Case Study: MyHome – Real Estate System

1.1 INSTRUCTIONS

- Create a directory by your name in drive "**D:**" Under that create a subdirectory called **MiniProject** and Store your Project there.
- Use C or C++. You can refer to your course material.
- The code modules in the mini project should follow all the coding standards.
- You may also look up the help provided in the MSDN Library.

Number of participants for this project = 6 to 8

PROBLEM STATEMENT

1.2 OBJECTIVE

To create a real estate MyHome application like Magicbricks which will assist user to home buying or renting decision. The application is to be develope as Desktop Application. There are 2 entities Admin and User.

1.3 ABSTRACT OF THE PROJECT

1. Admin and user should be able to login into the application.
2. User should be able to search for the estate.
3. Provide all advance filters needed for the search.
4. Show at least 5 images of the property in property details.
5. User should be able to download brochure about the estate.
6. Admin should be able to see all the estates user visited.
7. If user shows interest (like downloading brochure) in property then admin should be notified.
8. Add various statistics so admin can view and make business strategy.
9. Handle data and errors properly. Show appropriate messages to user.
10. Display good input, output messages and reports in proper format.
11. Security features should be implemented whereever possible. For example user passwords can be stored in encrytped format.

1.4 FUNCTIONAL COMPONENTS OF THE PROJECT

Following is a list of functionalities of the system. Wherever, the description of functionality is not adequate; you can make appropriate assumptions and proceed.

1. Data files

Create initial data in these files as comma separated fields. Each line stores one record.

- Maintain information about properties in "properties.txt" file. Property is identified by Property ID which is auto increment number.
- Photos of property along with filename where photo is present, property ID is stored in "photos.txt" file.
- Brochures along with filename where brochure is present, property ID is stored in "brochures.txt" file.

- User information is stored in “users.txt” file. User is identified by username which is given by user at the time of registeration.

In addition to above master files following transaction files should be maintained by MyHome.

- Maintain “visited.txt” file containing username, Property ID, Date when visited whenever property details are viewed by user.
- Maintain “message.txt” file to store messages for admin. For example – when broucher is downloaded or photos are browsed by users.

2. When MyHome starts it displays Following Screen -

----------Login Screen----------
1. Login as admin
2. Login as user
3. Register new user
0. Quit

Enter your option = <option>

option = 1 (Login as admin)
By default admin/admin is username/password of administrator. When this option is selected “Admin menu” will be displayed.

option = 2 (Login as user)
username and password will be asked and verified with “users.txt” file. When this option is selected “Property menu” will be displayed.

option = 3 (Register new user)
All information about user will be asked and stored in “users.txt” file.

3. When admin logs in MyHome displays “Admin Menu”

---------- Admin Menu----------
1. Add a property
2. Add owner details
3. Add brochure
4. Add photos
5. View visited users
6. View interested users
0. Quit
Enter your option : <option>

option = 1 (Add a property)

Property details will be asked and stored in file “properties.txt”. Property is identified by Property ID which is auto increment number.

option = 2 (Add owner details)
Property owner details will be asked and stored in file “owners.txt”. Property ID + Owner ID will uniquely identify a record in this file. There can be multiple owners for a property. Within property Owner ID is auto increment number.

option = 3 (Add brochure)

Brochure details will be entered by admin and stored in “brochures.txt” file. Property ID + Brochure ID will uniquely identify a record in this file. There can be multiple brochure for a property. Within property Brochure ID is auto increment number.

option = 4 (Add photos)

Photos details will be entered by admin and stored in “photos.txt” file. Property ID + Photo ID will uniquely identify a record in this file. There should be at least 5 photos for a property. Within property Photo ID is auto increment number.

option = 5 (View visited users)

Admin will enter Property ID, this option will display number of users who browsed that property listing along with usernames from “visited.txt” file.

option = 6 (View interested users)

Admin will enter Property ID, this option will display number of users who browsed that property listing along with usernames from “message.txt” file.

3. When user logs in MyHome displays “Property Menu”

---------- Property Menu----------

1. List properties by area
2. List properties by pincode
3. List properties to sell within given price range

<provide at least 10 filters on property attributes>

11. Download Brochure
12. View Photos
0. Quit

Enter your option : <option>

Option 1 to 10 should be implemented as filters on data present in all txt files maintained by MyHome

option = 11 (Download Brochure)
Ask for Property ID from user and display brochure using information in

"brochures.txt" file. You should provide option to download brochure.

option = 12 (View Photos)
Ask for Property ID from user and display photos using information in "brochures.txt" file. You should provide option to browse photos back and forth.

Assumptions:

Note:

If you are using C

1. Use Linked Lists to read data from corresponding text files at the beginning of program.
2. Updates on data during program execution should be done in these Linked Lists.
3. When user quits the program all Linked Lists data to be writting in corresponding text files so that updated data is available in next program execution.

If you are using C++ Arrays of objects can be used instead of Linked List.

Set Up Checklist for Mini Project

Software Requirement:

Visual studio 6.0 and above (To write code using C or C++).

Minimum System / Hardware Requirements:

Intel Pentium 90 or higher (P166 recommended)
Microsoft Windows 95, 98, or NT 4.0, 2k, XP, Access to UNIX server through Telnet
Memory: 32MB of RAM (64MB or more recommended)