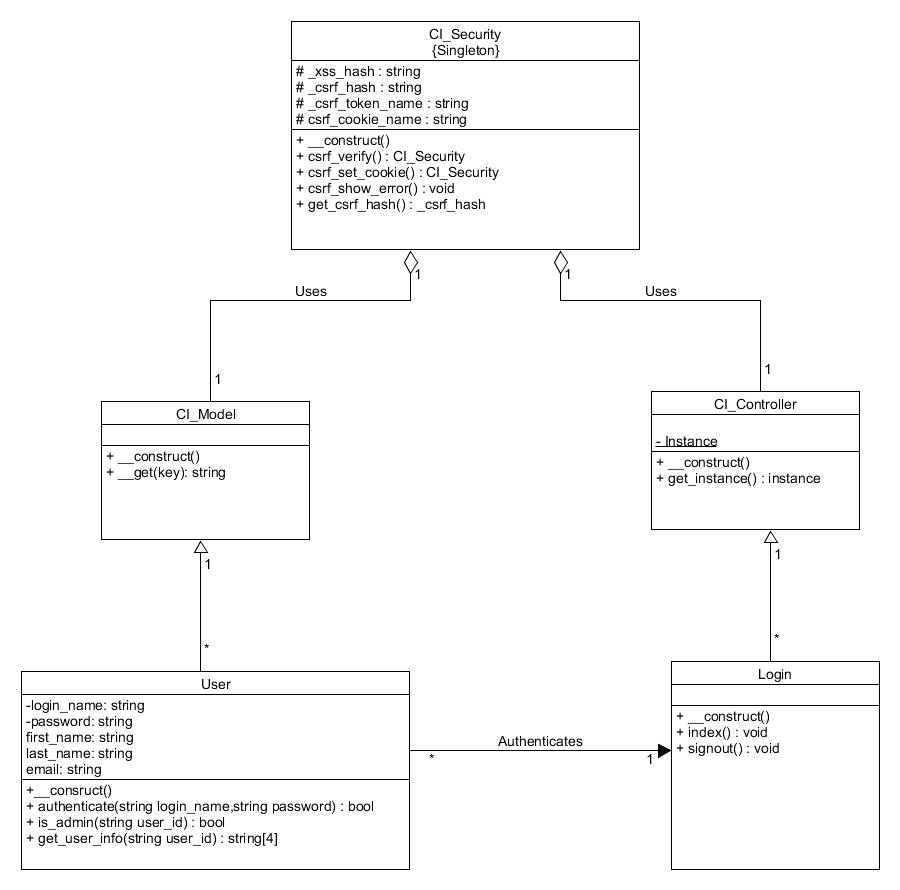
AUTHENTICATION SUBSYTEM



4.1

The authentication subsystem is responsible for validating user logins as well as providing security throughout the entire session and hashing sensitive information such as passwords. The user class is the starting point of the authentication subsystem, it is found within the models therefore it must inherit from the template class CI\_Model. The user begins by entering their username and password in order to login to MyTinerary. This in turn calls the authenticate method which will take in the entered login name and password and verify them by querying the database. If the query results in one row and the internal password verify method returns true than the login is successful. During this time the User class will interact with the Login Controller class. If the authenticate method is successful the Login class will redirect the user to the index page using the index method as well as providing the user with a new session. The CI\_Security is a singleton class that is the cornerstone of MyTinerary’s secure session handling. It creates a unique hashed Cross Site Request Forgery protection cookie for every user session. MyTinerary also uses the BlowFish algorithm within the database to hash all sensitive data such as passwords or course details.

**4.2**

The following two classes are provided by the CodeIgniter framework that allow for a separation of concerns via the MVC pattern which promotes high cohesion and low coupling by allowing each module to work independently of one another. Please note these classes do not need to be defined but simply inherited from.

**CI\_Model:** Is a template class provided by CodeIgniter in which any class containing the domain logic will inherit from.

Attributes:

* N/A

Methods:

* \_\_construct(): Class constructor initializing a default log message.
* \_\_get(key): Debugging method that returns an error message.

**CI\_Controller:** Is a template class provided by CodeIgniter in which any class that manipulates the user interface classes will inherit from.

Attributes:

* N/A

Methods:

* \_\_constructor(): Class constructor loading respective view classes.
* get\_instance(): Returns controller instance.

**Login:** Is a controller class that redirects the user according the result of the User’s authentication method.

Attributes:

* N/A

Methods:

* \_\_construct(): Class constructor
* index(): Validates if authentication method has succeeded, sets session cookies, and redirects user to home page. Returns void.
* signout(): Destroys current user session and redirects to default login page. Returns void.

**User:** Is a model class that directly interacts with the database containing all user information.

Attributes:

* login\_name: string
* password: string
* first\_name: string
* last\_name: string
* email: string

Methods:

* \_\_construct(): Class constructor calling the parent constructor.
* authenticate(login\_name,password): This method takes in a two strings being the login name and password and queries the database to find a match. Note the strings are immediately hashed using the blowfish algorithm to match with the database elements. Returns a boolean.
* is\_admin(user\_id): Checks if user is an administrator returning a Boolean used to grant different access rights compared to a regular student.
* get\_user\_info(user\_id): Returns user information from the database such as login name and email. Returns user information in the form of an array of strings.

**CI\_Security:** Is a singleton class provided by CodeIgniter in which every component of the MVC classes will use. The CI\_Security class contains many attributes and methods however only a select few are relevant to the authentication subsystem. The CI\_Security provides protection for sensitive information as well as secure session handling.

Attributes:

* \_xss\_hash: Randomly generated hash for protecting MyTinerary’s URLs.
* \_csrf\_hash: Randomly generated hash for Cross Site Request Forgery protection cookie.
* \_csrf\_token\_name: Token name for Cross Site Request Forgery protection cookie.
* \_csrf\_cookie\_name: Cookie name for Cross Site Request Forgery protection cookie.

The xss and csrf hash allows for the protection of user sessions since the hash is randomly generated and provides secure access using either the token or cookie name. Please note that the usage of this class is all done internally within the CodeIgniter framework.

Methods:

* \_\_construct(): Class constructor that will create the cookies and set the hash
* csrf\_verify(): Provides security validation on POST requests, validates if URI has been whitelisted, as well as garbage collection on finished session arrays. Returns the session if verified otherwise returns null.
* csrf\_set\_cookie(): Modifies the internal cookie and returns it. Returns void.
* csrf\_show\_error(): Displays an error message when user of application attempts an action that is either not permitted or not secure. Returns void.
* get\_csrf\_hash(): Return the hash as a string.