Front-end Class Diagram

Front-end data and operator types

**Interface:**

Operations

* -search(): trigger an html event that calls to SearchBuilder and passes html forum information
* -view(): trigger an html event that calls the Viewer and passes challengeID
* -filter(): calls lister to filter
* DisplayView(page): displays new html page received from viewer
* DisplayList(page): displays a new list that is returned from Lister
* DisplayError(error): displays error message based on type

**Viewer:**

Attributes:

* -Selection: a challenge datatype
* -isChallenge: bool to check if it has an actual challenge

Operations:

* -buildPage(): build html from Challenge info and call interface to display
* -sendError(): error detected when trying to create page
* +recieveChallenge(challenge): receive a challenge data, assign to selection and begin calling either buildPage or sendError

**SearchBuilder**:

Attributes:

* -searchParameters: String
* -validSearch: Bool

Operations

* +buildSearch(): creates search parameters based on passed forum information
* -query(searchParameters): send a search to db connection
* -checkParameters(): check to see if valid parameters or return error.

**Lister**:

Attributes:

* +CurrentList: List datatype
* -isList: bool

Operations:

* -organizeList(): sort and organize list for the optimal viewing
* -FilterLIst(filters): filter List based on filters
* -buildPage(): build html page and call the interface to display
* -sendError(): send error to interface
* +recieveList(list): assing list as currentList and call buildPage() or sendError()

**Challenge**

Attributes:

* +id: int
* +name: string
* +detailedDes string
* +originSite string
* +originLink string

**DBconnection**

Attributes:

* -isConnected bool

Operations:

* -checkConnection(): check to see if there is a response from database
* +listQuery(searchParam): called by searchBuilder query database and then calls resulthandler
* +viewQuery(selection): called by viewer returns data to resulthandler

**List**

Attributes:

* +ids: int array
* +names: string array
* +shrotDes: string array
* +orignSites: string array

**resultHandler**

Attributes:

* isData : bool
* error: int

Operations:

* +formatList(ListData): turns database data into List data structure
* +formatChallenge(challengeData): turns database data into Challenge data structure
* -sendList(list,error): send a List to listbuilder and include any errors that occurred
* -sendChallenge(challenge,error): send a Challenge to viewer and include any errors that occurred