Per quanto riguarda i caratteri speciali, se presenti/richiesti in input, sono considerati pericolosi (innescano diverse vulnerabilità, Cfr. [paragrafo 6.1.1]) poichè la loro combinazione non può essere considerata semplice 'testo'.
Di seguito qualche esempio di possibile combinazione:

Caratteri pericolosi	Possibile utilizzazione
< >	identificano tag HTML
! | & ;	esecuzione comandi
' " * %	database queries
? $ @	programmi e script
( ) [ ]	programmi e script
..\./	filesystem paths

Inoltre, caratteri speciali quali:

; | ! ~ ' " - * % ` \ / < > ? $ @ : ( ) [ ] { } .

devono essere identificati e neutralizzati (input sanitizing) con tecniche specifiche quali l'escaping (i caratteri pericolosi devono essere sempre convertiti prima del salvataggio), di seguito alcuni esempi di sostituzione:

Carattere pericoloso	Sostituito con
<	&lt;
>	&gt;
#	&#35;
&	&#38;
(	&#40;
)	&#41;

Il controllo, quindi, deve sempre verificare che non siano inseriti script potenzialmente dannosi. È importante sottolineare che la convalida dell'input utente non deve mai essere svolta lato client, ma sempre dal back-end, sul server, poiché sul client i dati sono sempre visibili e modificabili.

5.2.8 Gestione dell'output

L'applicazione deve fornire in output solamente le informazioni pertinenti e conformi alle richieste avanzate dagli utenti, al fine di evitare qualsiasi raccolta d'informazioni (information gathering) o rivelazione di dati (disclosure) non autorizzate.

5.3 Formattazione del codice

La formattazione del codice e la sintassi devono seguire le seguenti direttive standard:

- Ogni file deve contenere un'intestazione (file header) in cui si specificano l'autore del codice, la data di creazione dello stesso e la storia degli aggiornamenti successivi (se presenti);
- Ogni file header deve contenere la dichiarazione di una ed una sola classe;
- Le dichiarazioni correlate ad una classe riportata all'interno di un file, devono essere poste all'interno dello stesso file;
- Le righe di codice devono avere un numero di caratteri uguale o inferiore a quello previsto dal formato ISO/ANSI per la descrizione delle dimensioni dello schermo (80 caratteri x 24 righe).

5.3.1 Stile e sintassi

Alla dichiarazione di ogni funzione, metodo o classe deve sempre precedere un commento che riporti: