/// <summary>

/// Checks for the "data" directory and creates one if there isn't.

/// Checks if each file exists. Creates new files for the missing files.

/// </summary>

private void CheckForMissingFiles()

{

List<string> missingFiles = new List<string>();

string outputMessage = "";

if (!Directory.Exists(Application.StartupPath + "\\data"))

Directory.CreateDirectory(Application.StartupPath + "\\data");

string filePath = Application.StartupPath + "\\data\\";

if (!File.Exists(filePath + "books.txt"))

{

missingFiles.Add("books");

File.Create(filePath + "books.txt");

}

if (!File.Exists(filePath + "bookCopies.txt"))

{

missingFiles.Add("bookCopies");

File.Create(filePath + "bookCopies.txt");

}

if (!File.Exists(filePath + "titles.txt"))

{

missingFiles.Add("titles");

File.Create(filePath + "titles.txt");

}

if (!File.Exists(filePath + "mediaTypes.txt"))

{

missingFiles.Add("mediaTypes");

File.Create(filePath + "mediaTypes.txt");

}

if (!File.Exists(filePath + "authors.txt"))

{

missingFiles.Add("authors");

File.Create(filePath + "authors.txt");

}

if (!File.Exists(filePath + "publishers.txt"))

{

missingFiles.Add("publishers");

File.Create(filePath + "publishers.txt");

}

if (!File.Exists(filePath + "themes.txt"))

{

missingFiles.Add("themes");

File.Create(filePath + "themes.txt");

}

if (!File.Exists(filePath + "bookThemes.txt"))

{

missingFiles.Add("bookThemes");

File.Create(filePath + "bookThemes.txt");

}

if (!File.Exists(filePath + "genres.txt"))

{

missingFiles.Add("genres");

File.Create(filePath + "genres.txt");

}

if (!File.Exists(filePath + "bookGenres.txt"))

{

missingFiles.Add("bookGenres");

File.Create(filePath + "bookGenres.txt");

}

if (!File.Exists(filePath + "members.txt"))

{

missingFiles.Add("members");

File.Create(filePath + "members.txt");

}

if (!File.Exists(filePath + "associatedMembers.txt"))

{

missingFiles.Add("associatedMembers");

File.Create(filePath + "associatedMembers.txt");

}

if (!File.Exists(filePath + "transaction.txt"))

{

missingFiles.Add("transaction");

File.Create(filePath + "transaction.txt");

}

if (!File.Exists(filePath + "transactionBooks.txt"))

{

missingFiles.Add("transactionBooks");

File.Create(filePath + "transactionBooks.txt");

}

if (!File.Exists(filePath + "staff.txt"))

{

missingFiles.Add("staff");

File.Create(filePath + "staff.txt");

}

if (!File.Exists(filePath + "settings.txt"))

{

missingFiles.Add("settings");

File.Create(filePath + "settings.txt");

}

if (missingFiles.Count > 0)

{

foreach (var missingFile in missingFiles)

outputMessage += missingFile + ", ";

outputMessage = outputMessage.Remove(outputMessage.Length - 3, 2);

MessageBox.Show("Missing files detected. Replacing with new files: " + outputMessage);

}

}

if (membersFileContents.Length % 14 == 0)

{

int item = 0;

try

{

for (; item < membersFileContents.Length / 14; item += 14)

{

member tempMemberRecord = new member();

int offset = 0;

tempMemberRecord.Barcode = membersFileContents[item + offset++];

tempMemberRecord.FirstName = membersFileContents[item + offset++];

tempMemberRecord.LastName = membersFileContents[item + offset++];

tempMemberRecord.DateOfBirth = membersFileContents[item + offset++];

tempMemberRecord.CustomerType = (customerType)Convert.ToInt32(membersFileContents[item + offset++]);

tempMemberRecord.EmailAddress = membersFileContents[item + offset++];

tempMemberRecord.PhoneNumber = membersFileContents[item + offset++];

tempMemberRecord.AddressLine1 = membersFileContents[item + offset++];

tempMemberRecord.AddressLine2 = membersFileContents[item + offset++];

tempMemberRecord.AddressLine3 = membersFileContents[item + offset++];

tempMemberRecord.AddressLine4 = membersFileContents[item + offset++];

tempMemberRecord.AddressLine5 = membersFileContents[item + offset++];

tempMemberRecord.Postcode = membersFileContents[item + offset++];

tempMemberRecord.JoinDate = membersFileContents[item + offset++];

members.Add(tempMemberRecord);

}

}

catch

{

MessageBox.Show("Failed to load member file: invalid data at record " + (item / 14));

}

}

else

{

MessageBox.Show("Failed to load member file: missing data");

}

/// <summary>

/// Outputs a messagebox declaring the missing data item and it's ID

/// </summary>

/// <param name="item">missing data item</param>

/// <param name="ID">ID of the missing data item</param>

private static void MissingItemID(string item, int ID)

{

item = item[0].ToString().ToUpper() + item.Substring(1);

MessageBox.Show($"{item} not found. ID: {ID}");

}