

Technical Document: BasicTies Platform

By Daniel Knight and Keegan Preston

Introduction

This document offers a comprehensive overview of the main classes and components pivotal to the BasicTies platform. Each section details a specific class or feature, highlighting its properties, methods, and other essential attributes.

Namespace & Directives

Namespace:

- BasicSocial: This namespace wraps all related classes and functions of the "BasicTies" platform.

Using Directives:

- System: Base .NET namespace.
- System.Collections.Generic: Interfaces and classes defining generic collections.
- System.Data: Access to and management of data from different data sources.
- System.Data.SqlClient: ADO.NET provider for SQL Server.
- System.Web: Enhances browser/server communication.
- BasicSocial.Backend: Contains the backend logic for the BasicSocial platform.

1. Member Class

Properties:

- Email: Member's email with validation.
- Password: Member's password with validation.
- FirstName & LastName: Member's names with validation.
- Birthday: Member's birthday (validation ensures age > 13).
- MembershipId: Unique ID with validation.
- AccountBalance: Account balance with validation.

- IsAdmin: Boolean flag indicating admin status, with validation.
- Feedback: Read-only property for feedback messages.
- MemberSince: Static read-only property displaying current date-time.

Methods:

- SearchMembers_DS(string name):
 - **Parameters:**
 - name: Name of the member to be searched.
 - **Return:** Dataset with member details based on the provided name.
 - **Description:** Allows users to search for members by name.
- AddMember():
 - **Return:** Boolean indicating the successful addition of a member.
 - **Description:** Introduces a new member to the database.
- FindMember(int ID):
 - **Parameters:**
 - ID: Member's unique ID.
 - **Return:** Dataset with member details using the provided ID.
 - **Description:** Searches for a member by their unique membership ID.
- DeleteMember(int ID):
 - **Parameters:**
 - ID: Member's unique ID to be deleted.
 - **Return:** Boolean indicating the successful deletion of a member.
 - **Description:** Removes a member from the database using their unique ID.
- UpdateMember():
 - **Return:** Boolean indicating the successful update of member data.
 - **Description:** Modifies member data in the database.

2. Post Class

Properties:

- _postId: Unique ID for the post.
- _posterFrom: ID of the member posting.
- _postMessage: Content/message of the post.
- Feedback: Storage for feedback/error messages related to posts.

Methods:

- **GetPosts_DS():**
 - **Return:** Dataset containing posts.
 - **Description:** Retrieves all posts in a dataset format.
- **AddPost(string content):**
 - **Parameters:**
 - content: The content of the post.
 - **Return:** Boolean indicating the successful addition of a post.
 - **Description:** Adds a new post and if needed, manages replies.
- **DeletePost(int ID):**
 - **Parameters:**
 - ID: Unique post ID.
 - **Return:** Boolean indicating the successful deletion of a post.
 - **Description:** Removes a post using its unique ID.

3. Validation Class

Methods:

- **GetConnected():**
 - **Return:** Connection string containing server, database name, user ID, and password specifics.
 - **Description:** Supplies the necessary information to create a connection to the database.
- **IntOnly(string stringInput):**
 - **Parameters:**
 - stringInput: String to be verified for integer values.
 - **Return:** Boolean indicating whether the string is an integer.
 - **Description:** Confirms if the string is an integer.
- **DoubleOnly(string stringInput):**
 - **Parameters:**
 - stringInput: String to be verified for double values.
 - **Return:** Boolean indicating whether the string is a double.
 - **Description:** Confirms if the string is a double.
- **IsItFilledIn(string temp):**
 - **Parameters:**
 - temp: String to be verified for content.
 - **Return:** Boolean indicating if the string is non-blank.
 - **Description:** Validates that the string is not empty.

- **IsValidMailAddress**(string email):

- **Parameters:**

- email: Email string to be verified.

- **Return:** Boolean indicating if the email format is valid.

- **Description:** Validates the email address using .NET's MailAddress class.

- **IsValidEmail**(string stringInput):

- **Parameters:**

- stringInput: Email string to be verified.

- **Return:** Boolean indicating the authenticity of the email format.

- **Description:** Validates the email format via a regex pattern.

- **IsValidPhoneNumberFormat**(string number):

- **Parameters:**

- number: Phone number string to be verified.

- **Return:** Boolean indicating the authenticity of the phone number format.

- **Description:** Validates phone numbers, recognizing multiple formats like "(123) 456-7890", "123-456-7890", "123.456.7890", and "1234567890".

4. MasterFeed Page (BasicSocial Backend)

Overview: Represents the primary interface for viewing, crafting, and managing posts.

Dependencies:

- Uses System namespaces and others as mentioned earlier.
- Leverages System.Web.UI and related namespaces for the user interface.

Properties:

- **ExpandedPosts**

- Type: List<int>

- Description: Gets or sets the list of posts that are expanded.

Methods:

- **Page_Load**

- Parameters:

- sender: The source of the event.

- EventArgs e: Contains the event data.

- Description: Handles the loading of the main feed page.

•	PopulatePostsGrid
	<ul style="list-style-type: none"> Parameters: None Return: Void Description: Populates the data grid with public posts from the database.
•	GetMemberInfo
	<ul style="list-style-type: none"> Parameters: None Return: Void Description: Retrieves and displays the currently logged-in member's basic information.
•	ShowModal
	<ul style="list-style-type: none"> Parameters: <ul style="list-style-type: none"> sender: The source of the event. EventArgs e: Contains the event data. Description: Displays the modal window, determining the action type (Post or Reply) based on which button was clicked.
•	btnPost_Click
	<ul style="list-style-type: none"> Parameters: <ul style="list-style-type: none"> sender: The source of the event. EventArgs e: Contains the event data. Description: Handles the action when a user clicks to post or reply to a post.
•	btnLikePost_Click
	<ul style="list-style-type: none"> Parameters: <ul style="list-style-type: none"> sender: The source of the event. EventArgs eventArgs: Contains the event data. Description: Handles the like functionality for posts.
•	btnDeletePost_Click
	<ul style="list-style-type: none"> Parameters: <ul style="list-style-type: none"> sender: The source of the event. EventArgs e: Contains the event data. Description: Allows a user to delete a post.
•	GetPostClass
	<ul style="list-style-type: none"> Parameters: <ul style="list-style-type: none"> int depthLevel: The depth level of the post. Return: string Description: Returns a CSS class string based on the depth level of the post.
•	dgResults_ItemDataBound
	<ul style="list-style-type: none"> Parameters:

	<ul style="list-style-type: none"> • sender: The source of the event. • DataGridViewEventArgs e: Contains the event data.
	<ul style="list-style-type: none"> • Description: Manages data bound to the posts data grid.
• btnShowReplies_Click	
	<ul style="list-style-type: none"> • Parameters:
	<ul style="list-style-type: none"> • sender: The source of the event. • EventArgs e: Contains the event data.
	<ul style="list-style-type: none"> • Description: Handles the show or hide replies functionality for posts.
• BindPostsGrid	
	<ul style="list-style-type: none"> • Parameters: None • Return: Void • Description: Binds the posts to the data grid.
• FilterPosts	
	<ul style="list-style-type: none"> • Parameters:
	<ul style="list-style-type: none"> • DataTable sourceTable: The source table containing posts.
	<ul style="list-style-type: none"> • Return: DataTable • Description: Filters the posts based on which are expanded.
• IsVisible	
	<ul style="list-style-type: none"> • Parameters:
	<ul style="list-style-type: none"> • DataRow row: The current row being checked. • int postId: The ID of the post.
	<ul style="list-style-type: none"> • Return: bool • Description: Determines if a post should be visible.
• IsTopLevelPost	
	<ul style="list-style-type: none"> • Parameters:
	<ul style="list-style-type: none"> • DataRow row: The current row being checked. • int postId: The ID of the post.
	<ul style="list-style-type: none"> • Return: bool • Description: Determines if a post is a top-level post.
• IsChildOf	
	<ul style="list-style-type: none"> • Parameters:
	<ul style="list-style-type: none"> • DataRow currentRow: The current row being checked. • List<int> expandedPosts: List of posts that are expanded.
	<ul style="list-style-type: none"> • Return: bool • Description: Determines if a post is a child of any expanded post.

Utility and Helper Methods:

- These offer additional functionalities commonly used across the application.

User Interface Components:

- Elements like labels, grids, modals, and buttons for information display and user interactions.

5. MemberSearch Page (BasicSocial Backend)

Overview: Provides a member search feature for the BasicSocial platform.

Dependencies:

- Utilizes the Member class for member searching functionality.

Page Lifecycle Methods:

- Page_Load():
 - **Return:** Void.
 - **Description:** Initializes the page, validates user session, retrieves and displays member search results based on criteria.

Core Methods:

- SearchMembers():
 - **Parameters:** None (Relies on user-entered criteria).
 - **Return:** Void.
 - **Description:** Performs a member search based on the entered criteria and displays results.
- PendingInvites_DS():
 - **Return:** Dataset containing pending friend invites.
 - **Description:** Retrieves pending friend invites for the current user.
- InvitesSent_DS():
 - **Return:** Dataset containing sent friend invites.
 - **Description:** Retrieves friend invites sent by the current user.
- AddFriend_Click():
 - **Parameters:** None.
 - **Return:** Void.
 - **Description:** Sends a friend request to a selected member.

- **AcceptRequest_Click():**
 - **Parameters:**
 - emailFrom: Email of the member who sent the friend request.
 - **Return:** Void.
 - **Description:** Accepts a friend request from a selected member.
- **CancelRequest_Click():**
 - **Parameters:**
 - emailTo: Email of the member to whom the friend request was sent.
 - **Return:** Void.
 - **Description:** Cancels a sent friend request to a selected member.

Utility Methods:

- **HasRequestOrFriendship(SqlConnection conn, string query, string email):**
 - **Parameters:**
 - conn: SqlConnection object for database connection.
 - query: SQL query for checking friend requests or friendships.
 - email: Email of the member being checked.
 - **Return:** Boolean indicating the presence of friend requests or friendships.
 - **Description:** Checks if there are pending friend requests or existing friendships.
- **AcceptFriendRequest(string emailFrom):**
 - **Parameters:**
 - emailFrom: Email of the member who sent the friend request.
 - **Return:** Void.
 - **Description:** Accepts a friend request and establishes a friendship.
- **DeleteFriendRequest(string emailTo):**
 - **Parameters:**
 - emailTo: Email of the member to whom the friend request was sent.
 - **Return:** Void.
 - **Description:** Deletes a friend request.

6. MembershipPanel Page (BasicSocial Backend)

Overview: The MembershipPanel class serves as a user interface handler for membership-related functionalities. It allows for viewing and editing user profile details, managing posts, and more.

Dependencies:

- Utilizes the Member class for member searching functionality.

Page Lifecycle Methods:

- **Page_Load():**
 - **Return:** Void.
 - **Description:** Initializes the page, validates user session, retrieves and displays member search results based on criteria.

Core Methods:

- **Page_Load**
 - **Description:** Initializes the page and sets the visibility of various labels and controls based on the session state.
 - **Parameters:**
 - **sender:** The source of the event.
 - **EventArgs e:** Contains the event data.
 - **Returns:** None.
- **PopulateUserInfo**
 - **Description:** Retrieves and populates the user's information from the database.
 - **Parameters:** None.
 - **Returns:** None.
- **SetProfileImage**
 - **Description:** Sets the user's profile image based on the provided email.
 - **Parameters:**
 - **email:** The email of the user.
 - **Returns:** None.
- **GetTotalNumberOfFriends**
 - **Description:** Calculates and displays the total number of friends for the current user.
 - **Parameters:** None.

	<ul style="list-style-type: none"> • Returns: None.
• PopulateFriendsList	<ul style="list-style-type: none"> • Description: Fetches and displays the user's list of friends. • Parameters: None. • Returns: None.
• PopulatePrivateFeed	<ul style="list-style-type: none"> • Description: Retrieves and binds the user's private feed of posts to a data grid. • Parameters: None. • Returns: None.
• EditProfile	<ul style="list-style-type: none"> • Description: Provides an interface and functionality for the user to edit their profile details. • Parameters: None. • Returns: None.
• ChangePassword	<ul style="list-style-type: none"> • Description: Provides an interface for the user to change their password. Validates the old password before allowing the change. • Parameters: <ul style="list-style-type: none"> • oldPassword: The user's current password. • newPassword: The new password the user wishes to set. • Returns: <ul style="list-style-type: none"> • bool: True if the password was changed successfully, False otherwise.
• AddPost	<ul style="list-style-type: none"> • Description: Allows the user to add a new post to their feed. • Parameters: <ul style="list-style-type: none"> • postContent: The content of the post. • Returns: None.
• DeletePost	<ul style="list-style-type: none"> • Description: Allows the user to delete a specific post from their feed. • Parameters: <ul style="list-style-type: none"> • postId: The unique identifier of the post to be deleted. • Returns: <ul style="list-style-type: none"> • bool: True if the post was deleted successfully, False otherwise.