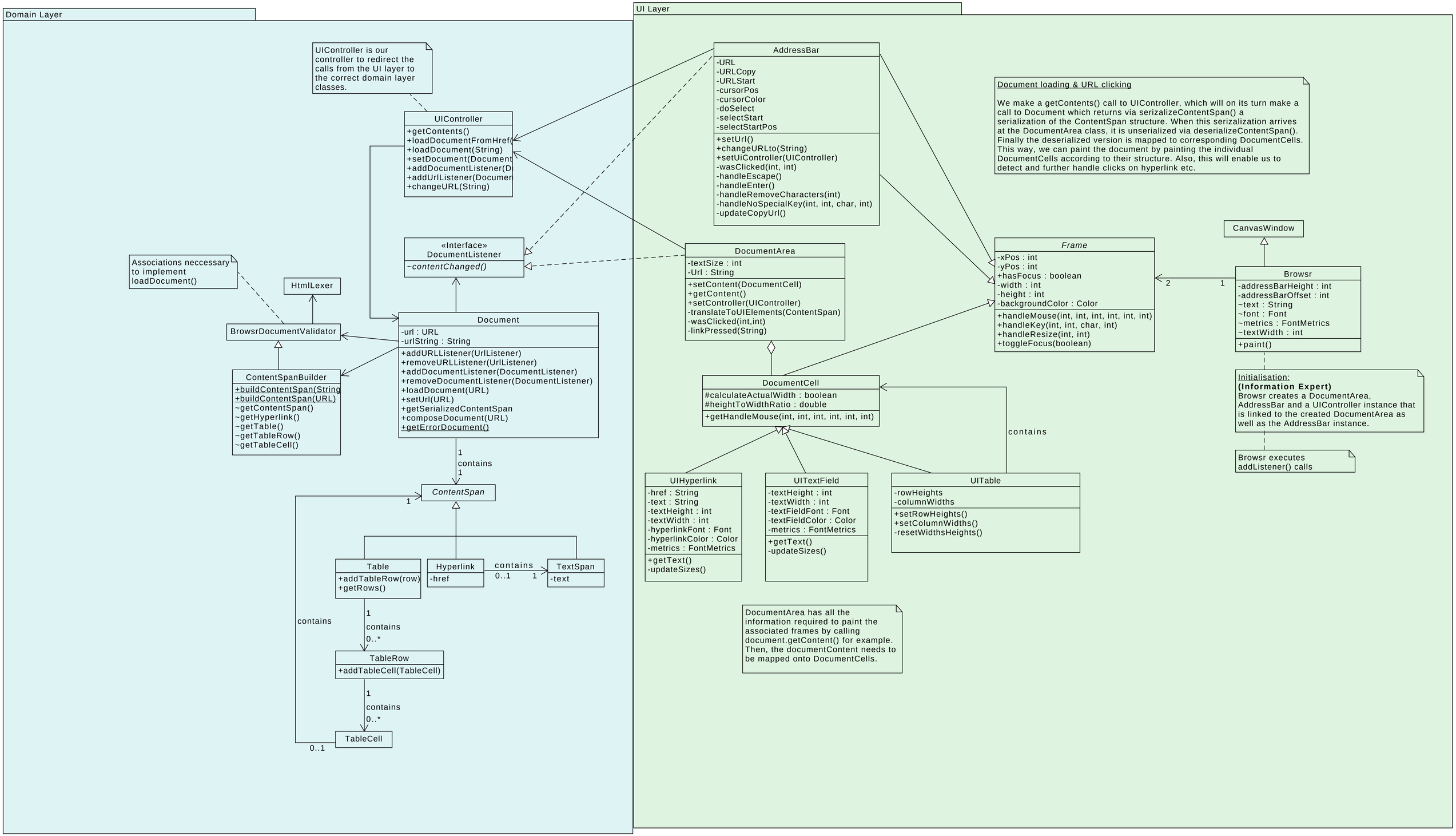
Domain Layer
UIController is our controller to redirect the calls from the UI layer to the correct domain layer classes.
UIController
+getContents()
+loadDocumentFromHref(
+loadDocument(String)
+setDocument(Document
+addDocumentListener(D
+addUrlListener(Documen
+changeURL(String)
«Interface»
DocumentListener
~contentChanged()
Associations neccessary to implement loadDocument()
HtmlLexer
BrowsrDocumentValidator
Document
-url : URL
-urlString : String
+addURLListener(UrlListener)
+removeURLListener(UrlListener)
+addDocumentListener(DocumentListener)
+removeDocumentListener(DocumentListener)
+loadDocument(URL)
+setUrl(URL)
+getSerializedContentSpan
+composeDocument(URL)
+getErrorDocument()
ContentSpanBuilder
+buildContentSpan(String
+buildContentSpan(URL)
~getContentSpan()
~getHyperlink()
~getTable()
~getTableRow()
~getTableCell()
1
contains
1
ContentSpan
1
Table
+addTableRow(row)
+getRows()
Hyperlink
-href
contains
0..1
1
TextSpan
-text
contains
1
contains
0..*
TableRow
+addTableCell(TableCell)
1
contains
0..*
TableCell
0..1
UI Layer
AddressBar
-URL
-URLCopy
-URLStart
-cursorPos
-cursorColor
-doSelect
-selectStart
-selectStartPos
+setUrl()
+changeURLto(String)
+setUiController(UIController)
-wasClicked(int, int)
-handleEscape()
-handleEnter()
-handleRemoveCharacters(int)
-handleNoSpecialKey(int, int, char, int)
-updateCopyUrl()
Document loading & URL clicking
We make a getContents() call to UIController, which will on its turn make a call to Document which returns via serializeContentSpan() a serialization of the ContentSpan structure. When this serizalization arrives at the DocumentArea class, it is unserialized via deserializeContentSpan(). Finally the deserialized version is mapped to corresponding DocumentCells. This way, we can paint the document by painting the individual DocumentCells according to their structure. Also, this will enable us to detect and further handle clicks on hyperlink etc.
CanvasWindow
Frame
-xPos : int
-yPos : int
+hasFocus : boolean
-width : int
-height : int
-backgroundColor : Color
+handleMouse(int, int, int, int, int, int)
+handleKey(int, int, char, int)
+handleResize(int, int)
+toggleFocus(boolean)
2
1
Browsr
-addressBarHeight : int
-addressBarOffset : int
~text : String
~font : Font
~metrics : FontMetrics
~textWidth : int
+paint()
DocumentArea
-textSize : int
-Url : String
+setContent(DocumentCell)
+getContent()
+setController(UIController)
-translateToUIElements(ContentSpan)
-wasClicked(int,int)
-linkPressed(String)
Initialisation:
(Information Expert)
Browsr creates a DocumentArea, AddressBar and a UIController instance that is linked to the created DocumentArea as well as the AddressBar instance.
Browsr executes addListener() calls
DocumentCell
#calculateActualWidth : boolean
#heightToWidthRatio : double
+getHandleMouse(int, int, int, int, int, int)
contains
UIHyperlink
-href : String
-text : String
-textHeight : int
-textWidth : int
-hyperlinkFont : Font
-hyperlinkColor : Color
-metrics : FontMetrics
+getText()
-updateSizes()
UITextField
-textHeight : int
-textWidth : int
-textFieldFont : Font
-textFieldColor : Color
-metrics : FontMetrics
+getText()
-updateSizes()
UITable
-rowHeights
-columnWidths
+setRowHeights()
+setColumnWidths()
-resetWidthsHeights()
DocumentArea has all the information required to paint the associated frames by calling document.getContent() for example. Then, the documentContent needs to be mapped onto DocumentCells.