File Manager: an efficient solution for managing mobile storage

Erik Abels, James Erin, Jason Swarts,
Arielle Wilson, Piper Rainen, Denis Popel

Computer Science Department,
Baker University

Baldwin City, KS 66006

`denis.popel@bakeru.edu`

Abstract

This paper outlines research and development done to create an efficient management tool to manipulate files on mobile devices such as Pocket PCs, Smartphones, etc. The outcome of this software engineering project is an application, named *File Manager*, which is capable of secure file transfer between a mobile device and a server through an FTP (File Transfer Protocol) connection. The application has an intuitive user-friendly interface and numerous options to enable multiple user profiles. Application testing has been performed for various networking settings including wireless (e.g., Wi-Fi) and wired connections.

1 Introduction

Overview of the program

The File Manager program has been created to address an increasing demand for a flexible storage and management solution for mobile devices. The application allows for files to be shared between a handheld device and a server through a variety of networking connections including wireless. The main limitation of mobile devices is the amount of available memory (ranging from 16Mb to 512Mb) to store applications and data. Currently this limitation is addressed (a) by adding external memory slots with limited capacity, or (b) by designing applications for installing/uninstalling/moving programs and files in and out of mobile devices. The vast majority of applications for the latter solution make use of synchronization software connecting a mobile device with a selected stationary workstation. In order to avoid a specific linkage to a dedicated workstation, File Manager uses a configurable network storage and standard protocols for file transmission lifting the storage constraints. With the increasing number of handheld devices equipped with wireless technology, such a solution enables mobility and flexibility in file storage and management.

File Manager users

Corporate environments Entire businesses can host a file transfer server and allow their employees to access it through their mobile devices with the use of File Manager.

Professional users Backing up important data is a valuable mitigation plan and with File Manager users have a way to save vital information in case of a device failure or lack of memory.

Home users Everyday use at home is another way the File Manager can be valuable. This includes storing digital media like pictures and music, to being able to share information between family members.

Design

The Use Cases involved in the development of the File Manager can be seen in the Use Case Diagram (Figure 1) The different Use Cases include:

Pocket PC

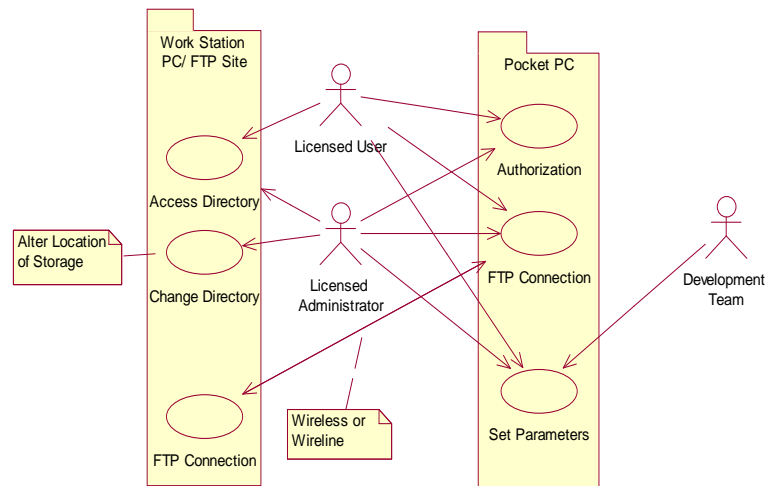- Authorization
- Set parameters
- FTP connection

Figure 1: Use Case diagram

Workstation PC/FTP site

- Access directory
- Change directory
- FTP connection

Each Use Case is discussed in greater detail next.

Authentication and authorization

Authentication and authorization is the first step when accessing the File Manager System. This Use Case depicted in Figure 2 gives the step by step process followed during the authentication and authorization. Conceptually this process validates the user as well as the parameters each user requires for the use of the File Manager.

Set parameters

Within the Set parameters Use Case, there are steps identified to create, edit, or delete a variety of settings on the File Manager System. Figure 3 gives the visual representation of the use case as it is conceptually intended. Both the users and administrators are able to update settings through the Set parameter Use Case with therein authorization level being the decision criteria for permission.
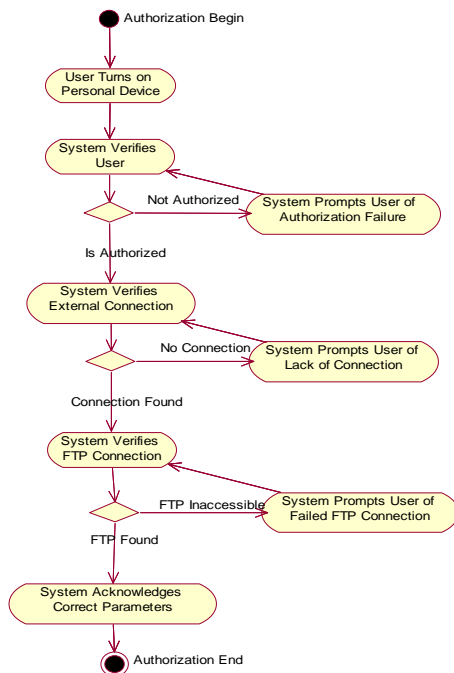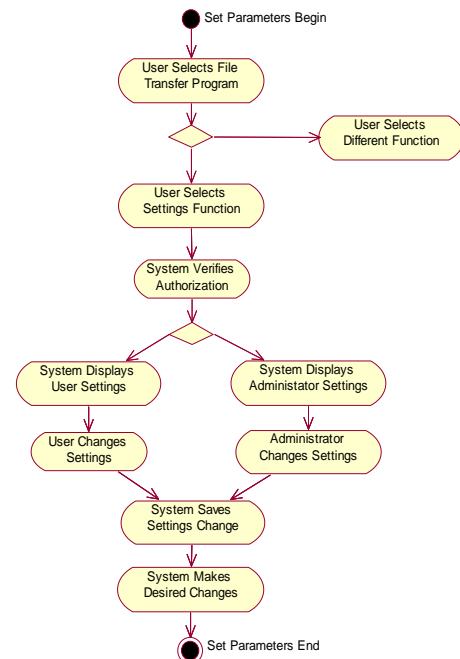
Figure 2: Authorization process

Figure 3: Set parameters process

FTP connection

The FTP connection Use Case is associated with both the Pocket PC as well as the FTP site. The diagram shown in Figure 4 gives both approaches to the use case and shows the different alternatives for the use case's path. Within the FTP connection lies the main functionality of the File Manager System. This Use Case illustrates the process identified to transfer to and from an FTP site.

Access directory

The Access directory Use Case takes place on the Workstation PC side of the Use Case Diagram. The user is able to access the location of their stored files through the FTP protocol and their personal computer. Accessing the storage site through the Pocket PC is possible by following the FTP Connection Use Case (Figure 5).

Change directory

The Change directory Use Case, seen in Figure 6, gives the process that is to be followed in order to change settings related to the FTP directory. This use case is to be accessed by the system administrator because of the control implications this edit can have on the data storage, as well as the performance of the File Manager system.
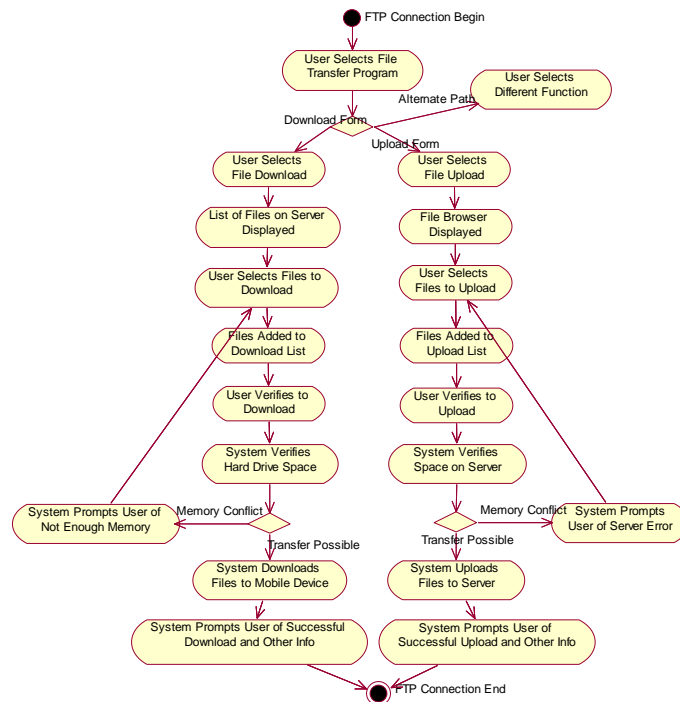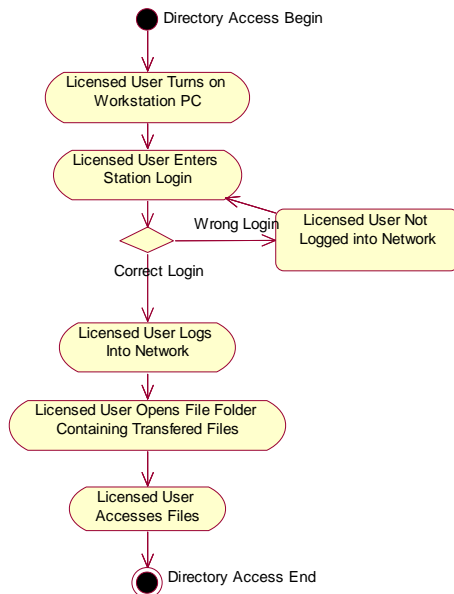
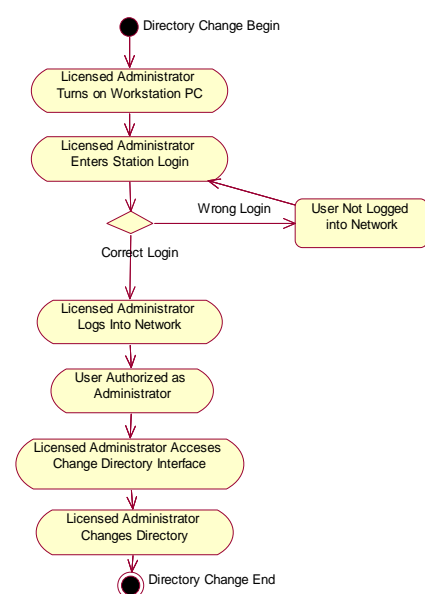Figure 4: FTP connection process

Figure 5: Access directory process

Figure 6: Change directory process

Application architecture

User Class This class handles the username and passwords for the File Manager system (Figure 7).

ProfilePreferences Class This class handles the FTP configurations and the preferences for that profile (Figure 7).

ProfileSelect Class This class handles the file reading and writing and works with the Security Class for the encryption (Figure 7).

Security Class This class handles all of the encryption and conversion from characters to the hexadecimal equivalent (Figure 8).

ErrorHandling Class This class handles the error codes and the error box display (Figure 9).

CompactFileList Class This is an abstract class to handle children (Figure 10).

CompactPPCFileList Class This class controls the list view of the files on the Pocket PC (Figure 10).

CompactFTPFileList Class This class controls the list vies of the files on the FTP Server (Figure 10).
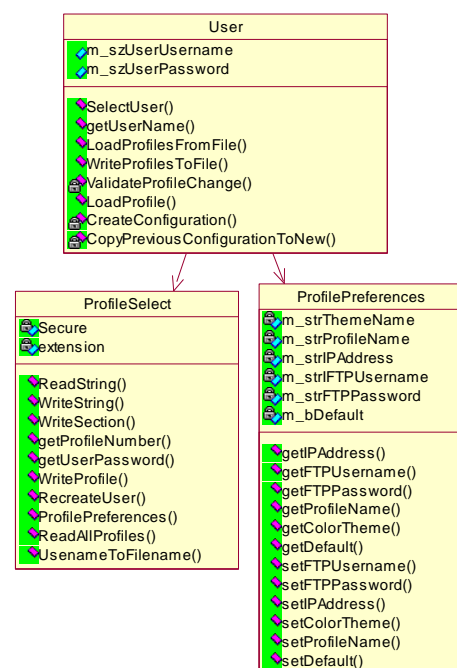
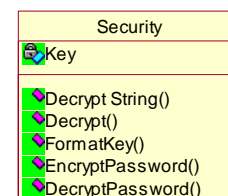Figure 7: User, ProfilePreferences and ProfileSelect classes
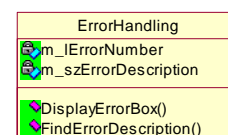
Figure 8: Security class

Figure 9: Error handling class

CompactFileList

m_cmbDirectoryPicker
m_lvFileList
m_strFilter
m_strUnformattedPath
m_strUnformattedInitialPath
m_PathDelimiter
m_strInitialPath
m_strPath
m_aFileArray
FileArray

MemoryForOutsideObjects()
AssignOutsideObjects()
InitializeComponent()
GetShortFilename()
PopulateFiles()
PopulateListView()
GetArrrayOfFilesToPopulate()
GetArrayOfDirsToPopulate()
PopulateDirectoryCombo()
CheckForAnyElementOfPathChecked()
CheckForExactElementChecked()
ConvertFileSizeUnits()
ChangeDirectory()
RemoveChildrenElements()
NumNonEmptyElementsInArray()
SlideUpArray()
KeyInitializeArray()
KeyPathFromFullPath()
FileNameFromFullPath()
GetDirectories()
GetFiles()
RecursiveAddDirectoryToArray()
m_lvFileList_ItemCheck()
AddFileToArray()
RecoverMemoryArray()
SelectAll()
m_lvFileList_ItemActivate()
m_cmbDirectoryPicker_SelectedIndexChanged()
FileOKToTransfer()

CompactFTPFileList

FileArray
StrFileListing
m_FTPConnection
MemoryForOutsideObjects
PopulateFiles

InitializeComponent()
PopulateFiles()
GetArrayOfDirsToPopulate()
GetArrayOfFilesToPopulate()
ChangeDirectory()
GetDirectories()
GetFiles()
FileOKToTransfer()

CompactPPCFileList

MemoryForOutsideObjects

InitializeComponent()
PopulateFiles()
GetArrayOfDirsToPopulate()
GetArrayOfFilesToPopulate()
ChangeDirectory()
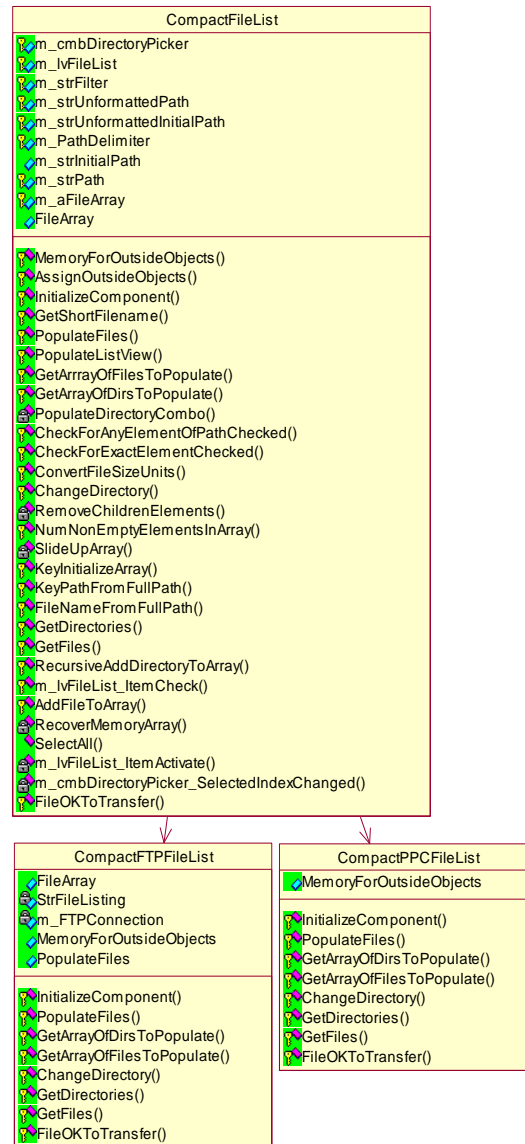GetDirectories()
GetFiles()
FileOKToTransfer()

Figure 10: File list classes

Implementation

The major part of the implementation was to design visual interfaces to create user profiles. This was accomplished by C# modules for user authorization and authentication, user profiling, file browsing, password encryption, and the FTP client. Some screenshots are given in Figure 11.
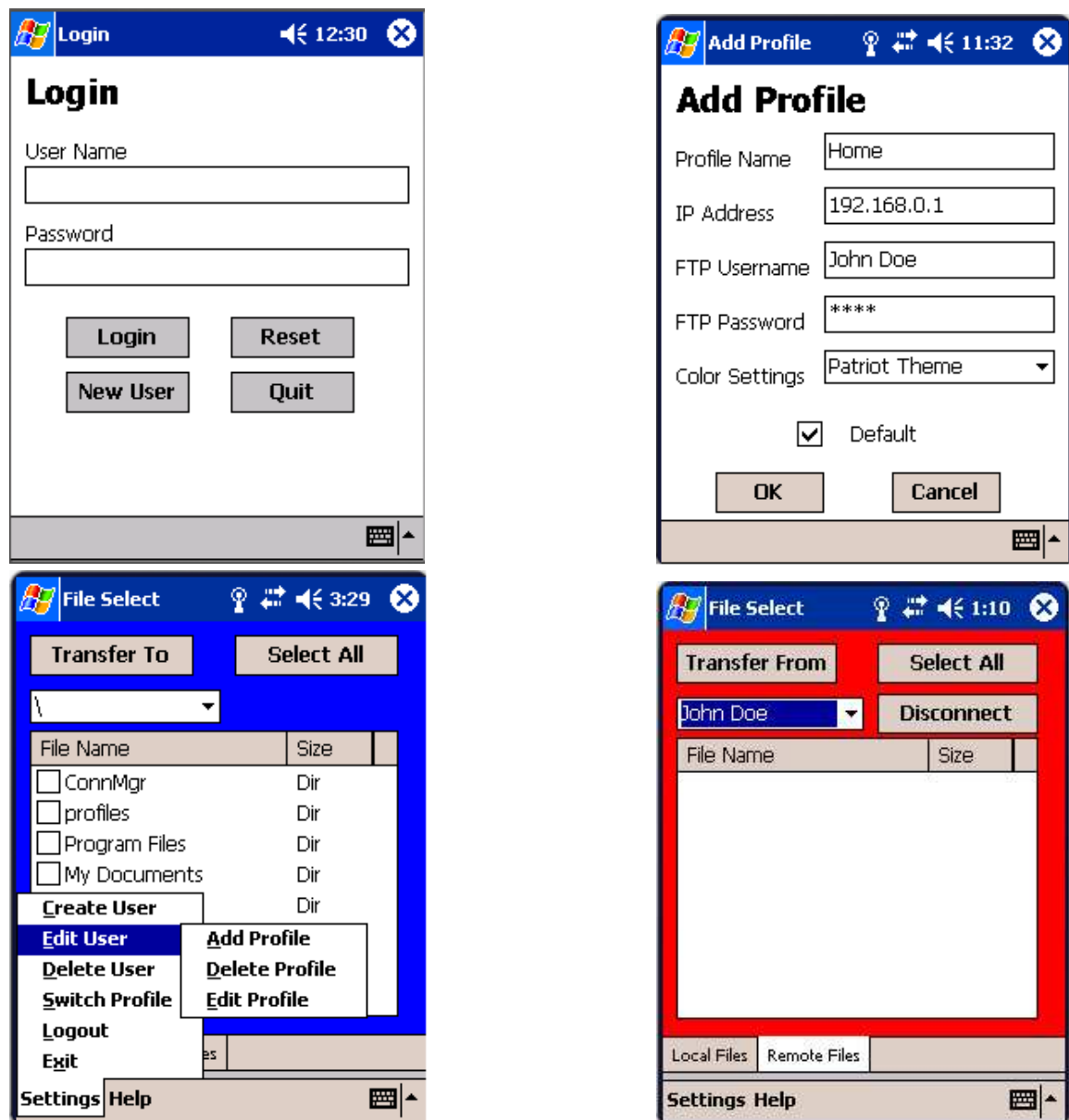
Figure 11: File Manager screenshots: user authentication (upper left), user profiling (upper right), menu choices (lower left), and remote connection (lower right)

Concluding remarks

This paper outlines the idea of a flexible and efficient tool for managing file storage on mobile devices enabling file transfer through various networking connections including wireless. An application named File Manager has been developed with a user-friendly interface and some comprehensive choices for user profiling and FTP connection. Both corporate and home use environments can benefit from integrating File Manager in order to synchronize file structure, move and/or update files. The application can be adopted to other networking and file management solutions.