Code Conventions

1. **Use camelCasing** for method arguments, local variables and fields.

2. **Use Screaming Snake Case** for constants.

   - Ex: `public const int MESSAGE_MAXIMUM_SIZE = 4112;`

3. **Use PascalCasing** for everything that is not in the first two rules.

4. **Use meaningful names** for every component of a class and its local variables.

   - Ex:

```
String endPointNullResult = "Not Connected";
```

```
17 references
public class TwoFactorAuthenticationService : ITwoFactorAuthenticationService
```

```
public async partial void ConnectUserToServer()
```

5. Avoid using abbreviations in variable names.

   - Ex don't: `private User currUsr;`

   - Ex do: `private User currentUser;`

   Exceptions: abbreviations commonly used as names
   -Ex exceptions: `private int userId;` (others: uri, xml, ftp)

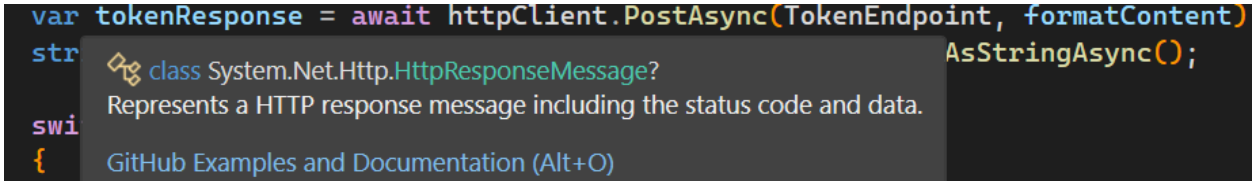
6. Class fields will not be prefixed with underscores regardless of their access level.

   - Ex do: `private Socket serverSocket;`

- Ex don't: `private Socket _serverSocket;`

7. **Don't** use implicit type **var**.

   Helper: If you don't know the type of a variable, write it as var and then hover over the type.



   Afterwards replace it with the real one: `HttpResponseMessage tokenResponse`

8. **Use vertically aligned** curly brackets.

   - Ex:

```
switch (user)
{
    case null:
        // Don't know why email is used as username but let's vibe with it
        User newUser = new User { UserId = userId, Username = email, PasswordHash = string.Empty, TwoFASecret = null };
        bool wasCreated = UserDatabaseAdapter.CreateUser(newUser);
        break;
    case not null:
        break;
}
```

9. Any class member **must** be prefixed by "**this**".

   - Ex: `String ipAddress = this.socketsAndAddresses.GetValueOrDefault(clientSocket) ?? "";`

10. Comments will focus on explaining **why** something is done, instead of simply writing **what** the code does. *Jokes are welcomed.*

    - Ex don't: This chat gpt code :<

```python
def check_winner(board, player):
    # Check rows, columns, and diagonals
    for row in board:
        if all(cell == player for cell in row):
            return True
    for col in range(3):
        if all(board[row][col] == player for row in range(3)):
            return True
    if all(board[i][i] == player for i in range(3)) or \
       all(board[i][2 - i] == player for i in range(3)):
        return True
    return False
```

11. Interfaces are **prefixed** with "I".

    - Ex: `internal interface IService`

12. Use **consistent indentation**.

    - Ex don't:

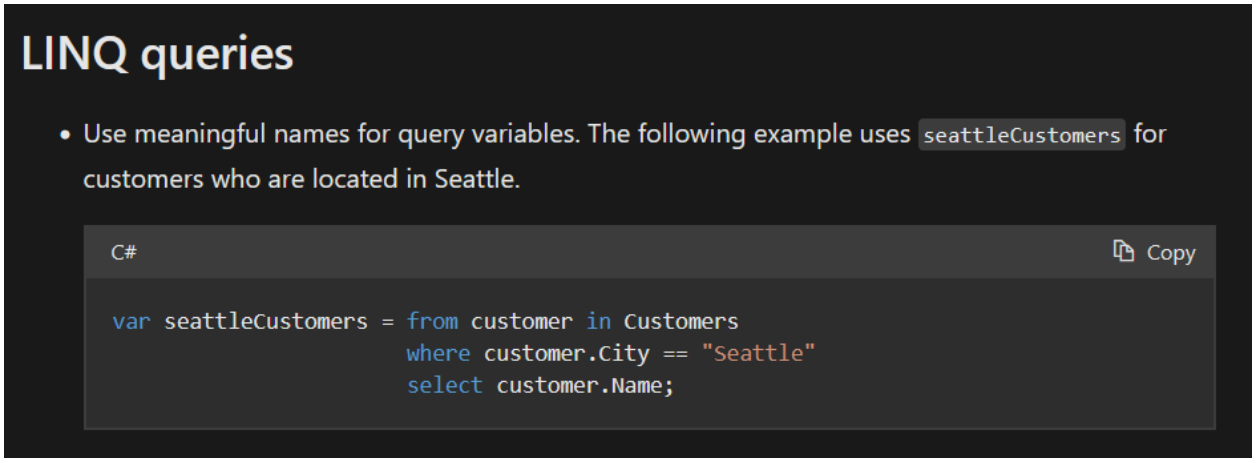
```
public static Guid CreateGloballyUniqueIdentifier(string identifier)
{
        using (MD5 cryptographicHasher = MD5.Create())
        {
                byte[] hashResult = cryptographicHasher.ComputeHash(Encoding.UTF8.GetBytes(identifier));
            return new Guid(hashResult);
              }
    }
```

13. Class member variables and properties **go** at the **top**.

    - Ex:

```
private Client? client;
private DispatcherQueue uiThread;
private Server? server;

public event EventHandler<MessageEventArgs> NewMessageEvent;
public event EventHandler<ClientStatusEventArgs> ClientStatusChangedEvent;
public event EventHandler<ExceptionEventArgs> ExceptionEvent;

private String userName;
private String userIpAddress;
private String serverInviteIp;

public const String HOST_IP_FINDER = "None";
public const String GET_IP_REPLACER = "NULL";

/// <summary>
/// Creates the handler for any operations or for errors encountered
/// </summary>
/// <param name="userName">The name of the user who joined the chat room</param>
/// <param name="serverInviteIp">The ip address of the user who sent the invite,
///                              will be HOST_IP_FINDER if the user is the host</param>
/// <param name="uiThread">Updating the ui can be done using only the main thread</param>
public Service(String userName, String serverInviteIp, DispatcherQueue uiThread)
{
    this.userName = userName;
    this.userIpAddress = Service.GetIpAddressOfCurrentUser();
    this.serverInviteIp = serverInviteIp;
    this.uiThread = uiThread;
}
```

14. For **LINQ** with example from Microsoft Doc:



Remark: Listen to rule 7) about "var"

15. **Enums** will have **singular** nouns and will not contain the "Enum" suffix

- Ex:

```
public enum Coin
{
  Penny,
  Nickel,
  Dime,
  Quarter,
  Dollar
}
```

Exceptions: Bit fields enums can have plural nouns.

- Ex:

```
[Flags]
public enum Dockings
{
  None = 0,
  Top = 1,
  Right = 2,
  Bottom = 4,
  Left = 8
}
```

16. **No summaries** (overriding the stylecop rules).

17. **No magic numbers**. This means that no variable (could be a number, could be a string) should directly be passed into functions or simple checks.

    Helper: Consider between local variables, class constants or enum.

    - Ex:

```
float playerRadius = 0.3f;
bool canMove = !Physics.CapsuleCast(transform.position, transform.position + Vector3.up * playerHeight, playerRadius, directionVector, moveDistance);
```

```
public enum Scene
{
    MainMenu,
    LoadingScene,
    FirstFloor,
    CastleEntrance
}
```

```
private const string IS_ATTACKING = "IsAttacking";
```