNode.getChildCount());
                    }

                    showChildren(parentNode);
                } else {
                    String msg = "The file '" +
                        file +
                        "' could not be created.";
                    showErrorMessage(msg, "Create Failed");
                }
            } catch(Throwable t) {
```

```
            showThrowable(t);
        }
    }
    gui.repaint();
}

private void showErrorMessage(String errorMessage, String errorTitle) {
    JOptionPane.showMessageDialog(
        gui,
        errorMessage,
        errorTitle,
        JOptionPane.ERROR_MESSAGE
        );
}

private void showThrowable(Throwable t) {
    t.printStackTrace();
    JOptionPane.showMessageDialog(
        gui,
        t.toString(),
        t.getMessage(),
        JOptionPane.ERROR_MESSAGE
        );
    gui.repaint();
}

/** Update the table on the EDT */
private void setTableData(final File[] files) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            if (fileTableModel==null) {
                fileTableModel = new FileTableModel();
                table.setModel(fileTableModel);
            }
            table.getSelectionModel().removeListSelectionListener(listSelectionListener);
            fileTableModel.setFiles(files);
            table.getSelectionModel().addListSelectionListener(listSelectionListener);
            if (!cellSizesSet) {
                Icon icon = fileSystemView.getSystemIcon(files[0]);
```

```
            // size adjustment to better account for icons
            table.setRowHeight( icon.getIconHeight()+rowIconPadding );

            setColumnWidth(0,-1);
            setColumnWidth(3,60);
            table.getColumnModel().getColumn(3).setMaxWidth(120);
            setColumnWidth(4,-1);
            setColumnWidth(5,-1);
            setColumnWidth(6,-1);
            setColumnWidth(7,-1);
            setColumnWidth(8,-1);
            setColumnWidth(9,-1);

            cellSizesSet = true;
          }
        }
    });
}

private void setColumnWidth(int column, int width) {
    TableColumn tableColumn = table.getColumnModel().getColumn(column);
    if (width<0) {
        // use the preferred width of the header..
        JLabel label = new JLabel( (String)tableColumn.getHeaderValue() );
        Dimension preferred = label.getPreferredSize();
        // altered 10->14 as per camickr comment.
        width = (int)preferred.getWidth()+14;
    }
    tableColumn.setPreferredWidth(width);
    tableColumn.setMaxWidth(width);
    tableColumn.setMinWidth(width);
}

/** Add the files that are contained within the directory of this node.
Thanks to Hovercraft Full Of Eels. */
private void showChildren(final DefaultMutableTreeNode node) {
    tree.setEnabled(false);
    progressBar.setVisible(true);
    progressBar.setIndeterminate(true);
```

```
    SwingWorker<Void, File> worker = new SwingWorker<Void, File>() {
        @Override
        public Void doInBackground() {
            File file = (File) node.getUserObject();
            if (file.isDirectory()) {
                File[] files = fileSystemView.getFiles(file, true); //!!
                if (node.isLeaf()) {
                    for (File child : files) {
                        if (child.isDirectory()) {
                            publish(child);
                        }
                    }
                }
                setTableData(files);
            }
            return null;
        }

        @Override
        protected void process(List<File> chunks) {
            for (File child : chunks) {
                node.add(new DefaultMutableTreeNode(child));
            }
        }

        @Override
        protected void done() {
            progressBar.setIndeterminate(false);
            progressBar.setVisible(false);
            tree.setEnabled(true);
        }
    };
    worker.execute();
}

/** Update the File details view with the details of this File. */
private void setFileDetails(File file) {
    currentFile = file;
    Icon icon = fileSystemView.getSystemIcon(file);
    fileName.setIcon(icon);
```

```
    fileName.setText(fileSystemView.getSystemDisplayName(file));
    path.setText(file.getPath());
    date.setText(new Date(file.lastModified()).toString());
    size.setText(file.length() + " bytes");
    readable.setSelected(file.canRead());
    writable.setSelected(file.canWrite());
    executable.setSelected(file.canExecute());
    isDirectory.setSelected(file.isDirectory());

    isFile.setSelected(file.isFile());

    JFrame f = (JFrame)gui.getTopLevelAncestor();
    if (f!=null) {
      f.setTitle(
        APP_TITLE +
        " :: " +
        fileSystemView.getSystemDisplayName(file) );
    }

    gui.repaint();
  }

  public static boolean copyFile(File from, File to) throws IOException {

    boolean created = to.createNewFile();

    if (created) {
      FileChannel fromChannel = null;
      FileChannel toChannel = null;
      try {
        fromChannel = new FileInputStream(from).getChannel();
        toChannel = new FileOutputStream(to).getChannel();

        toChannel.transferFrom(fromChannel, 0, fromChannel.size());

        // set the flags of the to the same as the from
        to.setReadable(from.canRead());
        to.setWritable(from.canWrite());
        to.setExecutable(from.canExecute());
      } finally {
```

```
            if (fromChannel != null) {
                fromChannel.close();
            }
            if (toChannel != null) {
                toChannel.close();
            }
            return false;
        }
    }
    return created;
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            try {
                // Significantly improves the look of the output in
                // terms of the file names returned by FileSystemView!
                UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
            } catch(Exception weTried) {
            }
            JFrame f = new JFrame(APP_TITLE);
            f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

            FileManager fileManager = new FileManager();
            f.setContentPane(fileManager.getGui());

            try {
                URL urlBig = fileManager.getClass().getResource("fm-icon-32x32.png");
                URL urlSmall = fileManager.getClass().getResource("fm-icon-16x16.png");
                ArrayList<Image> images = new ArrayList<Image>();
                images.add( ImageIO.read(urlBig) );
                images.add( ImageIO.read(urlSmall) );
                f.setIconImages(images);
            } catch(Exception weTried) {}

            f.pack();
            f.setLocationByPlatform(true);
            f.setMinimumSize(f.getSize());
            f.setVisible(true);
```

```
            fileManager.showRootFile();
          }
        });
    }
}

/** A TableModel to hold File[]. */
class FileTableModel extends AbstractTableModel {

    private File[] files;
    private FileSystemView fileSystemView = FileSystemView.getFileSystemView();
    private String[] columns = {
        "Icon",
        "File",
        "Path/name",
        "Size",
        "Last Modified",
        "R",
        "W",
        "E",
        "D",
        "F",
    };

    FileTableModel() {
        this(new File[0]);
    }

    FileTableModel(File[] files) {
        this.files = files;
    }

    public Object getValueAt(int row, int column) {
        File file = files[row];
        switch (column) {
            case 0:
                return fileSystemView.getSystemIcon(file);
            case 1:
                return fileSystemView.getSystemDisplayName(file);
```

```
            case 2:
                return file.getPath();
            case 3:
                return file.length();
            case 4:
                return file.lastModified();
            case 5:
                return file.canRead();
            case 6:
                return file.canWrite();
            case 7:
                return file.canExecute();
            case 8:
                return file.isDirectory();
            case 9:
                return file.isFile();
            default:
                System.err.println("Logic Error");
        }
        return "";
    }

    public int getColumnCount() {
        return columns.length;
    }

    public Class<?> getColumnClass(int column) {
        switch (column) {
            case 0:
                return ImageIcon.class;
            case 3:
                return Long.class;
            case 4:
                return Date.class;
            case 5:
            case 6:
            case 7:
            case 8:
            case 9:
                return Boolean.class;
```

```
        }
        return String.class;
    }

    public String getColumnName(int column) {
        return columns[column];
    }

    public int getRowCount() {
        return files.length;
    }

    public File getFile(int row) {
        return files[row];
    }

    public void setFiles(File[] files) {
        this.files = files;
        fireTableDataChanged();
    }
}

/** A TreeCellRenderer for a File. */
class FileTreeCellRenderer extends DefaultTreeCellRenderer {

    private FileSystemView fileSystemView;

    private JLabel label;

    FileTreeCellRenderer() {
        label = new JLabel();
        label.setOpaque(true);
        fileSystemView = FileSystemView.getFileSystemView();
    }

    @Override
    public Component getTreeCellRendererComponent(
        JTree tree,
        Object value,
        boolean selected,
```

```
        boolean expanded,
        boolean leaf,
        int row,
        boolean hasFocus) {

        DefaultMutableTreeNode node = (DefaultMutableTreeNode)value;
        File file = (File)node.getUserObject();
        label.setIcon(fileSystemView.getSystemIcon(file));
        label.setText(fileSystemView.getSystemDisplayName(file));
        label.setToolTipText(file.getPath());

        if (selected) {
            label.setBackground(backgroundSelectionColor);
            label.setForeground(textSelectionColor);
        } else {
            label.setBackground(backgroundNonSelectionColor);
            label.setForeground(textNonSelectionColor);
        }

        return label;
    }
}
```

WEBSITE

Index.xml

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="shortcut icon" href="assets/ico/favicon.png">

    <title>File Management System</title>

    <!-- Bootstrap core CSS -->
    <link href="assets/css/bootstrap.css" rel="stylesheet">
```

```
    <!-- HTML5 shim and Respond.js IE8 support of HTML5 elements and media queries -->
    <!--[if lt IE 9]>
      <script src="../../assets/js/html5shiv.js"></script>
      <script src="../../assets/js/respond.min.js"></script>
    <![endif]-->

    <!-- Custom styles for this template -->
    <link href="assets/css/carousel.css" rel="stylesheet">
  </head>

  <body>
    <!-- NAVBAR
    ================================================== -->
    <div class="navbar-wrapper">
      <div class="container">
        <div class="navbar navbar-inverse navbar-static-top">
          <div class="container">
            <div class="navbar-header">
              <button type="button" class="navbar-toggle" data-toggle="collapse" data-
target=".navbar-collapse">
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
              </button>
              <a class="navbar-brand" href="#">File Management System</a>
            </div>
            <div class="collapse navbar-collapse navbar-ex1-collapse">
              <ul class="nav navbar-nav">
                <li class="active"><a href="#">Home</a></li>
                <li><a href="#">About</a></li>
              </ul>
              <form class="navbar-form navbar-left" role="search">
                <div class="form-group">
                  <input type="text" class="form-control" placeholder="Search">
                </div>
                <button type="submit" class="btn btn-default">Submit</button>
              </form>
              <ul class="nav navbar-nav navbar-right">
                <li><a href="#">Support</a></li>
                <li class="dropdown">
```

```
                    <a href="#" class="dropdown-toggle" data-toggle="dropdown">Sign In <b
class="caret"></b></a>
                    <ul class="dropdown-menu" role="menu">
                      <li><a href="signin.html">Sign in</a></li>
                      <li><a href="#">Sign Up</a></li>
                    </ul>
                  </li>
                  <li><a></a></li>
                </ul>
              </div>
            </div>
          </div>
        </div>
      </div>

      <!-- Carousel
      ================================================== -->
      <div id="myCarousel" class="carousel slide">
        <!-- Indicators -->
        <ol class="carousel-indicators">
          <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
          <li data-target="#myCarousel" data-slide-to="1"></li>
          <li data-target="#myCarousel" data-slide-to="2"></li>
        </ol>
        <div class="carousel-inner">
          <div class="item active">
            <img src="" data-
src="holder.js/900x500/auto/#777:#7a7a7a/text:File Management System" alt="File
Management System">
            <div class="container">
              <div class="carousel-caption">
                <h1>File Management System </h1>
                <p>File Management System is standalone software for managing files in various
storage devices.</p>
                <p><a class="btn btn-lg btn-primary" href="#" role="button">Sign up today</a></p>
              </div>
            </div>
          </div>
          <div class="item">
            <img src="" data-
src="holder.js/900x500/auto/#666:#6a6a6a/text:Web based server" alt="Web based server">
```

```
        <div class="container">
          <div class="carousel-caption">
            <h1> Web based server</h1>
            <p>FMS provides a web based server which will store information about files stored in user's machine and storage devices.</p>
            <p><a class="btn btn-lg btn-primary" href="#" role="button">Learn more</a></p>
          </div>
        </div>
      </div>
      <div class="item">
        <img src="" data-src="holder.js/900x500/auto/#555:#5a5a5a/text:Applications" alt="Applications">
        <div class="container">
          <div class="carousel-caption">
            <h1>Applications</h1>
            <p>The clients/host apps (windows,Linux, android) will analyze the file systems and upload the information to web server.</p>
            <p><a class="btn btn-lg btn-primary" href="#" role="button">Browse Apps</a></p>
          </div>
        </div>
      </div>
    </div>
    <a class="left carousel-control" href="#myCarousel" data-slide="prev"><span class="glyphicon glyphicon-chevron-left"></span></a>
    <a class="right carousel-control" href="#myCarousel" data-slide="next"><span class="glyphicon glyphicon-chevron-right"></span></a>
  </div><!-- /.carousel -->



  <!-- Marketing messaging and featurettes
  ================================================== -->
  <!-- Wrap the rest of the page in another container to center all the content. -->



  <!-- Bootstrap core JavaScript
  ================================================== -->
  <!-- Placed at the end of the document so the pages load faster -->
  <script src="assets/js/jquery.js"></script>
```

```
    <script src="assets/js/bootstrap.min.js"></script>
    <script src="assets/js/holder.js"></script>
  </body>
</html>
```

Signin.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="shortcut icon" href="assets/ico/favicon.png">

    <title>Signin Template for Bootstrap</title>

    <!-- Bootstrap core CSS -->
    <link href="assets/css/bootstrap.css" rel="stylesheet">

    <!-- Custom styles for this template -->
    <link href="assets/css/signin.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js IE8 support of HTML5 elements and media queries -->
    <!--[if lt IE 9]>
      <script src="../../assets/js/html5shiv.js"></script>
      <script src="../../assets/js/respond.min.js"></script>
    <![endif]-->
  </head>

  <body>

    <div class="container">

      <form class="form-signin">
        <h2 class="form-signin-heading">Please sign in</h2>
        <input type="text" class="form-control" placeholder="Email address" autofocus>
        <input type="password" class="form-control" placeholder="Password">
        <label class="checkbox">
```

```
        <input type="checkbox" value="remember-me"> Remember me
      </label>
      <p><a class="btn btn-lg btn-primary btn-block" type="submit" href="main.html"
role="button">Sign in</a></p>
    </form>
  </div> <!-- /container -->


  <!-- Bootstrap core JavaScript
  ================================================== -->
  <!-- Placed at the end of the document so the pages load faster -->
 </body>
</html>
```

Main.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <!-- Meta, title, CSS, favicons, etc. -->
  <meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="description" content="">
<meta name="author" content="">

<title>

  File Management System

</title>

<!-- Bootstrap core CSS -->
<link href="assets/css/bootstrap.css" rel="stylesheet">

<!-- Documentation extras -->
<link href="assets/css/docs.css" rel="stylesheet">
<link href="assets/css/pygments-manni.css" rel="stylesheet">
<link href="assets/css/font-awesome.css" rel="stylesheet">

<!-- HTML5 shim and Respond.js IE8 support of HTML5 elements and media queries -->
```

```
<!--[if lt IE 9]>
  <script src="../assets/js/html5shiv.js"></script>
  <script src="../assets/js/respond.min.js"></script>
<![endif]-->


<!-- Favicons -->
<link rel="apple-touch-icon-precomposed" sizes="144x144" href="../assets/ico/apple-touch-icon-144-precomposed.png">
<link rel="apple-touch-icon-precomposed" sizes="114x114" href="../assets/ico/apple-touch-icon-114-precomposed.png">
  <link rel="apple-touch-icon-precomposed" sizes="72x72" href="../assets/ico/apple-touch-icon-72-precomposed.png">
                <link rel="apple-touch-icon-precomposed" href="../assets/ico/apple-touch-icon-57-precomposed.png">
                              <link rel="shortcut icon" href="../assets/ico/favicon.png">


<script>
  var _gaq = _gaq || [];
  _gaq.push(['_setAccount', 'UA-146052-10']);
  _gaq.push(['_trackPageview']);
  (function() {
    var ga = document.createElement('script'); ga.async = true;
    ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-analytics.com/ga.js';
    var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);
  })();
</script>
<style>
    .bs-example:after {
  content: "Folder";
    }
</style>
    <!-- Place anything custom after this. -->
  </head>
  <body>
    <div class="navbar-wrapper">
      <div class="container">
        <div class="navbar navbar-inverse navbar-static-top">
          <div class="container">
            <div class="navbar-header">
              <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
```

```
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </button>
          <a class="navbar-brand" href="#">File Management System</a>
        </div>
        <div class="collapse navbar-collapse navbar-ex1-collapse">
          <ul class="nav navbar-nav">
            <li class="active"><a href="#">Home</a></li>
            <li><a href="#">About</a></li>
          </ul>
          <form class="navbar-form navbar-left" role="search">
            <div class="form-group">
              <input type="text" class="form-control" placeholder="Search">
            </div>
            <button type="submit" class="btn btn-default">Submit</button>
          </form>
          <ul class="nav navbar-nav navbar-right">
            <li><a href="#">Support</a></li>
            <li class="dropdown">
              <a href="#" class="dropdown-toggle" data-toggle="dropdown">Sign In <b
class="caret"></b></a>
              <ul class="dropdown-menu" role="menu">
                <li><a href="signin.html">Sign in</a></li>
                <li><a href="#">Sign Up</a></li>
              </ul>
            </li>
            <li><a></a></li>
          </ul>
        </div>
      </div>
    </div>
  </div>

  <div class="container bs-docs-container">
    <div class="row">
      <div class="col-md-3">
        <div class="bs-sidebar hidden-print" role="complementary">
          <ul class="nav bs-sidenav">
```

```
          <li>
            <a href="#Desktop"><span class="badge pull-right">24</span>Dekstop</a>
          </li>
          <li>
             <a href="#Mobile"><span class="badge pull-right">24</span>Mobile</a>
          </li>
          <li>
             <a href="#Pendrive"><span class="badge pull-right">24</span>Pendrive</a>
            </li>
          <li>
            <a href="#SDcard"><span class="badge pull-right">24</span>SD card</a>
          </li>
          <li>
            <a href="#CDDVD"><span class="badge pull-right">24</span>CD/DVD</a>
          </li>
        </ul>
       </div>
      </div>
      <div class="col-md-3">
       <div class="bs-sidebar hidden-print" role="complementary">
        <ul class="nav bs-sidenav">
          <li>
            <a href="#Desktop">Dekstop</a>
            <hr></hr>
               <ul class="nav">
                  <li><a href="#Soumen-PC"><span class="label label-success pull-
right">12</span>Soumen-PC</a></li>
                  <li><a href="#Technicise-PC"><span class="label label-success pull-
right">12</span>Technicise-PC</a></li>
               </ul>
          </li>
          <li>
             <a href="#Mobile">Mobile</a>
             <hr></hr>
              <ul class="nav">
                  <li><a href="#Android-mob"><span class="label label-success pull-
right">12</span>Android-mob</a></li>
                  <li><a href="#Windows-mob"><span class="label label-success pull-
right">12</span>Windows-mob</a></li>
               </ul>
          </li>
```

```
          <li>
              <a href="#Pendrive">Pendrive</a>
              <hr></hr>
              <ul class="nav">
                 <li><a href="#HP"><span class="label label-success pull-
right">12</span>HP</a></li>
                 <li><a href="#Sony"><span class="label label-success pull-
right">12</span>Sony</a></li>
              </ul>
            </li>
          <li>
            <a href="#SDcard">SD card</a>
            <hr></hr>
              <ul class="nav">
                 <li><a href="#Soumen-SD"><span class="label label-success pull-
right">12</span>Soumen-SD</a></li>
                 <li><a href="#Technicise-SD"><span class="label label-success pull-
right">12</span>Technicise-SD</a></li>
              </ul>
          </li>
          <li>
            <a href="#CDDVD">CD/DVD</a>
            <hr></hr>
              <ul class="nav">
                 <li><a href="#MovieCD"><span class="label label-success pull-
right">12</span>Movie CD</a></li>
                 <li><a href="#SongDVD"><span class="label label-success pull-
right">12</span>Song DVD</a></li>
              </ul>
          </li>
        </ul>
      </div>
    </div>
    <div class="col-md-6" role="main">
      <!-- Overview
 ================================================== -->
 <div class="bs-docs-section">
  <div class="page-header">
   <h1 id="Desktop"><i class="icon-desktop"></i>       Desktop</h1>
   <div class="bs-example">
        <legend id="Soumen-PC">Soumen-PC</legend>
   <div class="row">
```

```
<div class="col-sm-3 col-md-3">
  <a href="#" class="thumbnail">
    <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Music</h5>
   </a>
</div>
<div class="col-sm-3 col-md-3">
  <a href="#" class="thumbnail">
    <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">New Folder</h5>
   </a>
</div>
<div class="col-sm-3 col-md-3">
  <a href="#" class="thumbnail">
    <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Study_mat</h5>
   </a>
</div>
<div class="col-sm-3 col-md-3">
  <a href="#" class="thumbnail">
    <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Personal</h5>
   </a>
</div>
<div class="col-sm-3 col-md-3">
  <a href="#" class="thumbnail">
    <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Movies</h5>
   </a>
</div>
              <div class="col-sm-3 col-md-3">
  <a href="#" class="thumbnail">
    <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Songs</h5>
   </a>
</div>
              <div class="col-sm-3 col-md-3">
  <a href="#" class="thumbnail">
    <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Desktop</h5>
```

```
    </a>
  </div>
                <div class="col-sm-3 col-md-3">
   <a href="#" class="thumbnail">
    <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Documents</h5>
    </a>
  </div>
  </div>
</div>

     <div class="bs-example">
      <legend id="Technicise-PC">Technicise-PC</legend>
<div class="row">
  <div class="col-sm-3 col-md-3">
   <a href="#" class="thumbnail">
    <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Music</h5>
    </a>
  </div>
  <div class="col-sm-3 col-md-3">
   <a href="#" class="thumbnail">
    <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">New Folder</h5>
    </a>
  </div>
  <div class="col-sm-3 col-md-3">
   <a href="#" class="thumbnail">
    <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Study_mat</h5>
    </a>
  </div>
  <div class="col-sm-3 col-md-3">
   <a href="#" class="thumbnail">
    <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Personal</h5>
    </a>
  </div>
  <div class="col-sm-3 col-md-3">
   <a href="#" class="thumbnail">
```

```
        <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Movies</h5>
        </a>
      </div>
                <div class="col-sm-3 col-md-3">
       <a href="#" class="thumbnail">
        <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Songs</h5>
        </a>
      </div>
                <div class="col-sm-3 col-md-3">
       <a href="#" class="thumbnail">
        <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Desktop</h5>
        </a>
      </div>
                <div class="col-sm-3 col-md-3">
       <a href="#" class="thumbnail">
        <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Documents</h5>
        </a>
      </div>
      </div>
    </div>

         </div>
  </div>

 <!-- Mobile
 ================================================== -->
 <div class="bs-docs-section">
  <div class="page-header">
    <h1 id="Mobile"><i class="icon-mobile-phone"></i>          Mobile</h1>
  </div>
    <div class="bs-example">
          <legend id="Android-mob"><i class="icon-android"></i>     Android-mob</legend>
    <div class="row">
<div class="col-sm-3 col-md-3">
        <a href="#" class="thumbnail">
         <img src="folder-my-pictures-icon.png" alt="">
```

```
                        <h5 style="text-align:center">Music</h5>
  </a>
</div>
<div class="col-sm-3 col-md-3">
 <a href="#" class="thumbnail">
  <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">New Folder</h5>
  </a>
</div>
<div class="col-sm-3 col-md-3">
 <a href="#" class="thumbnail">
  <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Study_mat</h5>
  </a>
</div>
<div class="col-sm-3 col-md-3">
 <a href="#" class="thumbnail">
  <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Personal</h5>
  </a>
</div>
<div class="col-sm-3 col-md-3">
 <a href="#" class="thumbnail">
  <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Movies</h5>
  </a>
</div>
            <div class="col-sm-3 col-md-3">
 <a href="#" class="thumbnail">
  <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Songs</h5>
  </a>
</div>
            <div class="col-sm-3 col-md-3">
 <a href="#" class="thumbnail">
  <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Desktop</h5>
  </a>
</div>
            <div class="col-sm-3 col-md-3">
```

```
        <a href="#" class="thumbnail">
          <img src="folder-my-pictures-icon.png" alt="">
                              <h5 style="text-align:center">Documents</h5>
          </a>
        </div>

      </div>
    </div>

          <div class="bs-example">
            <legend id="Windows-mob"><i class="icon-windows"></i> Windows-mob</legend>
     <div class="row">
<div class="col-sm-3 col-md-3">
        <a href="#" class="thumbnail">
          <img src="folder-my-pictures-icon.png" alt="">
                              <h5 style="text-align:center">Music</h5>
          </a>
        </div>
        <div class="col-sm-3 col-md-3">
          <a href="#" class="thumbnail">
            <img src="folder-my-pictures-icon.png" alt="">
                              <h5 style="text-align:center">New Folder</h5>
          </a>
        </div>
        <div class="col-sm-3 col-md-3">
          <a href="#" class="thumbnail">
            <img src="folder-my-pictures-icon.png" alt="">
                              <h5 style="text-align:center">Study_mat</h5>
          </a>
        </div>
        <div class="col-sm-3 col-md-3">
          <a href="#" class="thumbnail">
            <img src="folder-my-pictures-icon.png" alt="">
                              <h5 style="text-align:center">Personal</h5>
          </a>
        </div>
        <div class="col-sm-3 col-md-3">
          <a href="#" class="thumbnail">
            <img src="folder-my-pictures-icon.png" alt="">
                              <h5 style="text-align:center">Movies</h5>
```

```
        </a>
      </div>
                  <div class="col-sm-3 col-md-3">
       <a href="#" class="thumbnail">
         <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Songs</h5>
        </a>
      </div>
                  <div class="col-sm-3 col-md-3">
       <a href="#" class="thumbnail">
         <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Desktop</h5>
        </a>
      </div>
                  <div class="col-sm-3 col-md-3">
       <a href="#" class="thumbnail">
         <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Documents</h5>
        </a>
      </div>

    </div>
   </div>

         </div>




  <!-- Modal
  ================================================== -->
  <div class="bs-docs-section">
    <div class="page-header">
      <h1 id="Pendrive"><i class="icon-stackexchange"></i>      Pendrive</h1>
    </div>
    <div class="bs-example">
           <legend id="HP"><i class="icon-bitbucket"></i>       HP</legend>
      <div class="row">
<div class="col-sm-3 col-md-3">
       <a href="#" class="thumbnail">
```

```
    <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Music</h5>
   </a>
</div>
<div class="col-sm-3 col-md-3">
  <a href="#" class="thumbnail">
    <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">New Folder</h5>
   </a>
</div>
<div class="col-sm-3 col-md-3">
  <a href="#" class="thumbnail">
    <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Study_mat</h5>
   </a>
</div>
<div class="col-sm-3 col-md-3">
  <a href="#" class="thumbnail">
    <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Personal</h5>
   </a>
</div>
<div class="col-sm-3 col-md-3">
  <a href="#" class="thumbnail">
    <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Movies</h5>
   </a>
</div>
              <div class="col-sm-3 col-md-3">
  <a href="#" class="thumbnail">
    <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Songs</h5>
   </a>
</div>
              <div class="col-sm-3 col-md-3">
  <a href="#" class="thumbnail">
    <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Desktop</h5>
   </a>
</div>
```

```
                    <div class="col-sm-3 col-md-3">
        <a href="#" class="thumbnail">
          <img src="folder-my-pictures-icon.png" alt="">
                              <h5 style="text-align:center">Documents</h5>
        </a>
      </div>

    </div>
   </div>

          <div class="bs-example">
           <legend id="Sony"><i class="icon-xing"></i> Sony</legend>
    <div class="row">
<div class="col-sm-3 col-md-3">
        <a href="#" class="thumbnail">
          <img src="folder-my-pictures-icon.png" alt="">
                              <h5 style="text-align:center">Music</h5>
        </a>
      </div>
      <div class="col-sm-3 col-md-3">
        <a href="#" class="thumbnail">
          <img src="folder-my-pictures-icon.png" alt="">
                              <h5 style="text-align:center">New Folder</h5>
        </a>
      </div>
      <div class="col-sm-3 col-md-3">
        <a href="#" class="thumbnail">
          <img src="folder-my-pictures-icon.png" alt="">
                              <h5 style="text-align:center">Study_mat</h5>
        </a>
      </div>
      <div class="col-sm-3 col-md-3">
        <a href="#" class="thumbnail">
          <img src="folder-my-pictures-icon.png" alt="">
                              <h5 style="text-align:center">Personal</h5>
        </a>
      </div>
      <div class="col-sm-3 col-md-3">
        <a href="#" class="thumbnail">
          <img src="folder-my-pictures-icon.png" alt="">
```

```
                        <h5 style="text-align:center">Movies</h5>
        </a>
      </div>
                <div class="col-sm-3 col-md-3">
       <a href="#" class="thumbnail">
         <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Songs</h5>
        </a>
      </div>
                <div class="col-sm-3 col-md-3">
       <a href="#" class="thumbnail">
         <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Desktop</h5>
        </a>
      </div>
                <div class="col-sm-3 col-md-3">
       <a href="#" class="thumbnail">
         <img src="folder-my-pictures-icon.png" alt="">
                        <h5 style="text-align:center">Documents</h5>
        </a>
      </div>

    </div>
   </div>
          </div>
   <!-- sample modal content -->


  <!-- Dropdowns
  ================================================== -->
  <div class="bs-docs-section">
    <div class="page-header">
      <h1 id="CDDVD">CD/DVD</h1>
    </div>
<div class="bs-example">
<legend id="MovieCD"><i class="icon-cloud-upload"></i>    Movie-CD</legend>
     <div class="row">
       <ul class="bs-glyphicons">
                <li><i class="icon-camera-retro icon-4x"></i> Message</li>
                <li><span class="icon-camera icon-4x"></span> DICM</li>
```

```
                <li><span class="icon-download icon-4x"></span> Download</li>
                <li><span class="icon-dropbox icon-4x"></span> Music</li>
                <li><span class="icon-cloud-download icon-4x"></span> Document</li>
                <li><span class="icon-dashboard icon-4x"></span> Pics</li>
                <li><span class="icon-anchor icon-4x"></span> Movies </li>
                <li><span class="icon-bell icon-4x"></span> Technicise</li>
                <li><span class="icon-book icon-4x"></span> Personal</li>
                <li><span class="icon-asterisk icon-4x"></span> Exam</li>
                <li><span class="icon-download icon-4x"></span> Download</li>
                <li><span class="icon-dropbox icon-4x"></span> Music</li>
                <li><span class="icon-cloud-download icon-4x"></span> Document</li>
                <li><span class="icon-dashboard icon-4x"></span> Pics</li>
  </ul>
    </div>
   </div>

        <div class="bs-example">
        <legend id="SongDVD"><i class="icon-music"></i> Songs-DVD</legend>
  <div class="row">
   <ul class="bs-glyphicons">
                <li><i class="icon-camera-retro icon-4x"></i> Message</li>
                <li><span class="icon-camera icon-4x"></span> DICM</li>
                <li><span class="icon-download icon-4x"></span> Download</li>
                <li><span class="icon-dropbox icon-4x"></span> Music</li>
                <li><span class="icon-cloud-download icon-4x"></span> Document</li>
                <li><span class="icon-dashboard icon-4x"></span> Pics</li>
                <li><span class="icon-anchor icon-4x"></span> Movies </li>
                <li><span class="icon-bell icon-4x"></span> Technicise</li>
                <li><span class="icon-book icon-4x"></span> Personal</li>
                <li><span class="icon-asterisk icon-4x"></span> Exam</li>
                <li><span class="icon-download icon-4x"></span> Download</li>
                <li><span class="icon-dropbox icon-4x"></span> Music</li>
                <li><span class="icon-cloud-download icon-4x"></span> Document</li>
                <li><span class="icon-dashboard icon-4x"></span> Pics</li>
  </ul>
    </div>
   </div>


</div>
```

```
<!-- Tooltips
================================================== -->
<div class="bs-docs-section">
 <div class="page-header">
  <h1 id="SDcard">SD Card</small></h1>
 </div>
  <div class="bs-example">
        <legend id="Soumen-SD"><i class="icon-microphone"></i> Soumen-SD</legend>
  <div class="row">
   <ul class="bs-glyphicons">
                <li><i class="icon-camera-retro icon-4x"></i> Message</li>
                <li><span class="icon-camera icon-4x"></span> DICM</li>
                <li><span class="icon-download icon-4x"></span> Download</li>
                <li><span class="icon-dropbox icon-4x"></span> Music</li>
                <li><span class="icon-cloud-download icon-4x"></span> Document</li>
                <li><span class="icon-dashboard icon-4x"></span> Pics</li>
                <li><span class="icon-anchor icon-4x"></span> Movies </li>
                <li><span class="icon-bell icon-4x"></span> Technicise</li>
                <li><span class="icon-book icon-4x"></span> Personal</li>
                <li><span class="icon-asterisk icon-4x"></span> Exam</li>
                <li><span class="icon-download icon-4x"></span> Download</li>
                <li><span class="icon-dropbox icon-4x"></span> Music</li>
                <li><span class="icon-cloud-download icon-4x"></span> Document</li>
                <li><span class="icon-dashboard icon-4x"></span> Pics</li>
  </ul>
    </div>
    </div>

        <div class="bs-example">
        <legend id="Technicise-SD"><i class="icon-magic"></i>     Technicise-SD</legend>
  <div class="row">
   <ul class="bs-glyphicons">
                <li><i class="icon-camera-retro icon-4x"></i> Message</li>
                <li><span class="icon-camera icon-4x"></span> DICM</li>
                <li><span class="icon-download icon-4x"></span> Download</li>
                <li><span class="icon-dropbox icon-4x"></span> Music</li>
                <li><span class="icon-cloud-download icon-4x"></span> Document</li>
                <li><span class="icon-dashboard icon-4x"></span> Pics</li>
                <li><span class="icon-anchor icon-4x"></span> Movies </li>
```

```
                    <li><span class="icon-bell icon-4x"></span> Technicise</li>
                    <li><span class="icon-book icon-4x"></span> Personal</li>
                    <li><span class="icon-asterisk icon-4x"></span> Exam</li>
                    <li><span class="icon-download icon-4x"></span> Download</li>
                    <li><span class="icon-dropbox icon-4x"></span> Music</li>
                    <li><span class="icon-cloud-download icon-4x"></span> Document</li>
                    <li><span class="icon-dashboard icon-4x"></span> Pics</li>
    </ul>
       </div>
      </div>


           </div>


   <!-- Footer
   ================================================== -->
   <footer class="bs-footer" role="contentinfo">
     <div class="container">
       <div class="bs-social">
  <ul class="bs-social-buttons">

  </ul>
</div>
   </div>
    </footer>
   <!-- JS and analytics only. -->
   <!-- Bootstrap core JavaScript
================================================== -->
<!-- Placed at the end of the document so the pages load faster -->
<script src="assets/js/jquery.js"></script>
<script src="assets/js/bootstrap.js"></script>
<script src="http://platform.twitter.com/widgets.js"></script>
<script src="assets/js/holder.js"></script>
<script src="assets/js/application.js"></script>
<!-- Analytics
================================================== -->
<script>
  var _gauges = _gauges || [];
  (function() {
```

```
  var t   = document.createElement('script');
  t.async = true;
  t.id    = 'gauges-tracker';
  t.setAttribute('data-site-id', '4f0dc9fef5a1f55508000013');
  t.src = '//secure.gaug.es/track.js';
  var s = document.getElementsByTagName('script')[0];
  s.parentNode.insertBefore(t, s);
 })();
</script>
 </body>
 </html>
```

ANDROID APP

Activity_storage.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin" >

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp"
        android:layout_marginTop="10dp"
        android:background="@color/Appbackground"
        android:paddingLeft="50dp"
        android:text="Selected Storage"
        android:textColor="@color/white"
        android:textSize="25dp"
        android:textStyle="bold" />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp"
        android:layout_marginTop="10dp"
```

```
        android:background="@color/Appbackground"
        android:paddingLeft="50dp"
        android:text="SD Card"
        android:textColor="@color/white"
        android:textSize="25dp"
        android:textStyle="bold" />
     <TableRow android:layout_width="fill_parent"
    android:layout_marginTop="10dp"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    >


        <Button
            android:textColor="@color/white"
           android:background="@android:color/holo_purple"
        android:text="File Type"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"

  android:padding="2dp"
   android:gravity="center"
    android:textStyle="italic"
    android:textSize="15dp"
        />
         <Button
             android:layout_marginLeft="0dp"
             android:textSize="15dp"
               android:textColor="@color/white"
        android:text="Memory Usage"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
android:background="@android:color/holo_purple"
  android:padding="2dp"
   android:gravity="center"
    android:textStyle="italic"
        />
    </TableRow>
     <TableRow android:layout_width="fill_parent"
```

```
    android:layout_marginTop="3dp"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    >


        <Button
            android:textColor="@color/white"
           android:background="@color/Appbackground"
        android:text="PDF"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"

  android:padding="2dp"
   android:gravity="center"
    android:textStyle="italic"
    android:textSize="15dp"
        />
         <Button
             android:layout_marginLeft="0dp"
             android:textSize="15dp"
               android:textColor="@color/white"
        android:text="35.60 MB"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
android:background="@color/Appbackground"
 android:padding="2dp"
  android:gravity="center"
    android:textStyle="italic"
        />

    </TableRow>
     <TableRow android:layout_width="fill_parent"
    android:layout_marginTop="3dp"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    >
```

```
        <Button
            android:textColor="@color/white"
           android:background="@color/Appbackground"
        android:text="MP3"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"

 android:padding="2dp"
  android:gravity="center"
   android:textStyle="italic"
   android:textSize="15dp"
        />
         <Button
             android:layout_marginLeft="0dp"
             android:textSize="15dp"
               android:textColor="@color/white"
        android:text="1.23 GB"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
android:background="@color/Appbackground"
 android:padding="2dp"
  android:gravity="center"
   android:textStyle="italic"
        />

   </TableRow>
    <TableRow android:layout_width="fill_parent"
   android:layout_marginTop="3dp"
   android:layout_height="wrap_content"
   android:gravity="center_vertical"
   >


        <Button
            android:textColor="@color/white"
           android:background="@color/Appbackground"
        android:text="MP4"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
```

```
    android:layout_weight="1"

  android:padding="2dp"
   android:gravity="center"
    android:textStyle="italic"
    android:textSize="15dp"
        />
         <Button
              android:layout_marginLeft="0dp"
              android:textSize="15dp"
                android:textColor="@color/white"
        android:text="928 MB"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
android:background="@color/Appbackground"
 android:padding="2dp"
  android:gravity="center"
    android:textStyle="italic"
        />

    </TableRow>
     <TableRow android:layout_width="fill_parent"
    android:layout_marginTop="3dp"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    >


        <Button
            android:textColor="@color/white"
           android:background="@color/Appbackground"
        android:text="APK"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"

  android:padding="2dp"
   android:gravity="center"
    android:textStyle="italic"
    android:textSize="15dp"
```

```
        />
         <Button
              android:layout_marginLeft="0dp"
              android:textSize="15dp"
                android:textColor="@color/white"
        android:text="3.56 GB"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
android:background="@color/Appbackground"
 android:padding="2dp"
  android:gravity="center"
    android:textStyle="italic"
        />

    </TableRow>
   <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"

         android:layout_marginTop="10dp"
            android:paddingLeft="10dp"
            android:text="Note: Tap on a File type to view details"
            android:textSize="15dp"
            android:textStyle="italic" />
        <TableRow android:layout_width="fill_parent"
    android:layout_marginTop="50dp"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    android:background="@android:color/transparent">


        <Button
        android:text="Back"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
  android:background="#b0b0b0"
 android:padding="10dp"
  android:gravity="center"
    android:textStyle="italic"
```

```
        />
         <Button
        android:text="Settings"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
   android:background="#b0b0b0"
  android:padding="10dp"
   android:gravity="center"
    android:textStyle="italic"
        />
    </TableRow>
</LineraLayout>
```

Activity_devices.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin" >

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp"
        android:layout_marginTop="10dp"
        android:background="@color/Appbackground"
        android:paddingLeft="50dp"
        android:text="Devices"
        android:textColor="@color/white"
        android:textSize="25dp"
        android:textStyle="bold" />
     <TableRow android:layout_width="fill_parent"
```

```
    android:layout_marginTop="10dp"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    >


        <Button
            android:textColor="@color/white"
           android:background="@android:color/holo_purple"
        android:text="Name"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"

  android:padding="2dp"
   android:gravity="center"
    android:textStyle="italic"
    android:textSize="15dp"
        />
        <Button
            android:textColor="@color/white"
           android:background="@android:color/holo_purple"
        android:text="Type"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"

  android:padding="2dp"
   android:gravity="center"
    android:textStyle="italic"
    android:textSize="15dp"
        />
         <Button
             android:layout_marginLeft="0dp"
             android:textSize="15dp"
               android:textColor="@color/white"
        android:text="Files"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
android:background="@android:color/holo_purple"
```

```
 android:padding="2dp"
  android:gravity="center"
   android:textStyle="italic"
       />
   </TableRow>
    <TableRow android:layout_width="fill_parent"
   android:layout_marginTop="3dp"
   android:layout_height="wrap_content"
   android:gravity="center_vertical"
   >


       <Button
            android:textColor="@color/white"
           android:background="@color/Appbackground"
       android:text="Nexus S-004"
      android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:layout_weight="1"

 android:padding="2dp"
  android:gravity="center"
   android:textStyle="italic"
   android:textSize="15dp"
       />
        <Button
            android:layout_marginLeft="0dp"
            android:textSize="15dp"
              android:textColor="@color/white"
       android:text="Mobile"
      android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:layout_weight="1"
android:background="@color/Appbackground"
 android:padding="2dp"
  android:gravity="center"
   android:textStyle="italic"
       />
        <Button
           android:textColor="@color/white"
          android:background="@color/Appbackground"
```

```
        android:text="677"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"

android:padding="2dp"
 android:gravity="center"
  android:textStyle="italic"
  android:textSize="15dp"
      />

  </TableRow>
   <TableRow android:layout_width="fill_parent"
  android:layout_marginTop="3dp"
  android:layout_height="wrap_content"
  android:gravity="center_vertical"
  >


      <Button
          android:textColor="@color/white"
         android:background="@color/Appbackground"
      android:text="Ajeya-PC"
     android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"

android:padding="2dp"
 android:gravity="center"
  android:textStyle="italic"
  android:textSize="15dp"
      />
      <Button
          android:textColor="@color/white"
         android:background="@color/Appbackground"
      android:text="PC"
     android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"

android:padding="2dp"
```

```
  android:gravity="center"
   android:textStyle="italic"
   android:textSize="15dp"
       />
        <Button
            android:layout_marginLeft="0dp"
            android:textSize="15dp"
              android:textColor="@color/white"
       android:text="7312"
      android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:layout_weight="1"
android:background="@color/Appbackground"
 android:padding="2dp"
  android:gravity="center"
   android:textStyle="italic"
       />

   </TableRow>
    <TableRow android:layout_width="fill_parent"
   android:layout_marginTop="3dp"
   android:layout_height="wrap_content"
   android:gravity="center_vertical"
   >



       <Button
           android:textColor="@color/white"
          android:background="@color/Appbackground"
       android:text="Chandra-PC"
      android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:layout_weight="1"

 android:padding="2dp"
  android:gravity="center"
   android:textStyle="italic"
   android:textSize="15dp"
       />
       <Button
           android:textColor="@color/white"
```

```
            android:background="@color/Appbackground"
          android:text="PC"
         android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:layout_weight="1"

   android:padding="2dp"
    android:gravity="center"
     android:textStyle="italic"
     android:textSize="15dp"
          />
           <Button
               android:layout_marginLeft="0dp"
               android:textSize="15dp"
                 android:textColor="@color/white"
          android:text="16576"
         android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:layout_weight="1"
  android:background="@color/Appbackground"
   android:padding="2dp"
    android:gravity="center"
     android:textStyle="italic"
          />

     </TableRow>
      <TableRow android:layout_width="fill_parent"
     android:layout_marginTop="3dp"
     android:layout_height="wrap_content"
     android:gravity="center_vertical"
     >

     </TableRow>
    <TextView
              android:layout_width="fill_parent"
              android:layout_height="wrap_content"

           android:layout_marginTop="10dp"
              android:paddingLeft="10dp"
              android:text="Note: Tap on a Device type to view details"
              android:textSize="15dp"
```

```
            android:textStyle="italic" />
        <TableRow android:layout_width="fill_parent"
    android:layout_marginTop="50dp"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    android:background="@android:color/transparent">


        <Button
        android:text="Back"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
   android:background="#b0b0b0"
  android:padding="10dp"
   android:gravity="center"
    android:textStyle="italic"
        />
         <Button
        android:text="Settings"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
   android:background="#b0b0b0"
  android:padding="10dp"
   android:gravity="center"
    android:textStyle="italic"
        />
    </TableRow>
</LinearLayout>
```

Activity_login.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
```

```
android:paddingTop="@dimen/activity_vertical_margin" >

<TextView
    android:text="Email/Phone Login"
   android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:paddingLeft="10dp"
android:textSize="20dp"
android:layout_marginTop="10dp"
android:layout_marginLeft="5dp"
android:textStyle="bold"
    />


<TableRow android:layout_width="fill_parent"
android:layout_marginTop="10dp"
android:layout_height="wrap_content"
android:gravity="center_vertical"
>
</TableRow>
 <TableRow android:layout_width="fill_parent"
android:layout_marginTop="1dp"
android:layout_height="wrap_content"
android:gravity="center_vertical"
>
</TableRow>

<TextView
    android:text="Username"
   android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:paddingLeft="10dp"
android:textSize="12dp"
android:layout_marginTop="10dp"
android:layout_marginLeft="5dp"
android:textStyle="italic"
    />

<EditText
    android:layout_width="fill_parent"
android:layout_height="wrap_content"
```

```
android:layout_marginLeft="10dp"
android:layout_marginTop="5dp"
android:hint="required"
android:background="@android:color/darker_gray"
    />
<TextView
    android:text="Password"
   android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:paddingLeft="10dp"
android:textSize="12dp"
android:layout_marginTop="10dp"
android:layout_marginLeft="5dp"
android:textStyle="italic"
    />
<EditText
    android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_marginLeft="10dp"
android:layout_marginTop="5dp"
android:hint="Required"
android:background="@android:color/darker_gray"
    />

 <TableRow android:layout_width="fill_parent"
android:layout_marginTop="10dp"
android:layout_height="wrap_content"
android:gravity="center_vertical"
android:background="@android:color/transparent">

    <CheckBox
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
      />

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"


        android:paddingLeft="10dp"
```

```
            android:text="Remember"
            android:textSize="15dp"
            android:textStyle="italic" />

    </TableRow>
    <TableRow android:layout_width="fill_parent"
    android:layout_marginTop="10dp"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    android:background="@android:color/transparent">


        <Button
        android:text="Back"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
   android:background="#b0b0b0"
  android:padding="10dp"
   android:gravity="center"
    android:textStyle="italic"
        />
         <Button
        android:text="Save"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
   android:background="#b0b0b0"
  android:padding="10dp"
   android:gravity="center"
    android:textStyle="italic"
        />
    </TableRow>
</LinearLayout>
```

Activity_Register.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```
android:orientation="vertical"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin" >

<TextView
    android:text="Email/Phone Login"
   android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:paddingLeft="10dp"
android:textSize="20dp"
android:layout_marginTop="10dp"
android:layout_marginLeft="5dp"
android:textStyle="bold"
    />


<TableRow android:layout_width="fill_parent"
android:layout_marginTop="10dp"
android:layout_height="wrap_content"
android:gravity="center_vertical"
>
</TableRow>
 <TableRow android:layout_width="fill_parent"
android:layout_marginTop="1dp"
android:layout_height="wrap_content"
android:gravity="center_vertical"
>
</TableRow>

<TextView
    android:text="Choose Username"
   android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:paddingLeft="10dp"
android:textSize="12dp"
android:layout_marginTop="10dp"
android:layout_marginLeft="5dp"
android:textStyle="italic"
    />
```

```
<EditText
    android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_marginLeft="10dp"
android:layout_marginTop="5dp"
android:hint=""
android:background="@android:color/darker_gray"
    />
<TextView
    android:text="Password"
   android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:paddingLeft="10dp"
android:textSize="12dp"
android:layout_marginTop="10dp"
android:layout_marginLeft="5dp"
android:textStyle="italic"
    />
<EditText
    android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_marginLeft="10dp"
android:layout_marginTop="5dp"
android:hint=""
android:background="@android:color/darker_gray"
    />
<TextView
    android:text="Confirm Password"
   android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:paddingLeft="10dp"
android:textSize="12dp"
android:layout_marginTop="10dp"
android:layout_marginLeft="5dp"
android:textStyle="italic"
    />
<EditText
    android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_marginLeft="10dp"
```

```
android:layout_marginTop="5dp"
android:hint=""
android:background="@android:color/darker_gray"
    />
 <TextView
    android:text="Email Address"
   android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:paddingLeft="10dp"
android:textSize="12dp"
android:layout_marginTop="10dp"
android:layout_marginLeft="5dp"
android:textStyle="italic"
    />
<EditText
    android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_marginLeft="10dp"
android:layout_marginTop="5dp"
android:hint=""
android:background="@android:color/darker_gray"
    />

 <TableRow android:layout_width="fill_parent"
android:layout_marginTop="10dp"
android:layout_height="wrap_content"
android:gravity="center_vertical"
android:background="@android:color/transparent">

</TableRow>
<TableRow android:layout_width="fill_parent"
android:layout_marginTop="10dp"
android:layout_height="wrap_content"
android:gravity="center_vertical"
android:background="@android:color/transparent">


    <Button
    android:text="Back"
   android:layout_width="fill_parent"
android:layout_height="wrap_content"
```

```
    android:layout_weight="1"
   android:background="#b0b0b0"
  android:padding="10dp"
   android:gravity="center"
    android:textStyle="italic"
        />
         <Button
        android:text="Finish"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
   android:background="#b0b0b0"
  android:padding="10dp"
   android:gravity="center"
    android:textStyle="italic"
        />
    </TableRow>
</Linearlayout>
```

Activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@color/Appbackground"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin" >

    <TextView
        android:text="Welcome to Fitness Tracker"
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:paddingLeft="10dp"
    android:textSize="20dp"
    android:textColor="@color/white"
    android:layout_marginTop="30dp"
    android:layout_marginLeft="5dp"
```

```
    android:textStyle="bold"
        />
    <TextView
        android:text="Fitness Tracker should be used after consulting your diet
with the doctor."
       android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:paddingLeft="10dp"
    android:textSize="10dp"
    android:textColor="@color/white"
    android:layout_marginTop="15dp"
    android:layout_marginLeft="5dp"
    android:textStyle="bold"
        />

       <Button
           android:layout_width="fill_parent"
           android:layout_height="wrap_content"
           android:background="@android:color/holo_purple"
                    android:layout_marginTop="20dp"
             android:text="Back"
           android:textStyle="italic"
           android:textColor="@color/white" />

         <Button
             android:id="@+id/action_settings"
             android:background="@android:color/holo_purple"
             android:layout_width="fill_parent"
             android:layout_height="wrap_content"
                    android:layout_marginTop="20dp"
             android:text="Save"
             android:textStyle="italic"
             android:textColor="@color/white"/>

    <TableRow android:layout_width="fill_parent"
    android:layout_marginTop="10dp"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    >
    </TableRow>
     <TableRow android:layout_width="fill_parent"
    android:layout_marginTop="1dp"
```

```
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    >
    </TableRow>

     <TableRow android:layout_width="fill_parent"
    android:layout_marginTop="10dp"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    android:background="@android:color/transparent">

    </TableRow>
    <TableRow android:layout_width="fill_parent"
    android:layout_marginTop="10dp"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    android:background="@android:color/transparent">
    </TableRow>

    </TableRow>
</Linearlayout>
```

Strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">File Management System</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>
<color name="Appbackground">#25bfda</color>
<color name="white">#ffffff</color>
</resources>
```

Dimens.xml

```
<resources>

    <!-- Default screen margins, per the Android Design guidelines. -->
```

```
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>

</resources>
```

AndroidMainfest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.filemanagementsystem"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="17"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.filemanagementsystem.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

COMMENTS AND DESCRIPTION OF CODING SEGMENTS

CODE COMMENTING

- *All comments have been written in the same language, be grammatically correct, and contain appropriate punctuation.*
- *Used // or /// but never /* … */*
- *Did not "flowerbox" comment blocks.*
- *Example:*
- *// **
- *// Comment block*
- *// **
- *Always Used inline-comments to explain assumptions, known issues, and algorithm insights.*
- *Never used inline-comments to explain obvious code. Well written code is self documenting.*
- *Only used comments for bad code to say "fix this code" – otherwise remove, or rewrite the code!*
- *Included comments using Task-List keyword flags to allow comment-filtering.*
- *Example:*
- *//always close the connection*
- *//Note.id = msqlReader.GetString("id");*
- *//define the command reference*
- *Always applied C# comment-blocks (///) to public, protected, and internal declarations.*
- *Only used C# comment-blocks for documenting the API.*
- *Included #region and #endregion where possible for whole sections to have a #region-like thing and collapse them.*
- *Example:*

```
#region User

        public static int DoRegisterNewUser(UserInfo NewUser)
        {
            return DoRegisterNewuserindb(NewUser);
        }

        private static int DoRegisterNewuserindb(UserInfo NewUser)
        {
            int returnVal = 0;
            MySql.Data.MySqlClient.MySqlConnection msqlConnection =
OpenDbConnection();

            try
            {
```

```
                //define the command reference
                MySql.Data.MySqlClient.MySqlCommand msqlCommand = new
MySql.Data.MySqlClient.MySqlCommand();

                //define the connection used by the command object
                msqlCommand.Connection = msqlConnection;

                msqlCommand.CommandText = "INSERT INTO
user(id,userid,passwrd,hints) " + "VALUES(@id,@userid,@passwrd,@hints)";

                msqlCommand.Parameters.AddWithValue("@id", NewUser.id);
                msqlCommand.Parameters.AddWithValue("@userid", NewUser.userId);
                msqlCommand.Parameters.AddWithValue("@passwrd", NewUser.pass);
                msqlCommand.Parameters.AddWithValue("@hints", NewUser.hints);



                msqlCommand.ExecuteNonQuery();

                returnVal = 1;
            }
            catch (Exception er)
            {
                returnVal = 0;
            }
            finally
            {
                //always close the connection
                msqlConnection.Close();
            }
            return returnVal;
        }

        #endregion
```

DESCRIPTION OF CODING

This software will follow Object Oriented Programming Paradigm and use below mentioned areas:

OOP Language: C#, Java, PHP

RDBMS: SQLite , MySQL

Networking: TCP/IP

Applications: Desktop Systems , Web Application , Personal Cloud

We need to implement several clients (Host App) and server as given below:

1. **Windows Host App**: Windows OS has an API called **FileSystemWatcher.** I will use FileSystemWatcher to watch for changes in a specified directory. You can watch for changes in files and subdirectories of the specified directory. You can create a component to watch files on a local computer, a network drive, or a remote computer.
2. **Ubuntu Linux Host App**: The QFileSystemWatcher class provides an interface for monitoring files and directories for modifications. QFileSystemWatcher monitors the file system for changes to files and directories by watching a list of specified paths. Call addPath() to watch a particular file or directory. Multiple paths can be added using the addPaths() function. Existing paths can be removed by using the removePath() and removePaths() functions. FileSystemWatcher examines each path added to it. Files that have been added to the QFileSystemWatcher can be accessed using the files() function, and directories using the directories() function. The fileChanged() signal is emitted when a file has been modified, renamed or removed from disk. Similarly, the directoryChanged() signal is emitted when a directory or its contents is modified or removed. Note that QFileSystemWatcher stops monitoring files once they have been renamed or removed from disk, and directories once they have been removed from disk.
3. **Android Host App**: FileObserver (android.os.FileObserver) Monitors files (using inotify) to fire an event after files are accessed or changed by by any process on the device (including this one). FileObserver is an abstract class; subclasses must implement the event handler onEvent(int, String). Each FileObserver instance monitors a single file or directory. If a directory is monitored, events

will be triggered for all files and subdirectories inside the monitored directory. An event mask is used to specify which changes or actions to report. Event type constants are used to describe the possible changes in the event mask as well as what actually happened in event callbacks.

4. **FMS Server / Controller**: This will be implemented using CodeIgniter framework and php. This will follow MVC architecture and expose REST interface so that data can be updated and synced from multiple and various kinds of operating systems.
5. **Web Viewer**: This will be implemented in HTML 5 and bootstrap which will allow users to view file information.

STANDARDIZATION OF THE CODING

Coding style causes the most inconsistency and controversy between developers. Each developer has a preference, and rarely are two the same. However, consistent layout, format, and organization are key to creating maintainable code. The following sections describe the preferred way to implement C# source code in order to create readable, clear, and consistent code that is easy to understand and maintain.

FORMATTING

- *Never declared more than 1 namespace per file.*
- *Avoided putting multiple classes in a single file.*
- *Always placed curly braces ({ and }) on a new line.*
- *Always used curly braces ({ and }) in conditional statements.*
- *Always used a Tab & Indention size of 4.*
- *Declared each variable independently – not in the same statement.*
- *Placed namespace "using" statements together at the top of file. Group .NET namespaces above custom namespaces.*
- *Grouped internal class implementation by type in the following order:*

Member variables.

Constructors & Finalizers.

Nested Enums, Structs, and Classes.

Properties

Methods

- *Sequence declarations within type groups based upon access modifier and visibility:*

Public

Protected

Internal

Private

- *Segregate interface Implementation by using #region statements.*
- *Append folder-name to namespace for source files within sub-folders.*
- *Recursively indent all code blocks contained within braces.*
- *Use white space (CR/LF, Tabs, etc) liberally to separate and organize code.*
- *Only declare related attribute declarations on a single line, otherwise stack each attribute as a separatedeclaration.*

Example:

```
// Bad!

[Attrbute1, Attrbute2, Attrbute3]

public class MyClass

{…}

// Good!

[Attrbute1, RelatedAttribute2]

[Attrbute3]

[Attrbute4]

public class MyClass

{…}
```

- *Place Assembly scope attribute declarations on a separate line.*
- *Place Type scope attribute declarations on a separate line.*
- *Place Method scope attribute declarations on a separate line.*
- *Place Member scope attribute declarations on a separate line.*
- *Place Parameter attribute declarations inline with the parameter.*
- *If in doubt, always err on the side of clarity and consistency.*

CODE EFFICIENCY

We started working on the project keeping in mind that we must develop it in a way that it not only provides a very easy to use GUI but also provide a fast and flexible service to the users. We know that a particular work can be done in more than one ways. We have tried all the options and then chose the one which provides the fastest and most secure performance. First of all, we have used the latest technologies of Microsoft like visual studio 2010 as IDE and WPF as GUI to keep our application's performance few steps ahead. We

have studies all the rules of software development life cycle and applied them to keep our application flexible. We have given special attention to the storage related codes. We have avoided all the unnecessary database codes and kept them as short as possible without harming our purpose so that insertion, updating, deletion and fetching of data take place flexibly. You can see the result as a user; our application does all the works very smoothly.

ERROR HANDLING

The C# language's exception handling features help us to deal with any unexpected or exceptional situations that occur when a program is running. Exception handling uses the try, catch, and finally keywords to try actions that may not succeed, to handle failures when you decide that it is reasonable to do so, and to clean up resources afterward. Exceptions can be generated by the common language runtime (CLR), by the .NET Framework or any third-party libraries, or by application code. Exceptions are created by using the throw keyword.

In many cases, an exception may be thrown not by a method that your code has called directly, but by another method further down in the call stack. When this happens, the CLR will unwind the stack, looking for a method with a catch block for the specific exception type, and it will execute the first such catch block that if finds. If it finds no appropriate catch block anywhere in the call stack, it will terminate the process and display a message to the user.

EXCEPTIONS OVERVIEW

Exceptions have the following properties:

Exceptions are types that all ultimately derive from System.Exception.

Use a try block around the statements that might throw exceptions.

Once an exception occurs in the try block, the flow of control jumps to the first associated exception handler that is present anywhere in the call stack. In C#, thecatch keyword is used to define an exception handler.

If no exception handler for a given exception is present, the program stops executing with an error message.

Do not catch an exception unless you can handle it and leave the application in a known state. If you catch System.Exception, rethrow it using the throw keyword at the end of the catch block.

If a catch block defines an exception variable, you can use it to obtain more information about the type of exception that occurred.

Exceptions can be explicitly generated by a program by using the throw keyword.

Exception objects contain detailed information about the error, such as the state of the call stack and a text description of the error.

Code in a finally block is executed even if an exception is thrown. Use a finally block to release resources, for example to close any streams or files that were opened in the try block.

Managed exceptions in the .NET Framework are implemented on top of the Win32 structured exception handling mechanism.

VALIDATION CHECKS

We have performed following data validation checks on available data:

ALLOWED CHARACTER CHECKS

Checks that ascertain that only expected characters are present in a field. For example a numeric field may only allow the digits 0-9, the decimal point and perhaps a minus sign or commas. A text field such as a personal name might disallow characters such as < and >, as they could be evidence of a markup-based security attack. An e-mail address might require exactly one @ sign and various other structural details. Regular expressions are effective ways of implementing such checks. (See also data type checks below)

BATCH TOTALS

Checks for missing records. Numerical fields may be added together for all records in a batch. The batch total is entered and the computer checks that the total is correct, e.g., add the 'Total Cost' field of a number of transactions together.

CARDINALITY CHECK

Checks that record has a valid number of related records. For example if Contact record classified as a Customer it must have at least one associated Order (Cardinality > 0). If order does not exist for a "customer" record then it must be either changed to "seed" or the order must be created. This type of rule can be complicated by additional conditions. For example if contact record in Payroll database is marked as "former employee", then this record must not have any associated salary payments after the date on which employee left organization (Cardinality = 0).

CHECK DIGITS

Used for numerical data. An extra digit is added to a number which is calculated from the digits. The computer checks this calculation when data are entered. For example the last digit of an ISBN for a book is a check digit calculated modulus 10.

CONSISTENCY CHECKS

Checks fields to ensure data in these fields corresponds, e.g., If Title = "Mr.", then Gender = "M".

CONTROL TOTALS

This is a total done on one or more numeric fields which appears in every record. This is a meaningful total, e.g., add the total payment for a number of Customers.

CROSS-SYSTEM CONSISTENCY CHECKS

Compares data in different systems to ensure it is consistent, e.g., The address for the customer with the same id is the same in both systems. The data may be represented differently in different systems and may need to be transformed to a common format to be compared, e.g., one system may store customer name in a single Name field as 'Doe, John Q', while another in three different fields: First_Name (John), Last_Name (Doe) and

Middle_Name (Quality); to compare the two, the validation engine would have to transform data from the second system to match the data from the first, for example, using SQL: Last_Name || ', ' || First_Name || substr(Middle_Name, 1, 1) would convert the data from the second system to look like the data from the first 'Doe, John Q'

DATA TYPE CHECKS

Checks the data type of the input and give an error message if the input data does not match with the chosen data type, e.g., In an input box accepting numeric data, if the letter 'O' was typed instead of the number zero, an error message would appear.

FILE EXISTENCE CHECK

Checks that a file with a specified name exists. This check is essential for programs that use file handling.

FORMAT OR PICTURE CHECK

Checks that the data is in a specified format (template), e.g., dates have to be in the format DD/MM/YYYY.

Regular expressions should be considered for this type of validation.

HASH TOTALS

This is just a batch total done on one or more numeric fields which appears in every record. This is a meaningless total, e.g., add the Telephone Numbers together for a number of Customers.

LIMIT CHECK

Unlike range checks, data are checked for one limit only, upper OR lower, e.g., data should not be greater than 2 (<=2).

LOGIC CHECK

Checks that an input does not yield a logical error, e.g., an input value should not be 0 when there will be a number that divides it somewhere in a program.

PRESENCE CHECK

Checks that important data are actually present and have not been missed out, e.g., customers may be required to have their telephone numbers listed.

RANGE CHECK

Checks that the data lie within a specified range of values, e.g., the month of a person's date of birth should lie between 1 and 12.

REFERENTIAL INTEGRITY

In modern Relational database values in two tables can be linked through foreign key and primary key. If values in the primary key field are not constrained by database internal mechanism,[4] then they should be validated. Validation of the foreign key field checks that referencing table must always refer to a valid row in the referenced table.

SPELLING AND GRAMMAR CHECK

Looks for spelling and grammatical errors.

UNIQUENESS CHECK

Checks that each value is unique. This can be applied to several fields (i.e. Address, First Name, Last Name).

TABLE LOOK UP CHECK

A table look up check takes the entered data item and compares it to a valid list of entries that are stored in a database table.

TESTING

TESTING TECHNIQUES AND TESTING STRATEGIES USED

FMS application will be tested using following strategies to ensure that the application succeeds to complete all the functional and non functional requirements:

DATABASE & DATA INTEGRITY TESTING

The databases and the database processes should be tested as a subsystem within the **FMS** Application.These subsystems should be tested with the target-of-test's User Interface as the interface to the database.

Test Objective:	Ensure that data is stored correctly, audits can be performed, access is controlled
Technique:	SQL queries will be executed in the DB to verify the data content and correctness.
Completion Criteria:	All planned tests have been executed. All defects that have been identified have been resolved All resolutions have been implemented.

FUNCTIONAL TESTING:

Function testing focuses on any requirements for test that can be traced directly to use cases or business functions and business rules. The goals of these tests are to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules. This type of testing is based upon black box techniques; that are verifying the application and its internal processes by interacting with the application via the Graphical

User Interface (GUI) and analyzing the output or results. Identified below is an outline of the function testing recommended for **FMS**:

Test Objective:	Ensure proper target-of-test functionality, including business process validation.
Technique:	Execute each use case, use-case flow, or function, using valid and invalid data, to verify the following: The expected results occur when valid data is used. The appropriate error or warning messages are displayed when invalid data is used. Business rules are properly applied. Black Box end to end testing of configured processes. Manual validation of required and optional fields.
Completion Criteria:	All planned tests have been executed. All defects that have been identified have been resolved All resolutions have been implemented.

REGRESSION TESTING:

Regression testing focuses on software functionality that may have been previously working however through subsequent changes may have been inadvertently impacted. The goals of these tests are to verify that the broader impact of changes has been verified. Identified below is an outline of the regression testing recommended for each application(s)/module(s) of **FMS**.

Test Objective:	Ensure that previously passed test cases continue to pass as the new system development is deployed and that surrounding systems that may be impacted by a change are still functioning as expected.

Technique:	Execute previous passed testing suites to ensure the following: The expected results occur when valid data is used. The appropriate error or warning messages are displayed when invalid data is used. Each business rule is properly applied.
Completion Criteria:	• All planned regression tests have been executed. • All identified defects have been resolved.

USER INTERFACE TESTING:

User Interface (UI) testing verifies a user's interaction with the software. The goal of UI testing is to ensure that the User Interface provides the user with the appropriate access and navigation through the functions of the target-of-test. In addition, UI testing ensures that the objects within the UI function as expected and conform to corporate or industry standards. Most of this testing will have been done during functional testing. The areas of focus will be on design, layout and navigation of the screens.

Test Objective:	UI testing will verify the screens and the layouts and navigation
Technique:	Verify the design and layout of the screen. Identify the integration links. Test the functioning of the links – that the proper page is displayed and correct messages, pop-ups are shown when they need to be displayed etc Validation of general navigation
Completion Criteria:	All navigation test cases have been executed. All screens have been verified as per design and layouts All defects that have been identified have been resolved.

PERFORMANCE PROFILING:

Performance profiling is a performance test in which response times, transaction rates, and other time-sensitive requirements are measured and evaluated. The goal of Performance

Profiling is to verify performance requirements have been achieved. Performance profiling is implemented and executed to profile and tune performance behaviours as a function of conditions such as workload or hardware configurations

Test Objective:	The purpose of performance profiling is to ensure the performance of the **FMS** application is up to the desired level.
Technique:	Use a subset of Test Procedures developed for Function and Business Cycle Testing. Modify data files to increase the number of transactions or the scripts to increase the number of iterations each transaction occurs. This will be done by using Load Runner or Quick Test Professional (QTP).
Completion Criteria:	Single Transaction or single user: Successful completion of the test scripts without any failures and within the expected or required time allocation per transaction. Results are recorded and a performance baseline is created for the major logical functions within the scenarios listed above. All performance defects are reviewed and triaged to an acceptable resolution.

.

LOAD TESTING:

Load testing is a performance test which subjects the target-of-test to varying workloads to measure and evaluate the performance behaviours and ability of the target-of-test to continue to function properly under these different workloads. The goal of load testing is to determine and ensure that the system functions properly at the expected maximum workload. Additionally, load testing evaluates the performance characteristics, such as response times, transaction rates, and other time sensitive issues.

Test Objective:	The purpose of load testing is to verify performance behaviour time for designated transactions or business cases under varying workload conditions.

Technique:	Use a subset of Test Procedures developed for Function and Business Cycle Testing. Scripts will be executed to simulate the peak load for 1 hour and report will be generated and analysed. This will be done using Load Runner.
Completion Criteria:	Multiple transactions or multiple users: Successful completion of the test scripts without any failures and within acceptable time allocation. Results are recorded and a performance baseline is created for the major business functions within the scenarios listed above. All load testing defects are reviewed and triaged to an acceptable resolution.

STRESS TESTING:

Stress testing is a type of performance test implemented and executed to find errors due to low resources or competition for resources. Low memory or disk space may reveal defects in the target-of-test that aren't apparent under normal conditions. Other defects might result from competition for shared resources like database locks or network bandwidth. Stress testing can also be used to identify the peak workload the target-of-test can handle, which is often beyond the production workload.

VOLUME TESTING:

Volume Testing subjects the target-of-test to large amounts of data to determine if limits are reached that cause the software to fail. Volume Testing also identifies the continuous maximum load or volume the target-of-test can handle for a given period. For example, if the target-of-test is processing a set of database records to generate a report, a Volume Test would use a large test database and check that the software behaved normally and produced the correct report.

SECURITY & ACCESS CONTROL TESTING:

Security and Access Control Testing focus on following key areas of security:

Application-level security, including access to the Data or Business Functions

Application-level security ensures the authentication and authorization of a user. Authentication ensures that the user is a valid user of the system and authorization ensures that the user has the proper privileges to perform specific tasks on desired resources of the system. Testing will be conducted to validate the rules by taking into considerations the various roles applicable for the system.

FAILOVER & RECOVERY TESTING:

Failover and Recovery Testing ensures that the target-of-test can successfully failover and recover from a variety of hardware, software or network malfunctions with undue loss of data or data integrity.

Failover testing ensures that, for those systems that must be kept running, when a failover condition occurs, the alternate or backup systems properly "take over" for the failed system without loss of data or transactions.

Recovery testing is an antagonistic test process in which the application or system is exposed to extreme conditions, or simulated conditions, to cause a failure, such as device Input/Output (I/O) failures or invalid database pointers and keys. Recovery processes are invoked and the application or system is monitored and inspected to verify proper application, or system, and data recovery has been achieved.

CONFIGURATION TESTING:

Configuration testing verifies the operation of the target-of-test on different software and hardware configurations. In most production environments, the particular hardware specifications for the client workstations, network connections and database servers vary. Client workstations may have different software loaded—for example, applications, drivers, and so on—and at any one time, many different combinations may be active using different resources.

INSTALLATION/DEPLOY & BACK OUT TESTING:

Installation testing has two purposes. The first is to ensure that the software can be installed under different conditions—such as a new installation, an upgrade and a complete or custom installation—under normal and abnormal conditions. Abnormal conditions include insufficient disk space, lack of privilege to create directories, and so on. The second purpose is to verify that, once installed; the software operates correctly and can be backed out successfully. This usually means running a number of the tests that were developed for Function testing before and after the back out.

POST PRODUCTION TESTING:

The purpose of Post production testing is to verify that, once deployed, the software operates correctly. This usually means running a number of the tests that were developed for Function Testing ensuring that no data is changed/modified in production. Typically, the business SME's assist with Post production testing.

UNIT TESTING:

Unit testing will take place within the construction phase of the project. After application module has been built to meet design specifications, each component (screen, view, interface, conversion program, etc.) will be tested individually to help confirm that it functions properly as an individual unit. Basic performance testing will also be completed during unit test to resolve obvious issues with performance prior to the System Testing.

The resource responsible for development will conduct testing of their module using the unit test conditions defined by the developer based on detailed design documents. The final

step of unit test will be a review by the team lead to obtain their signoff on the component test checklist.

SMOKE TESTING:

Test Objective:	Verifies the major functionality at high level in order to determine if further testing is possible.
Technique:	After initial deployment to the test environment validate all critical components of the application prior to proceeding with testing.
Completion Criteria:	Navigation through the application at high level is possible, testing can continue.

DATA MIGRATION TESTING:

This is the process of testing to verify whether or not the data migration (or conversion) has been successfully completed. The testing process will be carried out by running SQL scripts on both the source and destination databases.

The fields which are present in the newdata Model in the Destination DB(s) will be migrated from the existing systemssource DB(s) to Destination DB(s).

Test Objective:	The objective of this test is to verify that data migration is successful from source DB(s) to destination DB(s).
Technique:	The Team is notified before the data migration. Team runs queries on the source DB and fetches the data. Data Migration is done. Mapped data needs to be determined. Team runs the queries on the Destination DB and fetches the data. Cross verification of the data is done to see that data fetched from the old database is same as the data fetched from the new database. Verification of the table structure. Verification of record counts. Verification of the data formatting.

Completion Criteria:	Data fetched from the Source DB(s) and Destination DB(s) matches. The record count in the Source and the Destination databases should be equal. No data are truncated. Data formatting is proper (if required at any instance). All defects that have been identified have been resolved.

TESTING PLAN USED

Testing is a set of activities that can be planned in advance and conducted systematically. During testing, the program to tested is executed what a set of test cases, and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to.

In a software development project, errors can be injected at any stage during the development. Some requirement errors and design errors are likely to remain undetected. Ultimately, these errors will be reflected in the code. Hence, testing Performs a very critical role for quality assurance and for ensuring the reliability of software.

CAUSE OF TESTING:

The first test of the system is to see whether it produces the correct output. Following this a variety of other tests are conducted:

Response time: this test is conducted to measure the processing of the software.

Volume testing: In this test, we create as many data as would normally be used to a variety that the hardware and the software will function correctly.

Stress Testing: The purpose of this is to prove that the candidate system should not malfunction under peak load.

Recovery & Security: A forced system is induced to test a backup recovery procedure.

System testing is verification of the workability of a system. For this purpose the system is used experimentally to ensure that it will run according to its specification and in the way users expect. These are two stages to this:

The testing of the individual programs by their programmers called Unit testing and

The testing of the overall System testing.

UNIT TESTING INCLUDES THE FOLLOWING:

- Test for number of input parameters equal to number of arguments.
- Test the parameter and argument attributes match.
- Feasibility and validity checks on input data.

- Checks for interpretation of symbols correctly.
- Checks for accurate branching and looping.
- Checks on logical file addressing and searching.
- Checks on the capacity of storage areas and buffers.
- Checks on updating procedure.
- Checks on contents and layout of printed and displayed output.
- Checks for batch control totals.
- Checks on interfacing with other programs, software, database and operating system.
- Checks on documentation.

SYSTEM TESTING INCLUDES THE FOLLOWING:

- Interfacing of run within the system.
- Compilation and continuity of control totals.
- Capacity of logical file storage areas and the handling of overflow.
- Error correction procedures including user involvement.
- User request for amendments and output.
- Timing of runs and routines for the data volumes to be actually handled.
- Output preparation and distribution.
- Audit requirements.
- Logical physical file housekeeping and control.
- The usual procedures in testing are to create data for the initial tests and to use live data for later testing.

THE FOLLOWING POINTS SHOULD BE REMEMBERED PRIMARILY:

- Both the artificial and live data should be representative of reality;
- Live data can often be checked against the previous system's result;
- If the previous and new system differ, the two sets of result should be reconciled if at all possible;
- Logical files are usually needed to fully test the programs and routines;
- Data generating techniques are useful for simulating large volume of input data file records;
- In the final trial run of the complete routine, asset of input data is passed through to the resultant output and/or file updating stage(s);
- Test data should include known incorrect data with a view to test the validation and control procedure.

ANY ENGINEERED PRODUCT (AND MOST OTHER THINGS) CAN BE TESTED IN ONE OF TWO WAYS:

- Knowing the specified function that a product has been designed to perform, test can be conducted that demonstrate each function is fully operational, at the same time searching for errors in each function;
- Knowing the internal workings of a product, tests can be conducted to ensure that all internal operation performs according to specification and all internal components have been adequately exercised. The first test approach is called Black-box and the second, White-box testing.

WHITE-BOX TESTING:

White-box testing sometimes called glass-box testing, is a test case design to derive test cases. Using white-box testing method, the software engineer can test:

- Guarantee that all independent paths within a module have been exercised at least one;
- Exercise all logical decisions on their true and false sides;
- Execute all loops at there boundaries and within there operational bounds; and
- Exercise internal data structures to assure there validity.

A reasonable question might be posed at this juncture: "why spend time and energy worrying about (and testing) logical minutes when we might better spend effort ensuring that program requirements have been met?" stated another way, why don't we spend all of our energies on black-box testing? The answer lies in the nature of software defects.

- Logical errors and incorrect assumptions are inversely proportional to the probability that a program path will be executed. Errors tend to creep into our work when we design and implement function, conditions and control that are out of the mainstream. Every processing tends to be well understood (and well scrutinized), while "special case" processing tends to fall into the cracks.
- We often believe that a logical path is not likely to be executed when, in fact, it may be executed in regular basis. The logical flow of a program is sometimes counterintuitive, meaning that our unconscious assumption about flow of control and data may lead us to make design errors that are uncovered only when path testing commences.
- Typographical errors are random. When a program is translated into programming language source code, it is likely that some typing error will occur. Many will be uncovered by syntax checking mechanism, but others will go undetected until testing begins. It is likely that a type will exist on an obscure logical path as on a mainstream path.

Each of these reasons provides an argument for conducting white-box tests. Black-boxtesting, no matter how through, may miss the kinds of errors noted above. As Beizer has stated: "Bugs lurk in corners and congregate at boundaries". White-box testing is far more likely to uncover them.

BASIS PATH TESTING:

Basis path testing is a **White-box testing** technique first proposed by Tom MeCabe. The basis path methods enables the test case designer to derive a logical complexity measure of a procedural design and use this measure as a guide for defining a basic set that are guaranteed to execute every statement in the program at least one time during testing.

BLACK-BOX TESTING:

It focuses on the functional requirements of the software. That is Black-box testing enables software engineer to derive sets of input conditions that will fully exercise all functional requirements of a program. Black-box testing is not a alternative of white box testing. Rather, it is a complementary approach that is likely to uncover a different class of errors than White-box methods.

Black-box testing attempts to find errors in the following categories.

- Incorrect or missing function.
- Interface errors
- Errors in data structures or external data base access
- Performance errors
- Initializations or termination errors

Unlike **White-box** testing, which is performed early in the testing process, **black-box** testing tends to be applied during later stage of testing. Because **black-box** testing purposely disregards control structure, attention is focused on the information domain. Tests are based on source data from a

Previous period so that the result from the new system can be compared with that of the old one.

With those of the previous ones to answer the following questions:

- How is functional validity testing?
- What classes of input will make good test cases?
- Is the system particularly sensitive to certain input values?
- How are the boundaries of a data class isolated?
- What data rates and data volume can the system tolerate?
- What effect will specific combinations of the data have on system operation?
- By applying **Black-box** techniques, we derive a set of test cases that satisfy the following criteria:
- Test cases that reduce, by errors and a designed to achieve reasonableness testing.
- Test cases that tell us something about the presence or absence of classes of errors, than errors associated only with the specific test at hand.

TEST REPORTS FOR UNIT TEST CASES AND SYSTEM TEST CASES

TEST REPORTS FOR UNIT TEST CASES

Test Case Id	Comment	Status
FMS-001	NA	PASS
FMS-002	NA	PASS
FMS-003	NA	PASS
FMS-004	NA	PASS
FMS-005	NA	PASS
FMS-006	NA	PASS
FMS-007	NA	PASS

FMS-008	NA	PASS
FMS-009	NA	PASS
FMS-010	NA	PASS
FMS-011	NA	PASS
FMS-012	NA	PASS
FMS-013	NA	PASS
FMS-014	NA	PASS
FMS-015	NA	PASS
FMS-016	NA	PASS
FMS-017	NA	PASS
FMS-018	NA	PASS
FMS-019	NA	PASS
FMS-020	NA	PASS
FMS-021	NA	PASS
FMS-022	NA	PASS
FMS-023	NA	PASS
FMS-024	NA	PASS
FMS-025	NA	PASS
FMS-026	NA	PASS
FMS-027	NA	PASS
FMS-028	NA	PASS
FMS-029	NA	PASS
FMS-030	NA	PASS

FMS-031	NA	PASS
FMS-032	NA	PASS
FMS-033	NA	PASS
FMS-034	NA	PASS
FMS-035	NA	PASS
FMS-036	NA	PASS
FMS-037	NA	PASS
FMS-038	NA	PASS
FMS-039	NA	PASS
FMS-040	NA	PASS
FMS-041	NA	PASS
FMS-042	NA	PASS
FMS-043	NA	PASS
FMS-044	NA	PASS
FMS-045	NA	PASS
FMS-046	NA	PASS
FMS-047	NA	PASS
FMS-048	NA	PASS
FMS-049	NA	PASS
FMS-050	NA	PASS
FMS-051	NA	PASS
FMS-052	NA	PASS
FMS-053	NA	PASS

FMS-054	NA	PASS
FMS-055	NA	PASS
FMS-056	NA	PASS
FMS-057	NA	PASS
FMS-058	NA	PASS
FMS-059	NA	PASS
FMS-060	NA	PASS
FMS-061	NA	PASS
FMS-062	NA	PASS
FMS-063	NA	PASS
FMS-064	NA	PASS
FMS-065	NA	PASS
FMS-066	NA	PASS

TEST REPORTS FOR SYSTEM TEST CASES

Test Case Id	Comment	Status
FMS-067	NA	PASS
FMS-068	NA	PASS
FMS-069	NA	PASS
FMS-070	NA	PASS
FMS-071	NA	PASS
FMS-072	NA	PASS

FMS-073	NA	PASS
FMS-074	NA	PASS
FMS-075	NA	PASS
FMS-076	NA	PASS
FMS-077	NA	PASS
FMS-078	NA	PASS
FMS-079	NA	PASS
FMS-080	NA	PASS
FMS-081	NA	PASS
FMS-082	NA	PASS
FMS-083	NA	PASS
FMS-084	NA	PASS
FMS-085	NA	PASS
FMS-086	NA	PASS
FMS-087	NA	PASS
FMS-089	NA	PASS
FMS-090	NA	PASS
FMS-091	NA	PASS
FMS-092	NA	PASS
FMS-093	NA	PASS

DEBUGGING AND CODE IMPROVEMENT:

The steps in the bellow section demonstrate how to create a console application that uses the Debug class to provide information about the program execution.

When the program is run, we can use methods of the Debug class to produce messages that help we to monitor the program execution sequence, to detect malfunctions, or to provide performance measurement information. By default, the messages that the Debug class produces appear in the Output window of the Visual Studio Integrated Development Environment (IDE).

The sample code uses the WriteLine method to produce a message that is followed by a line terminator. When we use this method to produce a message, each message appears on a separate line in the Output window.

When we use the Assert method of the Debug class, the Output window displays a message only if a specified condition evaluates to false. The message also appears in a modal dialog box to the user. The dialog box includes the message, the project name, and the Debug.Assert statement number. The dialog box also includes the following three command buttons:

- **Abort:** The application stops running.
- **Retry:** The application enters debug mode.
- **Ignore:** The application proceeds.

The user must click one of these buttons before the application can continue.

We can also direct output from the **Debug** class to destinations other than the Output window. The **Debug** class has a collection named **Listeners** that includes **Listener** objects.

Each **Listener** object monitors **Debug** output and directs the output to a specified target.

Each **Listener** in the **Listener** collection receives any output that the **Debug** class generates. Use the **TextWriterTraceListener**class to define **Listener** objects. We can specify the target for a **TextWriterTraceListener** class through its constructor.

Some possible output targets include the following:

- The Console window by using the **System.Console.Out** property.
- A text (.txt) file by using the **System.IO.File.CreateText("FileName.txt")** statement.

After we create a **TextWriterTraceListener** object, we must add the object to the **Debug.Listeners** collection to receive Debug output.

CREATE A SAMPLE WITH THE DEBUG CLASS

1. Start Visual Studio or Visual C# Express Edition.
2. Create a new Visual C# Console Application project named **conInfo**. Class1 is created in Visual Studio .NET. Program.cs is created in Visual Studio 2005.

3. Add the following namespace at top in Class1 or Program.cs.
   using System.Diagnostics;
4. To initialize variables to contain information about a product, add the following declaration statements to **Main** method:
   string sProdName = "Widget";
   int iUnitQty = 100;
   double dUnitCost = 1.03;
5. Specify the message that the class produces as the first input parameter of the **WriteLine** method. Press the CTRL+ALT+O key combination to make sure that the Output window is visible.
   Debug.WriteLine("Debug Information-Product Starting ");
6. For readability, use the **Indent** method to indent subsequent messages in the Output window:
   Debug.Indent();
7. To display the content of selected variables, use the **WriteLine** method as follows:
   Debug.WriteLine("The product name is " + sProdName);
   Debug.WriteLine("The available units on hand are" + iUnitQty.ToString());
   Debug.WriteLine("The per unit cost is " + dUnitCost.ToString());
8. We can also use the **WriteLine** method to display the namespace and the class name for an existent object. For example, the following code displays the **System.Xml.XmlDocument** namespace in the Output window:
   System.Xml.XmlDocument oxml = new System.Xml.XmlDocument();
   Debug.WriteLine(oxml);
9. To organize the output, we can include a category as an optional, second input parameter of the **WriteLine** method. If we specify a category, the format of the Output window message is "category: message." For example, the first line of the following code displays "Field: The product name is Widget" in the Output window:
   Debug.WriteLine("The product name is " + sProdName,"Field");
   Debug.WriteLine("The units on hand are" + iUnitQty,"Field");
   Debug.WriteLine("The per unit cost is" + dUnitCost.ToString(),"Field");
   Debug.WriteLine("Total Cost is " + (iUnitQty * dUnitCost),"Calc");
10. The Output window can display messages only if a designated condition evaluates to true by using the **WriteLineIf**method of the **Debug** class. The condition to be evaluated is the first input parameter of the **WriteLineIf** method. The second parameter of **WriteLineIf** is the message that appears only if the condition in the first parameter evaluates to true.
    Debug.WriteLineIf(iUnitQty > 50, "This message WILL appear");
    Debug.WriteLineIf(iUnitQty < 50, "This message will NOT appear");
11. Use the **Assert** method of the **Debug** class so that the Output window displays the message only if a specified condition evaluates to false:
    Debug.Assert(dUnitCost > 1, "Message will NOT appear");
    Debug.Assert(dUnitCost < 1, "Message will appear since dUnitcost < 1 is false");
12. Create the **TextWriterTraceListener** objects for the Console window (tr1) and for a text file named Output.txt (tr2), and then add each object to the **Debug Listeners** collection:
    TextWriterTraceListener tr1 = new TextWriterTraceListener(System.Console.Out);
    Debug.Listeners.Add(tr1);

TextWriterTraceListener tr2 = new
TextWriterTraceListener(System.IO.File.CreateText("Output.txt"));
Debug.Listeners.Add(tr2);

13. For readability, use the **Unindent** method to remove the indentation for subsequent messages that the **Debug** class generates. When we use the **Indent** and the **Unindent** methods together, the reader can distinguish the output as group.
14. Debug.Unindent();
15. Debug.WriteLine("Debug Information-Product Ending");
16. To make sure that each **Listener** object receives all its output, call the **Flush** method for the **Debug** class buffers:
Debug.Flush();

USING THE TRACE CLASS

We can also use the **Trace** class to produce messages that monitor the execution of an application. The **Trace** and **Debug**classes share most of the same methods to produce output, including the following:

- **WriteLine**
- **WriteLineIf**
- **Indent**
- **Unindent**
- **Assert**
- **Flush**

We can use the **Trace** and the **Debug** classes separately or together in the same application. In a Debug Solution Configuration project, both **Trace** and **Debug** output are active. The project generates output from both of these classes to all**Listener** objects. However, a Release Solution Configuration project only generates output from a **Trace** class. The Release Solution Configuration project ignores any **Debug** class method invocations.

```
Trace.WriteLine("Trace Information-Product Starting ");

Trace.Indent();


Trace.WriteLine("The product name is "+sProdName);

Trace.WriteLine("The product name is"+sProdName,"Field" );

Trace.WriteLineIf(iUnitQty > 50, "This message WILL appear");

Trace.Assert(dUnitCost > 1, "Message will NOT appear");


Trace.Unindent();

Trace.WriteLine("Trace Information-Product Ending");
```

```
Trace.Flush();


Console.ReadLine();
```

VERIFY THAT IT WORKS

1. Make sure that **Debug** is the current solution configuration.
2. If the **Solution Explorer** window is not visible, press the CTRL+ALT+L key combination to display this window.
3. Right-click **conInfo**, and then click **Properties**.
4. In the left pane of the conInfo property page, under the **Configuration** folder, make sure that the arrow points to**Debugging**.

   Note In Visual C# 2005 and in Visual C# 2005 Express Edition, click **Debug** in the **conInfo** page.
5. Above the **Configuration** folder, in the **Configuration** drop-down list box, click **Active (Debug)** or **Debug**, and then click **OK**. In Visual C# 2005 and in Visual C# 2005 Express Edition, click **Active (Debug)** or **Debug** in the **Configuration**drop-down list box in the **Debug** page, and then click **Save** on the **File** menu.
6. Press CTRL+ALT+O to display the Output window.
7. Press the F5 key to run the code. When the **Assertion Failed** dialog box appears, click **Ignore**.
8. In the Console window, press ENTER. The program should finish, and the Output window should display the output that resembles the following

```
  Debug Information-Product Starting
  The product name is Widget
  The available units on hand are100
  The per unit cost is 1.03
  System.Xml.XmlDocument
  Field: The product name is Widget
  Field: The units on hand are100
  Field: The per unit cost is1.03
  Calc: Total Cost is  103
  This message WILL appear
  ---- DEBUG ASSERTION FAILED ----
---- Assert Short Message ----
Message will appear since dUnitcost  < 1 is false
---- Assert Long Message ----
```

```
   at Class1.Main(String[] args)  <%Path%>\class1.cs(34)

   The product name is Widget
   The available units on hand are100
   The per unit cost is 1.03
Debug Information-Product Ending
Trace Information-Product Starting
   The product name is Widget
   Field: The product name isWidget
   This message WILL appear
Trace Information-Product Ending
```

The Console window and the Output.txt file should display the following output:

```
The product name is Widget
   The available units on hand are 100
   The per unit cost is 1.03
Debug Information-Product Ending
Trace Information-Product Starting
   The product name is Widget
   Field: The product name is Widget
   This message WILL appear
Trace Information-Product Ending
```

Note The Output.txt file is located in the same directory as the conInfo executable (conInfo.exe). Typically, this is the \bin folder where the project source is stored. By default, this is C:\Documents and Settings*User login*\My Documents\Visual Studio Projects\conInfo\bin. In Visual C# 2005 and in Visual C# 2005 Express Edition, the Output.txt file is located in the following folder:

C:\Documents and Settings*User login*\My Documents\Visual Studio 2005\Projects\conInfo\conInfo\bin\Debug

COMPLETE CODE LISTING

```
using System;
using System.Diagnostics;
```

```
class Class1
  {
    [STAThread]
static void Main(string[] args)
    {
string sProdName = "Widget";
int iUnitQty = 100;
double  dUnitCost = 1.03;
Debug.WriteLine("Debug Information-Product Starting ");
Debug.Indent();
Debug.WriteLine("The product name is "+sProdName);
Debug.WriteLine("The available units on hand are"+iUnitQty.ToString());
Debug.WriteLine("The per unit cost is "+ dUnitCost.ToString());

       System.Xml.XmlDocument oxml = new System.Xml.XmlDocument();
Debug.WriteLine(oxml);

Debug.WriteLine("The product name is "+sProdName,"Field");
Debug.WriteLine("The units on hand are"+iUnitQty,"Field");
Debug.WriteLine("The per unit cost is"+dUnitCost.ToString(),"Field");
Debug.WriteLine("Total Cost is  "+(iUnitQty * dUnitCost),"Calc");

Debug.WriteLineIf(iUnitQty > 50, "This message WILL appear");
Debug.WriteLineIf(iUnitQty < 50, "This message will NOT appear");

Debug.Assert(dUnitCost > 1, "Message will NOT appear");
Debug.Assert(dUnitCost < 1, "Message will appear since dUnitcost  < 1 is false");

       TextWriterTraceListener tr1 = new TextWriterTraceListener(System.Console.Out);
Debug.Listeners.Add(tr1);

       TextWriterTraceListener tr2 = new
TextWriterTraceListener(System.IO.File.CreateText("Output.txt"));
Debug.Listeners.Add(tr2);



Debug.WriteLine("The product name is "+sProdName);
```

```
Debug.WriteLine("The available units on hand are"+iUnitQty);
Debug.WriteLine("The per unit cost is "+dUnitCost);
Debug.Unindent();
Debug.WriteLine("Debug Information-Product Ending");
Debug.Flush();

Trace.WriteLine("Trace Information-Product Starting ");
Trace.Indent();

Trace.WriteLine("The product name is "+sProdName);
Trace.WriteLine("The product name is"+sProdName,"Field" );
Trace.WriteLineIf(iUnitQty > 50, "This message WILL appear");
Trace.Assert(dUnitCost > 1, "Message will NOT appear");



Trace.Unindent();
Trace.WriteLine("Trace Information-Product Ending");

Trace.Flush();

Console.ReadLine();
    }
  }
```

TROUBLESHOOT

- If the solution configuration type is **Release**, the **Debug** class output is ignored.
- After we create a **TextWriterTraceListener** class for a particular target, **TextWriterTraceListener** receives output from the **Trace** and the **Debug** classes. This occurs regardless of whether we use the **Add** method of the **Trace** or the**Debug** class to add **TextWriterTraceListener** to the **Listeners** class.
- If we add a **Listeners** object for the same target in the **Trace** and the **Debug** classes, each line of output is duplicated, regardless of whether **Debug** or **Trace** generates the output.

```
        TextWriterTraceListener myWriter = new
TextWriterTraceListener(System.Console.Out);
Debug.Listeners.Add(myWriter);
```

```
        TextWriterTraceListener myCreator = new
TextWriterTraceListener(System.Console.Out);
Trace.Listeners.Add(myCreator);
```

SYSTEM SECURITY MEASURES:

DATABASE/DATA SECURITY:

- For mobile app, password protection will used for the entire app.
- For mobile app, password protection will be used for single items, like a video or an image.
- For desktop pp, password protection will be incorporated so that the phone back up stays protected.

CREATION OF USER PROFILES AND ACCESS RIGHTS

The software requires a predefined username and password to login.

It allows a guest login as well which lets a guest user this application withvery limited access to the user data.

COST ESTIMATION OF THE PROJECT ALONG WITH COST ESTIMATION MODEL

Software development is a highly labor intensive activity. A project of large dimension can easily turn into chaos if proper management controls are not imposed. Therefore the cost/expenditure and the profit gained after implementing the project has to be taken into account. That is we have to consider the cost benefit analysis.

This cost/benefit may be tangible or intangible, direct or indirect, fixed or variable. To build up a large software all the elements required, are estimated to get the development cost of the considering project. When we consider all this requirements we can develop a cost estimation model to find proposed cost of the developing project. And from this model we can track down the expenditure during the course of development.

Now after implementing the project we have to consider gain from it in terms of benefits, that is how much person month can be saved from this project. Therefore we have to consider the total expenditure and the benefit gain from the project once it has been implemented. Here we express the benefits in the terms of person month that is monthly salary of the person concerned for the system, which has to be replaced. Therefore this cost/benefit analysis report gives us a total picture of how a company gets benefit from candidate system once that has replace a older one.

The project developing components like hardware, personnel, facility and supply cost are also taken into consideration during the cost estimation. Then we identify the cost and benefit of a given system and categories them or analysis. And from that estimated cost we track the expenditure and then calculate the benefits.

In developing the cost estimation of a project we need to consider several cost elements. Among them is hardware, personal, facility, operating and supply cost are noteworthy.

The model for estimating cost is mainly based on the total lines of cop1 delivered. As this is not such code based rather than a p1ign based project so we estimate the cost on the consideration of how much time it can take in p1igning the user interfaces and how many interfaces are required. The cost is then calculated from the total p1ign hours and as it is a single handed project, so this is the time taken by a single person.

The cost of man-power involved in this project is not considered in this estimation. We should consider the cost when we shall go for any live project. This cost is depending on several factors like skill set, location of country etc. e.g. man-hour cost is around Rs.250.00 to Rs.300.00 in India whereas for USA it varies from US$60.00 to US$200.00. Most of the cases, Man (person) power cost is considerable higher than all other costs. Software cost and effort estimation will never be an exact science. Too many variables human, technical, environmental can affect the ultimate cost of the software and effort applied to develop it. To achieve reliable cost and effort estimates, a number of option arise, 1) Base estimates on similar projects that have been already completed; 2) Using relatively simple "decomposition techniques" to generate project cost and effort estimates; 3) Using one or more empirical models for software cost and effort estimation.

We used the basic COCOMO model, which gives an approximate estimate of our **FMS** project parameters. The basic COCOMO estimation model is given by the following expressions:

Effort = a1 * (KLOC)a2 PM

Tdev = b1 * (Effort)b2months

Where

KLOC is the estimated size of the software product expressed in Kilo Lines of Code a1, a2, b1, b2 are constants for each category of software products.

Tdev is the estimated time to develop the software, expressed in months.

Effort is the total effort required to develop the software product, expressed in person-month (PM).

Our project is semidetached type, because the development team consists of a mixture of experienced and inexperienced staff like my guide and me. Team members may have limited experience on related system but may be unfamiliar with aspects of the system being developed.

ESTIMATION OF DEVELOPMENT EFFORT

For our Semi-detached class software product **FMS**, the formula for estimating the effort based on the code size is shown below:

Semi-detached **FMS**: Tdev = 3.0*(KLOC)1.12 PM

ESTIMATION OF DEVELOPMENT TIME

For our Semi-detached class software product **FMS**, the formula for estimating the development time based on the effort is given below:

Semi-detached **FMS**: Tdev = 2.5*(Effort)0.35 months

Assume that the size of a Semi-detached **FMS**product has been estimated to be 3,200 lines of source code. Assume that the average salary of software engineer(me) is Rs. 20,000 per month.

Assume that the size of our

The basic COCOMO estimation formula for **FMS** semidetached software:

Our Effort =3.0*(3.2)1.12PM

= 11 PM

Normal Development time = 2.5*(11)0.35months

=6 months

Cost required to develop the product = Rs. 6 * 20000

= Rs. 120,000

REPORTS

- List of Devices updates could be generated.
- List of files could be generated.
- A list of duplicate files could be generated.
- List of most used files could be generated.
- List of File size and memory status could be generated.

FUTURE SCOPE AND FURTHER ENHANCEMENT OF THE PROJECT

- The application provides only Google drive cloud storage back up. We could develop skydrive and dropbox cloud storage back up as well.
- It could be really useful if the app could provide a video and image file locker as well

BIBLIOGRAPHY

WEBSITE

- http://msdn.microsoft.com/en-us/library/system.io.filesystemwatcher.aspx
- http://qt-project.org/doc/qt-5.0/qtcore/qfilesystemwatcher.html#details
- http://developer.android.com/reference/android/os/FileObserver.html
- http://en.wikipedia.org

- http://msdn.microsoft.com/en-us/
- http://www.microsoft.com/en-us/default.aspx
- http://www.codeplex.com/
- http://stackoverflow.com/
- http://www.codeguru.com/
- http://www.w3schools.com
- www.mysql.org

BOOKS

- Fundamentals of software engineering by Rajib Mall
- Pro C# 2010 and the .NET 4.0 Platform by Andrew Troselen
- C# Programming by Rob Miles
- IT Professionals and other Windows Phone users.
- PHP for Dummies
- Java for Dummies

APPENDICES

IDE USED:

VISUAL STUDIO 2010

Microsoft Visual Studio is a powerful IDE that ensures quality code throughout the entire application lifecycle, from design to deployment. Whether we are developing applications for SharePoint, the web, Windows, Windows Phone, and beyond, Visual Studio is the ultimate all-in-one solution. Visual Studio includes a code editor supporting IntelliSense as well as code refactoring. The

integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source-control systems (like Subversion and Visual SourceSafe) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).

STANDOUT FEATURES

- User interface built on Windows Presentation Foundation (WPF)
- Improved Start page
- Improved code editor
- Improved IntelliSense
- Call Hierarchy Viewer

WHAT PROBLEMS DOES IT SOLVE?

The newly designed user experience is refreshing for an application showing its age. The user interface is built on WPF and no longer relies on the limited MDI interface in previous versions; this allows for better multi-monitor support with fly-out windows. The first thing you might notice when opening Visual Studio 2010 is the new Start page. As an xaml file, this page is completely customizable and includes the ability to remove and pin project files in the Recent Projects section.

The code editor has a number of enhancements. You can scale the font by holding down [Ctrl] while scrolling the mouse wheel. In previous versions of Visual Studio,

users had to set the font size through a dialog and exit to see if the changes were correct.

In Visual Studio 2010, Box Selection is enhanced to allow for zero-length boxes and improved pasting.

The feature that will see the most use (by accident if not design) is Highlight References. By selecting any symbol, such as a variable or a property, all references to the symbol are highlighted. The symbols can then be navigated by holding down [Ctrl][Shift] and pressing the up/down keys.

IntelliSense has been improved to allow for acronyms based on Pascal casing. For example, typing String.INOE and then a non-alphanumeric character will convert the call toString.IsNullOrEmpty. This still doesn't prevent IntelliSense from interfering when you're writing code that doesn't exist, as you would with a unit test.

The Suggestion Completion mode allows you to type freely without IntelliSense changing the text you typed. You can toggle between Standard and Suggestion Completion modes by pressing [Ctrl][Alt]space.

IntelliSense for JavaScript has seen the most improvement, as it is now able to determine the correct structure of a variable even after the structure is changed.

In the past, I would use .NET Reflector or another tool to analyze a user's call hierarchy; now that functionality is built-in. Right-click the user and choose View Call Hierarchy, and calls to and from the user will be available for browsing.

NETBEANS 7.3.1

NetBeans is an open-source project dedicated to providing rock solid software development that address the needs of developers, users and the businesses who rely on NetBeans as a basis for their products; particularly, to enable them to develop these products quickly, efficiently and easily by leveraging the strengths of the Java platform and other relevant industry standards.

FEATURES:

Best Support for Latest Java Technologies

NetBeans IDE provides first-class comprehensive support for the newest Java technologies and latest Java enhancements before other IDEs. It is the first IDE providing support for JDK 7, Java EE 7, and JavaFX 2.

With its constantly improving Java Editor, many rich features and an extensive range of tools, templates and samples, NetBeans IDE sets the standard for developing with cutting edge technologies out of the box.

Fast & Smart Co de Editing

An IDE is much more than a text editor. The NetBeans Editor indents lines, matches words and brackets, and highlights source code syntactically and semantically. It also provides code templates, coding tips, and refactoring tools.

The editor supports many languages from Java, C/C++, XML and HTML, to PHP, Groovy, Javadoc, JavaScript and JSP. Because the editor is extensible, you can plug in support for many other languages.

Easy & Efficient Project Management

Keeping a clear overview of large applications, with thousands of folders and files, and millions of lines of code, is a daunting task. NetBeans IDE provides different views of your data, from multiple project windows to helpful tools for setting up your applications and managing them efficiently, letting you drill down into your data quickly and easily, while giving you versioning tools via Subversion, Mercurial, and Git integration out of the box.

Rapid User Interface Development

Design GUIs for Java EE, Java SE, and Java ME applications quickly and smoothly by dragging and positioning GUI components from a palette into the NetBeans Editor.

For Java SE applications, the NetBeans GUI Builder automatically takes care of correct spacing and alignment, while supporting in-place editing, as well. The GUI builder is so intuitive that it has been used to prototype GUIs at customer presentations.

Write Bug Free Code

The cost of buggy code increases the longer it remains unfixed. NetBeans provides static analysis tools, especially integration with the widely used FindBugs tool, for identifying and fixing common problems in Java code. In addition, the NetBeans Debugger lets you place breakpoints in your source code, add field watches, step through your code, run into methods, take snapshots and monitor execution as it occurs.

ECLIPSE IDE FOR ANDROID

In computer programming, **Eclipse** is a multi-language Integrated development environment (IDE) comprising a base workspace and an extensible plug-in system for customizing the environment. It is written mostly in Java. It can be used to develop applications in Java and, by means of various plug-ins, other programming languagesincluding Ada, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Perl, PHP,Python, R, Ruby (including Ruby on Rails framework), Scala, Clojure, Groovy, Scheme, and Erlang. It can also be used to develop packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

FRONT END

WPF (WINDOWS PRESENTATION FRAMEWORK)

Windows Presentation Foundation (WPF) is a next-generation presentation system for building Windows client applications with visually stunning user experiences. With WPF, you can create a wide range of both standalone and browser-hosted applications.

Windows Presentation Foundation (WPF) provides developers with a unified programming model for building rich Windows smart client user experiences that incorporate UI, media, and documents. Windows Presentation Foundation (WPF) is a next-generation presentation system for building Windows client applications with visually stunning user experiences. With WPF, you can create a wide range of both standalone and browser-hosted applications. The core of WPF is a resolution-independent and vector-based rendering engine that is built to take advantage of modern graphics hardware. WPF extends the core with a comprehensive set of application-development features that include Extensible Application Markup Language (XAML), controls, data binding, layout, 2-D and 3-D graphics, animation, styles, templates, documents, media, text, and typography. WPF is included in the Microsoft .NET Framework, so you can build applications that incorporate other elements of the .NET Framework class library.

The core of WPF is a resolution-independent and vector-based rendering engine that is built to take advantage of modern graphics hardware. WPF extends the core with a comprehensive set of application-development features that include Extensible Application Markup Language (XAML), controls, data binding, layout, 2-D and 3-D graphics, animation, styles, templates, documents, media, text, and typography. WPF is included in the Microsoft .NET Framework, so you can build applications that incorporate other elements of the .NET Framework class library.

PROGRAMMING WITH WPF

WPF exists as a subset of .NET Framework types that are for the most part located in the System.Windows namespace. If you have previously built applications with .NET Framework using managed technologies like ASP.NET and Windows Forms, the fundamental WPF programming experience should be familiar; you instantiate classes, set properties, call methods, and handle events, all using your favorite .NET Framework programming language, such as C# or Visual Basic.

MARKUP & CODE-BEHIND

WPF offers additional programming enhancements for Windows client application development. One obvious enhancement is the ability to develop an application using both markup and code-behind, an experience that ASP.NET developers should be familiar with. You generally use Extensible Application Markup Language (XAML) markup to implement the appearance of an application while using managed programming languages (code-behind) to implement its behavior.

SECURITY

Because XBAPs are hosted in a browser, security is important. In particular, a partial-trust security sandbox is used by XBAPs to enforce restrictions that are less than or equal to the

restrictions imposed on HTML-based applications. Furthermore, each HTML feature that is safe to run from XBAPs in partial trust has been tested using a comprehensive security process.

CONTROLS

The user experiences that are delivered by the application model are constructed controls. In WPF, "control" is an umbrella term that applies to a category of WPF classes that are hosted in either a window or a page, have a user interface (UI), and implement some behavior.

WPF CONTROLS BY FUNCTION

The built-in WPF controls are listed here.

- Buttons: Button and RepeatButton.
- Data Display: DataGrid, ListView,and TreeView.
- Date Display and Selection: Calendar and DatePicker.
- Dialog Boxes: OpenFileDialog, PrintDialog, and SaveFileDialog.
- Digital Ink: InkCanvas and InkPresenter.
- Documents: DocumentViewer, FlowDocumentPageViewer, FlowDocumentReader, FlowDocumentScrollViewer, andStickyNoteControl.
- Input: TextBox, RichTextBox, and PasswordBox.
- Layout: Border, BulletDecorator, Canvas, DockPanel, Expander, Grid, GridView, GridSplitter, GroupBox, Panel,ResizeGrip, Separator, ScrollBar, ScrollViewer, StackPanel, Thumb, Viewbox, VirtualizingStackPanel, Window, andWrapPanel.
- Media: Image, MediaElement, and SoundPlayerAction.
- Menus: ContextMenu, Menu, and ToolBar.
- Navigation: Frame, Hyperlink, Page, NavigationWindow, and TabControl.
- Selection: CheckBox, ComboBox, ListBox, RadioButton, and Slider.
- User Information: AccessText, Label, Popup, ProgressBar, StatusBar, TextBlock, and ToolTip.

LAYOUT

When you create a UI, you arrange your controls by location and size to form a layout. A key requirement of any layout is to adapt to changes in window size and display settings. Rather than forcing you to write the code to adapt a layout in these circumstances, WPF provides a first-class, extensible layout system for you.
The cornerstone of the layout system is relative positioning, which increases the ability to adapt to changing window and display conditions. In addition, the layout system manages the

negotiation between controls to determine the layout. The negotiation is a two-step process: first, a control tells its parent what location and size it requires; second, the parent tells the control what space it can have.
The layout system is exposed to child controls through base WPF classes. For common layouts such as grids, stacking, and docking, WPF includes several layout controls:

- Canvas : Child controls provide their own layout.
- DockPanel : Child controls are aligned to the edges of the panel.
- Grid : Child controls are positioned by rows and columns.
- StackPanel : Child controls are stacked either vertically or horizontally.
- VirtualizingStackPanel : Child controls are virtualized and arranged on a single line that is either horizontally or vertically oriented.
- WrapPanel : Child controls are positioned in left-to-right order and wrapped to the next line when there are more controls on the current line than space allows.

EXTENSIBLE APPLICATION MARKUP LANGUAGE (XAML)

XAML stands for Extensible Application Markup Language. Its a simple language based on XML to create and initialize .NET objects with hierarchical relations. Although it was originally invented for WPF it can by used to create any kind of object trees.

Today XAML is used to create user interfaces in WPF, Silverlight, declare workflows in WF and for electronic paper in the XPS standard.

All classes in WPF have parameter less constructors and make excessive usage of properties. That is done to make it perfectly fit for XML languages like XAML.

All you can do in XAML can also be done in code. XAML ist just another way to create and initialize objects. You can use WPF without using XAML. It's up to you if you want to declare it in XAML or write it in code. Declare your UI in XAML has some advantages:

- XAML code is short and clear to read
- Separation of designer code and logic
- Graphical design tools like Expression Blend require XAML as source.
- The separation of XAML and UI logic allows it to clearly separate the roles of designer and developer.

GRAPHICS

WPF introduces an extensive, scalable, and flexible set of graphics features that have the following benefits:

- **Resolution-independent and device-independent graphics**. The basic unit of measurement in the WPF graphics system is the device independent pixel, which is 1/96th of an inch, regardless of actual screen resolution, and provides the foundation for resolution-independent and device-independent rendering. Each device-independent pixel automatically scales to match the dots-per-inch (dpi) setting of the system it renders on.

- **Improved precision**. The WPF coordinate system is measured with double-precision floating-point numbers rather than single-precision. Transformations and opacity values are also expressed as double-precision. WPF also supports a wide color gamut (scRGB) and provides integrated support for managing inputs from different color spaces.
- **Advanced graphics and animation support**. WPF simplifies graphics programming by managing animation scenes for you; there is no need to worry about scene processing, rendering loops, and bilinear interpolation. Additionally, WPF provides hit-testing support and full alpha-compositing support.
- **Hardware acceleration**. The WPF graphics system takes advantage of graphics hardware to minimize CPU usage.

XML

Extensible Markup Language (**XML**) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications, all free open standards.

The design goals of XML emphasize simplicity, generality, and usability over the Internet. It is a textual data format with strong support via Unicode for the languages of the world. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures, for example in web services.

Many application programming interfaces (APIs) have been developed to aid software developers with processing XML data, and several schema systems exist to aid in the definition of XML-based languages.

HTML

HyperText Markup Language (**HTML**) is the main markup language for creating web pages and other information that can be displayed in a web browser.

HTML is written in the form of HTML elements consisting of *tags* enclosed in angle brackets (like `<html>`), within the web page

content. HTML tags most commonly come in pairs like `<h1>` and `</h1>`, although some tags represent *empty elements* and so are unpaired, for example `<img>`. The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called *opening tags* and *closing tags*). In between these tags web designers can add text, further tags, comments and other types of text-based content.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

PROGRAMMING FRAMEWORK

.NET 4.5

The .NET Framework is a development platform for building apps for Windows, Windows Phone, Windows Server, and Windows Azure. It consists of the common language runtime (CLR) and the .NET Framework class library, which includes classes, interfaces, and value types that support an extensive range of technologies. The .NET Framework provides a managed execution environment, simplified development and deployment, and integration with a variety of programming languages, including Visual Basic and Visual C#.

.NET FRAMEWORK CLASS LIBRARIES

The .NET Framework class library is a library of classes, interfaces, and value types that provide access to system functionality. It is the foundation on which .NET Framework applications, components, and controls are built. The namespaces and namespace categories in the class library are listed in the following table and documented in detail in this reference. The namespaces and categories are listed by usage, with the most frequently used namespaces appearing first.

Namespace	Description
System	The System namespace contains fundamental classes and base classes that define commonly-used value and reference data types, events and event handlers, interfaces, attributes, and processing exceptions.
System.Activities	The System.Activities namespaces contain all the classes necessary to create and work with activities in Window Workflow Foundation.
System.AddIn	The System.AddIn namespaces contain types used to identify, register, activate, and control add-ins, and to allow add-ins to communicate with a host application.
System.CodeDom	The System.CodeDom namespaces contain classes that represent the elements of a source code document and that support the generation and compilation of source code in supported programming languages.
System.Collections	The System.Collections namespaces contain types that define various standard, specialized, and generic collection objects.
System.ComponentModel	The System.ComponentModel namespaces contain types that implement the run-time and design-time behavior of components and controls. Child namespaces support the Managed Extensibility Framework (MEF), provide attribute classes that define metadata for ASP.NET Dynamic Data controls, and contain types that let you define the design-time behavior of components and their user interfaces.

System.Configuration	The System.Configuration namespaces contain types for handling configuration data, such as data in machine or application configuration files. Child namespaces contain types that are used to configure an assembly, to write custom installers for components, and to support a pluggable model for adding functionality to, or removing functionality from, both client and server applications.
System.Data	The System.Data namespaces contain classes for accessing and managing data from diverse sources. The top-level namespace and a number of the child namespaces together form the ADO.NET architecture and ADO.NET data providers. For example, providers are available for SQL Server, Oracle, ODBC, and OleDB. Other child namespaces contain classes used by the ADO.NET Entity Data Model (EDM) and by WCF Data Services.
System.Deployment	The System.Deployment namespaces contain types that support deployment of ClickOnce applications.
System.Device.Location	The System.Device.Location namespace allows application developers to easily access the computer's location by using a single API. Location information may come from multiple providers, such as GPS, Wi-Fi triangulation, and cell phone tower triangulation. The System.Device.Location classes provide a single API to encapsulate the multiple location providers on a computer and support seamless prioritization and transitioning between them. As a result, application developers who use this API do not need to tailor applications to specific hardware configurations.
System.Diagnostics	The System.Diagnostics namespaces contain types that enable you to interact with system processes, event logs, and performance counters. Child namespaces contain types to interact with code analysis tools, to support contracts, to extend design-time support for application monitoring and instrumentation, to log event data using the Event Tracing for Windows (ETW) tracing subsystem, to read to and write from event logs and collect performance data, and to read and write debug symbol information.

System.DirectoryServices	The System.DirectoryServices namespaces contain types that provide access to Active Directory from managed code.
System.Drawing	The System.Drawing parent namespace contains types that support basic GDI+ graphics functionality. Child namespaces support advanced two-dimensional and vector graphics functionality, advanced imaging functionality, and print-related and typographical services. A child namespace also contains types that extend design-time user-interface logic and drawing.
System.Dynamic	The System.Dynamic namespace provides classes and interfaces that support Dynamic Language Runtime.
System.EnterpriseServices	The System.EnterpriseServices namespaces contain types that define the COM+ services architecture, which provides an infrastructure for enterprise applications. A child namespace supports Compensating Resource Manager (CRM), a COM+ service that enables non-transactional objects to be included in Microsoft Distributed Transaction Coordinator (DTC) transactions. Child namespaces are described briefly in the following table and documented in detail in this reference.
System.Globalization	The System.Globalization namespace contains classes that define culture-related information, including language, country/region, calendars in use, format patterns for dates, currency, and numbers, and sort order for strings. These classes are useful for writing globalized (internationalized) applications. Classes such as StringInfo andTextInfo provide advanced globalization functionalities, including surrogate support and text element processing.
System.IdentityModel	The System.IdentityModel namespaces contain types that are used to provide authentication and authorization for .NET applications.
System.IO	The System.IO namespaces contain types that support input and output, including the ability to read and write

	data to streams either synchronously or asynchronously, to compress data in streams, to create and use isolated stores, to map files to an application's logical address space, to store multiple data objects in a single container, to communicate using anonymous or named pipes, to implement custom logging, and to handle the flow of data to and from serial ports.
System.Linq	The System.Linq namespaces contain types that support queries that use Language-Integrated Query (LINQ). This includes types that represent queries as objects in expression trees.
System.Management	The System.Management namespaces contain types that provide access to management information and management events about the system, devices, and applications instrumented to the Windows Management Instrumentation (WMI) infrastructure. These namespaces also contain types necessary for instrumenting applications so that they expose their management information and events through WMI to potential customers.
System.Media	The System.Media namespace contains classes for playing sound files and accessing sounds provided by the system.
System.Messaging	The System.Messaging namespaces contain types that enable you to connect to, monitor, and administer message queues on the network and to send, receive, or peek messages. A child namespace contains classes that can be used to extend design-time support for messaging classes.
System.Net	The System.Net namespaces contain classes that provide a simple programming interface for a number of network protocols, programmatically access and update configuration settings for the System.Net namespaces, define cache policies for web resources, compose and send e-mail, represent Multipurpose Internet Mail Exchange (MIME) headers, access network traffic data and network address information, and access peer-to-peer networking functionality. Additional child namespaces

	provide a managed implementation of the Windows Sockets (Winsock) interface and provide access to network streams for secure communications between hosts.
System.Numerics	The System.Numerics namespace contains numeric types that complement the numeric primitives, such as Byte, Double, and Int32, that are defined by the .NET Framework.
System.Printing	The System.Printing namespaces contain types that support printing, that provide access to the properties of print system objects and enable rapid copying of their property settings to another object of the same type, and that support the interconversion of managed System.PrintTicket objects and unmanaged GDI DEVMODE structures.
System.Reflection	The System.Reflection namespaces contain types that provide a managed view of loaded types, methods, and fields, and that can dynamically create and invoke types. A child namespace contains types that enable a compiler or other tool to emit metadata and Microsoft intermediate language (MSIL).
System.Resources	The System.Resources namespaces contain types that enable developers to create, store, and manage an application's culture-specific resources.
System.Runtime	The System.Runtime namespaces contain types that support an application's interaction with the common language runtime, and types that enable features such as application data caching, advanced exception handling, application activation within application domains, COM interop, distributed applications, serialization and deserialization, and versioning. Additional namespaces enable compiler writers to specify attributes that affect the run-time behavior of the common language runtime, define a contract for reliability between a set of code and other code that takes a dependency on it, and implement a persistence provider for Windows Communication Foundation (WCF).

System.Security	The System.Security namespaces contain classes that represent the .NET Framework security system and permissions. Child namespaces provide types that control access to and audit securable objects, allow authentication, provide crytographic services, control access to operations and resources based on policy, and support rights management of application-created content.
System.ServiceModel	The System.ServiceModel namespaces contain the types necessary to build Windows Communication Foundation (WCF) service and client applications.
System.ServiceProcess	The System.ServiceProcess namespaces contain types that enable you to implement, install, and control Windows service applications and extend design-time support for Windows service applications.
System.Speech	The System.Speech namespaces contain types that support speech recognition.
System.Text	The System.Text namespaces contain types for character encoding and string manipulation. A child namespace enables you to process text using regular expressions.
System.Threading	The System.Threading namespaces contain types that enable multithreaded programming. A child namespace provides types that simplify the work of writing concurrent and asynchronous code.
System.Timers	The System.Timers namespace provides the Timer component, which allows you to raise an event on a specified interval.
System.Transactions	The System.Transactions namespaces contain types that support transactions with multiple, distributed participants, multiple phase notifications, and durable enlistments. A child namespace contains types that describe the configuration options used by the

	System.Transactions types.
System.Web	The System.Web namespaces contain types that enable browser/server communication. Child namespaces include types that support ASP.NET forms authentication, application services, data caching on the server, ASP.NET application configuration, dynamic data, HTTP handlers, JSON serialization, incorporating AJAX functionality into ASP.NET, ASP.NET security, and web services.
System.Windows	The System.Windows namespaces contain types used in Windows Presentation Foundation (WPF) applications, including animation clients, user interface controls, data binding, and type conversion. System.Windows.Forms and its child namespaces are used for developing Windows Forms applications.
System.Workflow	The System.Workflow namespaces contain types used to develop applications that use Windows Workflow Foundation. These types provide design time and run-time support for rules and activities, to configure, control, host, and debug the workflow runtime engine.
System.Xaml	The System.Xaml namespaces contain types that support parsing and processing the Extensible Application Markup Language (XAML).
System.Xml	The System.Xml namespaces contain types for processing XML. Child namespaces support serialization of XML documents or streams, XSD schemas, XQuery 1.0 and XPath 2.0, and LINQ to XML, which is an in-memory XML programming interface that enables easy modification of XML documents.
Accessibility	The Accessibility and all of its exposed members are part of a managed wrapper for the Component Object Model (COM) accessibility interface.
Microsoft.Activities	The Microsoft.Activities namespaces contain types that

	support MSBuild and debugger extensions for Windows Workflow Foundation applications.
Microsoft.Aspnet.Snapin	The Microsoft.Aspnet.Snapin namespace defines the types necessary for the ASP.NET management console application to interact with Microsoft Management Console (MMC). For more information, see "MMC Programmer's Guide" in the MSDN Library.
Microsoft.Build	The Microsoft.Build namespaces contain types that provide programmatic access to, and control of, the MSBuild engine.
Microsoft.CSharp	The Microsoft.CSharp namespaces contain types that support compilation and code generation of source code written in the C# language, and types that support interoperation betwen the dynamic language runtime (DLR) and C#.
Microsoft.Data.Entity.Build.Tasks	The Microsoft.Data.Entity.Build.Tasks namespace contains two MSBuild tasks that are used by the ADO.NET Entity Data Model Designer (Entity Designer).
Microsoft.JScript	The Microsoft.JScript namespaces contain classes that support compilation and code generation using the JScript language.
Microsoft.SqlServer.Server	The Microsoft.SqlServer.Server namespace contains classes, interfaces, and enumerations that are specific to the integration of the Microsoft .NET Framework common language runtime (CLR) into Microsoft SQL Server, and the SQL Server database engine process execution environment.
Microsoft.VisualBasic	The Microsoft.VisualBasic namespaces contain classes that support compilation and code generation using the Visual Basic language. Child namespaces contain types that provide services to the Visual Basic compiler and types that include support for the Visual Basic application

	model, the My namespace, lambda expressions, and code conversion.
Microsoft.VisualC	The Microsoft.VisualC namespaces contain types that support the Visual C++ compiler and types that implement the STL/CLR Library and the generic interface to the STL/CLR Library.
Microsoft.Win32	The Microsoft.Win32 namespaces provide types that handle events raised by the operating system, that manipulate the system registry, and that represent file and operating system handles.
Microsoft.Windows	The Microsoft.Windows namespaces contain types that support themes and preview in Windows Presentation Framework (WPF) applications.
UIAutomationClientsideProviders	Contains a single type that maps client automation providers.
XamlGeneratedNamespace	Contains compiler-generated types that are not intended to be used directly from your code.

CODEIGNITER

CodeIgniter is an open source rapid development web application framework, for use in building dynamic web sites with PHP. "Its goal is to enable [developers] to develop projects much faster than writing code from scratch, by providing a rich set of libraries for commonly needed tasks, as well as a simple interface and logical structure to access these libraries." The first public version of CodeIgniter was released on February 28, 2006, and the latest stable version 2.1.4 was released July 8, 2013.

CodeIgniter is loosely based on the popular Model-View-Controller development pattern. While view and controller classes are a necessary part of development under CodeIgniter, models are optional.

ADT

The Android Developer Tools (ADT) plugin for Eclipse provides a professional-grade development environment for building Android apps. It's a full Java IDE with advanced features to help you build, test, debug, and package your Android apps.

Free, open-source, and runs on most major OS platforms.

FEATURES:

Full Java IDE

- Android-specific refactoring, quick fixes, integrated navigation between Java and XML resources.
- Enhanced XML editors for Android XML resources.
- Static analysis tools to catch performance, usability, and correctness problems.
- Build support for complex projects, command-line support for CI through Ant. Includes ProGuard and app-signing.
- Template-based wizard to create standard Android projects and components.

Graphical UI Builders

- Build rich Android UI with drag and drop.
- Visualize your UI on tablets, phones, and other devices. Switch themes, locales, even platform versions instantly, without building.
- Visual refactoring lets you extracts layout for inclusion, convert layouts, extract styles.
- Editor support for working with custom UI components.
-

Develop on Hardware Devices

- Use any commercial Android hardware device or multiple devices.
- Deploy your app to connected devices directy from the IDE.
- Live, on-device debugging, testing, and profiling.

Develop on Virtual Devices

- Emulate any device. Use custom screen sizes, keyboards, and other hardware components.
- Advanced hardware emulation, including camera, sensors, multitouch, and telephony.
- Develop and test for broad device compatibility.

Powerful Debugging

- Full Java debugger with on-device debugging and Android-specific tools.
- Built-in memory analysis, performance/CPU profiling, OpenGL ES tracing.
- Graphical tools for debugging and optimizing UI, runtime inspecton of UI structure and performance.
- Runtime graphical analysis of your app's network bandwidth usage.

Testing

- Fully instrumentated, scriptable test environment.
- Integrated reports using standard test UI.
- Create and run unit tests on hardware devices or emulator.

Native Development

- Support for compiling and packaging existing code written in C or C++.

- Support for packaging multiple architectures in a single binary, for broad compatibility.

DATABASE/BACKEND:

MYSQL

MySQL is the world's most popular open source database software, with over 100 million copies of its software downloaded or distributed throughout its history.

The MySQL Community Edition includes:

- Pluggable Storage Engine Architecture
- Multiple Storage Engines: InnoDB , MyISAM, NDB (MySQL Cluster),Memory ,Merge , Archive, CSV
- MySQL Replication to improve application performance and scalability
- MySQL Partitioning to improve performance and management of large database applications
- Stored Procedures to improve developer productivity

DETAILED FEATURES OF MYSQL

The following list shows the most important properties of MySQL. This section is directed to the reader who already has some knowledge of relational databases. We will use some terminology from the relational database world without defining our terms exactly. On the other hand, the explanations should make it possible for database novices to understand to some extent what we are talking about.

Relational Database System: Like almost all other database systems on the market, MySQL is a relational database system.

Client/Server Architecture: MySQL is a client/server system. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the

server; that is, they query data, save changes, etc. The clients can run on the same computer as the server or on another computer (communication via a local network or the Internet).

Almost all of the familiar large database systems (Oracle, Microsoft SQL Server, etc.) are client/server systems. These are in contrast to the file-server systems, which include Microsoft Access, dBase and FoxPro. The decisive drawback to file-server systems is that when run over a network, they become extremely inefficient as the number of users grows.

SQL compatibility: MySQL supports as its database language -- as its name suggests – SQL (Structured Query Language). SQL is a standardized language for querying and updating data and for the administration of a database. There are several SQL dialects (about as many as there are database systems). MySQL adheres to the current SQL standard (at the moment SQL:2003), although with significant restrictions and a large number of extensions.

Through the configuration setting sql-mode you can make the MySQL server behave for the most part compatibly with various database systems. Among these are IBM DB/2 and Oracle. (The setting sql-mode changes some of the syntax conventions, and performs no miracles.

SubSELECTs: Since version 4.1, MySQL is capable of processing a query in the form SELECT * FROM table1 WHERE x IN (SELECT y FROM table2) (There are also numerous syntax variants for subSELECTs.)

Views: Put simply, views relate to an SQL query that is viewed as a distinct database object and makes possible a particular view of the database. MySQL has supported views since version 5.0.

Stored procedures: Here we are dealing with SQL code that is stored in the database system.

Stored procedures (SPs for short) are generally used to simplify certain steps, such as inserting or deleting a data record. For client programmers this has the advantage that they do not have to process the tables directly, but can rely on SPs. Like views, SPs help in the administration of large database projects. SPs can also increase efficiency. MySQL has supported SPs since version 5.0.

Triggers: Triggers are SQL commands that are automatically executed by the server in certain database operations (INSERT, UPDATE, and DELETE). MySQL has supported triggers in a limited form from version 5.0, and additional functionality is promised for version 5.1.

Unicode: MySQL has supported all conceivable character sets since version 4.1, including Latin-1, Latin-2, and Unicode (either in the variant UTF8 or UCS2).

User interface: There are a number of convenient user interfaces for administering a MySQL server.

Full-text search: Full-text search simplifies and accelerates the search for words that are located within a text field. If you employ MySQL for storing text (such as in an Internet

discussion group), you can use full-text search to implement simply an efficient search function.

Replication: Replication allows the contents of a database to be copied (replicated) onto a number of computers. In practice, this is done for two reasons: to increase protection against system failure (so that if one computer goes down, another can be put into service) and to improve the speed of database queries.

Transactions: In the context of a database system, a transaction means the execution of several database operations as a block. The database system ensures that either all of the operations are correctly executed or none of them. This holds even if in the middle of a transaction there is a power failure, the computer crashes, or some other disaster occurs. Thus, for example, it cannot occur that a sum of money is withdrawn from account A but fails to be deposited in account B due to some type of system error.

Transactions also give programmers the possibility of interrupting a series of already executed commands (a sort of revocation). In many situations this leads to a considerable simplification of the programming process. In spite of popular opinion, MySQL has supported transactions for a long time. One should note here that MySQL can store tables in a variety of formats. The default table format is called MyISAM, and this format does not support transactions. But there are a number of additional formats that do support transactions. The most popular of these is InnoDB, which will be described extensively in this book.

Foreign key constraints: These are rules that ensure that there are no cross references in linked tables that lead to nowhere. MySQL supports foreign key constraints for InnoDB tables.

GIS functions: Since version 4.1, MySQL has supported the storing and processing of two-dimensional geographical data. Thus MySQL is well suited for GIS (geographic information systems) applications.

Programming languages: There are quite a number of APIs (application programming interfaces) and libraries for the development of MySQL applications. For client programming you can use, among others, the languages C, C++, Java, Perl, PHP, Python, and Tcl.

ODBC: MySQL supports the ODBC interface Connector/ODBC. This allows MySQL to be addressed by all the usual programming languages that run under Microsoft Windows (Delphi, Visual Basic, etc.). The ODBC interface can also be implemented under Unix, though that is seldom necessary.

Windows programmers who have migrated to Microsoft's new .NET platform can, if they wish, use the ODBC provider or the .NET interface Connector/NET.

Platform independence: It is not only client applications that run under a variety of operating systems; MySQL itself (that is, the server) can be executed under a number of operating systems. The most important are Apple Macintosh OS X, Linux, Microsoft

Windows, and the countless Unix variants, such as AIX, BSDI, FreeBSD, HP-UX, OpenBSD, Net BSD, SGI Iris, and Sun Solaris.

Speed: MySQL is considered a very fast database program. This speed has been backed up by a large number of benchmark.

SQLITE

SQLite is a relational database management system contained in a small C programming library. In contrast to other database management systems, SQLite is not a separate process that is accessed from the client application, but an integral part of it.

SQLite is ACID-compliant and implements most of the SQL standard, using a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity.

SQLite is a popular choice as embedded database for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems, among others. SQLite has many bindings to programming languages.

The source code for SQLite is in the public domain.

IDE FOR DATABASE

MYSQL WORKBENCH

MySQL Workbench is a visual database design tool that integrates SQL evelopment,administration, database design, creation and maintenance into a single integrated development environment for the MySQL database system. It is the successor to DBDesigner 4 from fabFORCE.net, and replaces the previous package of software,MySQL GUI Tools Bundle. MySQL Workbench enables a DBA, developer, or data architect to visually design,

generate, and manage all types of databases including Web, OLTP, and data warehouse databases. It includes everything a data modeler needs for creating complex ER models, and also delivers key features for performing difficult change management and documentation tasks that normally require much time and effort. MySQL Workbench is available on Windows, Linux and Mac OS.

BENEFITS

- Simplifies database design and maintenance
- Automates time-consuming and error-prone tasks
- Enables data architects to visualize requirements, communicate with stakeholders, and resolve design issues before a major investment of time and resources is made
- Enables model-driven database design—the most efficient methodology for creating valid and well-performing databases—while providing the flexibility to respond to evolving business requirements
- Provides capabilities to forward-engineer physical database designs and reverse-engineer existing databases
- Allows you to import SQL scripts to build models and export models to DDL scripts that can be run at a later time
- Enables you to compare two live databases or a model and a live database, visually see the differences, and perform a synchronization between a model and a live database or vice versa
- Simplifies the documentation of database designs, providing a point-and-click process that delivers documentation in HTML or plain-text format

TOOLS

The three main tools of MySQL Workbench are:

- SQL Development
- Data Modelling
- Server Administration

C# - C SHARP

C# is a type-safe, object-oriented language that is simple yet powerful, allowing programmers to build a breadth of applications. C# is a multi-paradigm programming

language encompassing imperative, declarative, functional, generic, object-oriented(class-based), and component-oriented programming disciplines. It was developed by Microsoft within the .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270). C# is one of the programming languages designed for the Common Language Infrastructure.

C# is intended to be a simple, modern, general-purpose, object-oriented programming language.

CHARACTERISTICS OF C#:

C# was developed to bring rapid development to C++ without sacrificing the power and control of C and C++. C# provides various characteristics, which are:

Simple:

C# eliminates the use of tedious operators such as -->, and pointers. C# treats inter and Boolean as two different data types, which enable the compiler to recognize the use of = in place of = = with if statement.

Consistent:-

C# supports only one integer type and there is no limitation of range.

Modern:-

C# contains various features necessary to develop web applications. Following are the features of C#:

It provides automatic garbage collection.

It provides robust security model.

It provides decimal data type for financial application.

It provides modern approach for debugging.

It provides a rich intrinsic model for error handling.

Object Oriented:-

C# supports all the features of object oriented language such as encapsulation, inheritance and polymorphism. It treats everything as an object and there are no global

functions, variables and constants in C#.

Type Safe:-

C# provides various type safe measures, which are:

Dynamically allocated objects and arrays are initialized to zero.

Products an error message while using an uninitialized variable.

Checks the range of an array and warns when the access goes out of bound.

Unsafe casts are not allowed.

Enforces overflow checking in arithmetic operations.

Versionable:-

C# supports versioning that enables the existing applications to run on different versions with the help of new and override command.

Compatible:

C# contains the .NET specifications and therefore, allows inter operation with other .NET languages.

Flexible:-

C# does not support pointers but you may use pointers to manipulate the data of certain classes and methods by declaring them unsafe.

Inter-operability:

C# enables a program to call out any native API. It also allows the use of COM objects written in different languages.

JAVA

Java is a general-purpose, concurrent, class-based, object-oriented computer programming language that is specifically designed to

have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to bytecode (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is, as of 2012, one of the most popular programming languages in use, particularly for client-server web applications, with a reported 10 million users.[10][11] Java was originally developed by James Gosling at Sun Microsystems (which has since merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

The original and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1991 and first released in 1995. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plug-in for applets).

PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers.[2] Originally created by Rasmus Lerdorf in 1995, the reference implementation of PHP is now produced by The PHP Group.[3] While PHP originally stood for *Personal Home Page*, it now stands for *PHP: Hypertext Preprocessor*, a recursive acronym.

PHP code is interpreted by a web server with a PHP processor module, which generates the resulting web page: PHP commands can be embedded directly into an HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone graphical applications.

PHP is free software released under the PHP License, which is incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term *PHP*. PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

LIBRARIES

TWITTER BOOTSTRAP

Bootstrap is a free collection of tools for creating websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.

It has been the most popular project in GitHub and has been used by NASA and MSNBC among others.

FLEXI AUTH

What is flexi auth?

flexi auth is a free open source user authentication/login library for use with the CodeIgniter 2.0+ framework.
The flexi auth library initially started out as a modified version of the popular Ion Auth library. As the original library was tweaked with

feature after feature being added, the original code base had transformed into a new library all of its own.
For those that have used the Ion Auth library, the general structure of the library may be familiar, but to help anyone wanting to get a running start with using flexi auth, there is an comprehensive user guide and demo detailing covering every function within the library.

flexi auth is designed with modularised features that can be mixed and matched, turned on or off, and can be customised to suit your requirements.

MAHANA MESSAGING LIBRARY

A small library to help jump start your internal messaging system, for the CodeIgniter framework

FEATURES:

- Create new thread
- Show thread
- Show messages
- Show all threads

OTHER TECHNOLOGIES

REST PROTOCOL

Representational State Transfer (**REST**) is an architectural style that abstracts the architectural elements within a distributed hypermedia system. REST ignores the details of component implementation and protocol syntax in order to focus on the roles of components, the constraints upon their interaction with other components, and their interpretation of significant data elements. REST has emerged as a predominant web API design model.

DRAWING TOOLS

DIA FOR DIAGRAM DRAWING &MODELING

Dia is free and open source general-purpose diagramming software, developed as part of the GNOME project's office suite and was originally created by Alexander Larsson. Dia uses a controlled single document interface (CSDI) similar to GIMP and Sodipodi.

Dia has a modular design with several shape packages available for different needs: flowchart, network diagrams, circuit diagrams, and more. It does not restrict symbols and connectors from various categories from being placed together.

Dia is a gtk+ based diagram creation program released under the GPL license.

Dia is inspired by the commercial Windows program 'Visio', though more geared towards informal diagrams for casual use. It can be used to draw many different kinds of diagrams. It currently has special objects to help draw entity relationship diagrams, UML diagrams, flowcharts, network diagrams, and many other diagrams. It is also possible to add support for new shapes by writing simple XML files, using a subset of SVG to draw the shape.

It can load and save diagrams to a custom XML format (gzipped by default, to save space), can export diagrams to a number of formats, including EPS, SVG, XFIG, WMF and PNG, and can print diagrams (including ones that span multiple pages).

GOOGLE SPREADSHEET INTERFACE:

With Google Spreadsheets, we can easily create, share, and edit spreadsheets online. Here are a few specific things we can do:

- *Import and export these file types: .xls, .csv, .txt and .ods. We can also export data to a PDF or an HTML file.*
- *Format cells and edit formulas so we can calculate results and make data look the way we want it.*
- *Chat in real time with others who are editing our spreadsheet.*
- *Embed a spreadsheet, or a section of a spreadsheet, in our blog or website.*

CACOO:: ONLINE DRAWING TOOL

Cacoo is a diagram creation tool that runs in your web browser.Multiple people can work together on the same diagram in real time.Diagrams can be published directly to websites, wikis, and blogs.

CREATING DIAGRAMS

- Elements can be dragged and drop to easily create diagrams.
- Elements can be linked together with connectors.
- Connectors automatically move when elements are repositioned.
- You can use a text box and put text anywhere you like.
- You can upload images from your PC and include them in Diagrams.
- You can take screenshots of your computer from within Cacoo.
- Smart styles can easily be applied to stencils.
- You can have multiple sheets in a diagram and use them as backgrounds or layers.
- When you move the objects on your canvas, they will be snapped at the objects or grids nearby and align automatically.
- Copying, pasting and other functionality of basic drawing software is also built in to Cacoo.
- All actions are stored so there are unlimited levels of undo.
- You can import an image from the other websites by indicating the URL.
- The imported image can be easily trimmed only using your mouse.
- According to your editing status, tips will be shown on the right bottom corner of the canvas.

COLLABORATION

- You can invite collaborators to work with you in Cacoo.
- Multiple people can edit a diagram in real time.
- There is a chat function in the editor so people can communicate while creating diagrams.
- People can leave comments about the diagrams.
- Each user can set their own user icon.
- When editing with multiple people, users icons appear on selected objects.
- Sharing diagrams become much smoother. Diagrams in the shared folders can be accessible and editable by people who you have shared the folder with.

SHARING DIAGRAMS

- If you keep the diagram private then other users can't see it.
- If you make the diagram URL public, then anyone who knows the URL can see it.
- Publishing a diagram to a blog can be useful in various ways.
- You can place code into blogs to create a slideshow
- Published images always display the most recent version.
- Diagrams can be exported to SVG format (Plus Plan users only) and PNG format. (More formats will be available in the future.)
- Diagrams can be posted to Twitter/Facebook/GoogleBuzz

- Diagrams can be displayed in SVG format for printing. (Plus Plan users only. A few browsers are not supported.)

MANAGING DIAGRAMS

- Diagrams can be placed into folders.
- Diagrams can be copied.
- Diagrams can be displayed as thumbnails or as a list.

LANGUAGES AND TIME ZONES

- All pages and notification e-mails support English and Japanese
- Users can enter text from almost all languages.
- Dates are displayed relative to your local time zone.

SECURITY

- Private diagrams can only be seen by users you select.
- URLs which you do not share can not be found by other users or search engines.
- All editing and management is protected by SSL.
- In order to access information about diagrams a Cacoo ID and password are requited.
- User passwords are encrypted on Cacoo's server.

API

- You can access Cacoo using the API.
- The Cacoo API supports OAuth and an API Key.

By using the Cacoo API you are able to interact with Cacoo from other services and applications.

Authorization Methods

There are two ways to access the Cacoo API.

1. API Key

The API key allows you make requests to the Cacoo API. You can make an API key here.

API Key

Append your API key to requests to the API to return data from your account.(Parameter name "apiKey")

Example: https://cacoo.com/api/v1/diagrams.json?apiKey=abcdefghijklmn

2. OAuth

OAuth 1.0a is supported as an authorization method for Cacoo. You can register applications here.

You can get your Access Token from the following links.

APPLICATIONS

Access Token:https://cacoo.com/oauth/access_token

Authorize:https://cacoo.com/oauth/authorize

Request Token:https://cacoo.com/oauth/request_token

VERSION CONTROL SYSTEM : GITHUB

GitHub is a web-based hosting service for software development projects that use the Git revision control system. GitHub offers both paid plans for private repositories, and free accounts for open source projects. As of May 2011, GitHub was the most popular open source code repository site.GitHub Inc. was founded in 2008 and is based in San Francisco, California.

DESCRIPTION

The site provides social networking functionality such as feeds, followers and the network graph to display how developers work on their versions of a repository.

GitHub also operates other services: a pastebin-style site called Gist that provides wikis for individual repositories and web pages that can be edited through a Git repository, a slide hosting service called Speaker Deck, and a web analytics platform called Gauges.

As of January 2010, GitHub is operated under the name GitHub, Inc.

The software that runs GitHub was written using Ruby on Rails and Erlang by GitHub, Inc. (previously known as Logical Awesome) developers Chris Wanstrath, PJ Hyett, and Tom Preston-Werner.

LIMITATIONS AND CONSTRAINTS

According to the terms of service,if an account's bandwidth usage significantly exceeds the average of other GitHub customers, the account's file hosting service may be immediately disabled or throttled until bandwidth consumption is reduced. In addition, while there is no hard limit, the guideline for the maximum size of a repository is one gigabyte.

GLOSSARY.

FMS File Management System

Apps Application

SRS Software Requirement Specification

DFD Data Flow Diagram

ERD Entity Relationship Diagram

GUI Graphical User Interface

UI	User Interface
DB	Database
API	Application Programming Interface
COCOMO	Constructive Cost Model
SDK	Sweater Development Kit
WPF	Windows Presentation Framework
XAML	Extensible application Markup Language
IDE	Integrated Development Environment
HTML	Hyper Text Markup Language
www	World Wide Web
DBMS	Database Management System
Sync	Synchronization
cs	C Sharp
KLOC	Estimated size of the software product expressed in Kilo
Tdev	Estimated time to develop the software, expressed in months.
Effort	Total effort required to develop the software product, expressed in person-month (PM).
PM	Person-month

----------Thank You----------