7.3 Sportsbook - Online & Retail

7.3.1 Pro•Line Stadium Front-End

Changes to the Pro-Line Stadium front-end will see SDI take ownership of displaying the header and footer, a responsibility currently residing with OpenBet.

7.3.1.1 Overview

The four steps of the bet placement workflow can be broken down as follows:

1. Games List
2. Bet Slip (aka: Bet Card)
3. Wager Confirmation
4. Bet Receipt

The current setup has a complex iframe setup for step 1, with OpenBet hosting the header, footer and games list, and SDI content surrounding the games list iframe. Steps 2 - 4 are entirely OpenBet hosted, with the header and footer being played out using the template player (no iframes are used).

The proposed change will host all four steps of the bet placement workflow within a single iframe. OpenBet will be required to remove the header and footer from the templates, and provide a JavaScript hook which can be used to inform the parent frame of any events (eg: movement to another step, or a customer successfully placing a bet). This is so SG can have OpenBet trigger actions such as dynamic resizing of the iframe or updating the account balance in the SDI hosted header.

7.3.1.2 Technical Specification

This section assumes integration with the SG PAM's session cookie to confirm the session information.

7.3.1.2.1 Header / Footer removal

Modify `header.html` and `footer.html` in the ALC portal to omit playing the Sportsbook header and footer templates.

7.3.1.2.2 Parent Frame Hook

Since the player will be navigating through the four steps within a single iframe on a static page, develop a lightweight JavaScript function which will call a function in the parent frame to inform SDI which step of the workflow the iframe is in. It will be called both when a customer initiates a jump to another step (eg: on clicking "Wager Now" on step 3) and also on having loaded a step (eg: arriving on Step 4). This is to provide SDI with maximum flexibility in deciding when to resize the iframe. It is optional for SDI to act on any of these events.

The name of the parent frame's function will be configurable. The OpenBet JavaScript function will check for the existence of the parent frame's function, and will avoid making the call if it doesn't exist. This is to avoid an JavaScript errors when testing bet placement outside of the iframe.

The data sent to the parent frame's JavaScript will be **a text string**, formatted as JSON, and will indicate the step the customer is on, as well as the step the customer is navigating to. In the case of final step 4, it will also include an element `wager_success` to indicate if the wager was placed successfully or not. SGI will refresh the account balance in their header if the wager was successful.

Element Name	Manditory	Possible Values
current_step	Yes	integer
destination_step	No (present if about to leave the current step and navigate to another. Not present when arriving on a step)	integer
wager_success	No (arrival on step 4 only)	boolean (0 = failure, 1 = success)

Example 1: Player just clicked "Wager Now" on Step 2 to go to Step 3

```
{
	"current_step": 2,
	"destination_step": 3
}
```

Example 2: Player arrives on Step 3

```
{
	"current_step": 3
}
```

Example 3: Player successfully places wager and arrives on Step 4

```
{
	"current_step": 4,
	"wager_success": 1
}
```

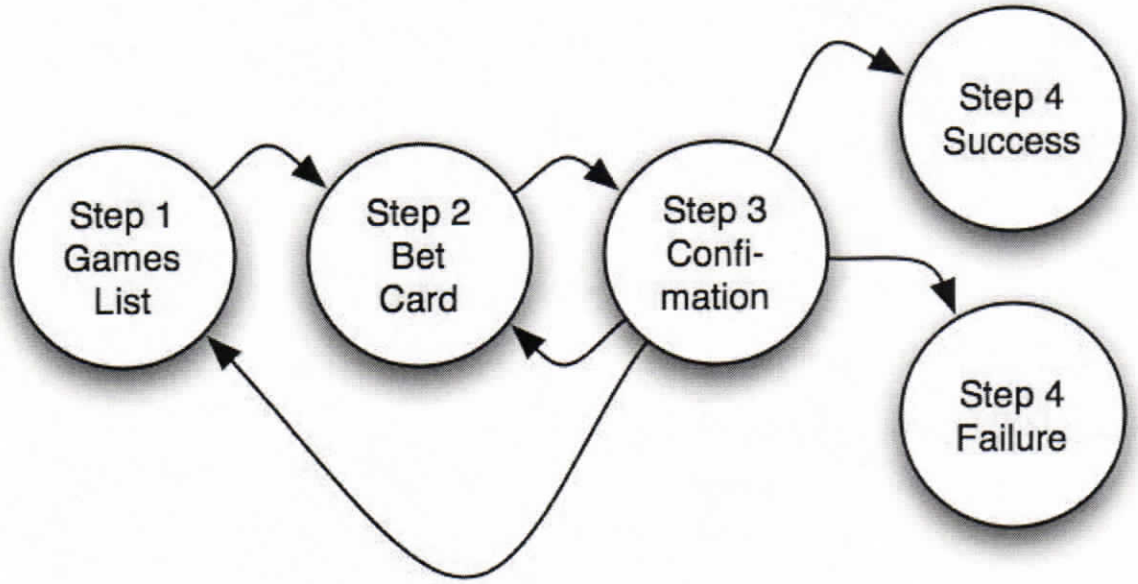
Example 4: Player fails to place wager and arrives on Step 4

```
{
	"current_step": 4,
	"wager_success": 0
}
```

Example 5: Player just clicked "Edit Selections" on Step 3 to navigate back to Step 2

```
{
	"current_step": 3,
	"destination_step": 2
}
```

The following finite state diagram described the possible movement through the bet placement workflow. Note how it is possible to return from step 3 back to step 2 *(edit selections)* and from step 3 back to step 1 *(cancel).*



7.3.2 Web Code Generation

This section details requirement 4.1.15, covered by section 1.5.1 of the BDD document.

> *Using an ALC provided algorithm, the sports system will generate a webcode for each retail wager request and include it in wager response message.*

7.3.2.1 Overview

ALC have requested the addition of a webcode on each printed betslip. The webcode will be generated by OpenBet and stored in the OpenBet database. The webcode can be redeemed via Sciplay.

7.3.2.2 Technical Specification

In order to preserve the XML schema for the ALC API, the webcode will be added to the existing `<barcode />` node in the response to `respWager`. The multiple items will be semicolon separated.

For example, the current appearance of `<barcode />` is:

```
<barcode>70266467430050965</barcode>
```

After development, it will appear as follows:

```
<barcode>70266467430050965;AABBBBBBBBBCDDEEEFFFFGGGG</barcode>
```

The developer may build upon the existing prototype in the OpenBet CVS repository.

7.3.2.2.1 Database

Create a new table, `tWebCode`, which will store the web codes. They will associate with `tSlip`