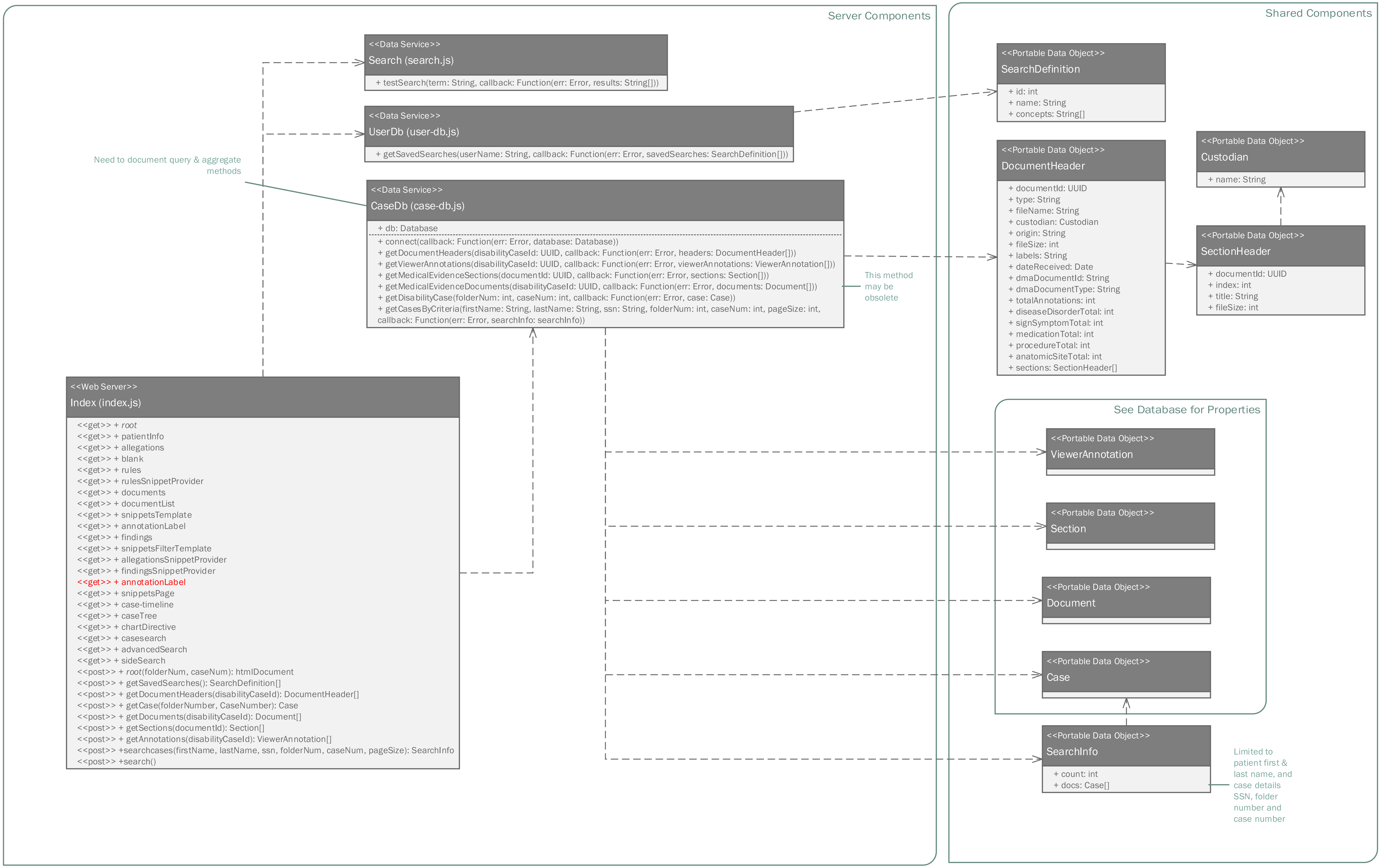
# NLP Viewer (CAV) Project

June 7, 2017



Server Components

<<Data Service>>
Search (search.js)

+ testSearch(term: String, callback: Function(err: Error, results: String[]))

<<Data Service>>
UserDb (user-db.js)

+ getSavedSearches(userName: String, callback: Function(err: Error, savedSearches: SearchDefinition[]))

Need to document query & aggregate methods

<<Data Service>>
CaseDb (case-db.js)

+ db: Database

+ connect(callback: Function(err: Error, database: Database))
+ getDocumentHeaders(disabilityCaseId: UUID, callback: Function(err: Error, headers: DocumentHeader[]))
+ getViewerAnnotations(disabilityCaseId: UUID, callback: Function(err: Error, viewerAnnotations: ViewerAnnotation[]))
+ getMedicalEvidenceSections(documentId: UUID, callback: Function(err: Error, sections: Section[]))
+ getMedicalEvidenceDocuments(disabilityCaseId: UUID, callback: Function(err: Error, documents: Document[]))
+ getDisabilityCase(folderNum: int, caseNum: int, callback: Function(err: Error, case: Case))
+ getCasesByCriteria(firstName: String, lastName: String, ssn: String, folderNum: int, caseNum: int, pageSize: int, callback: Function(err: Error, searchInfo: searchInfo))

This method may be obsolete

<<Web Server>>
Index (index.js)

<<get>> + *root*
<<get>> + patientInfo
<<get>> + allegations
<<get>> + blank
<<get>> + rules
<<get>> + rulesSnippetProvider
<<get>> + documents
<<get>> + documentList
<<get>> + snippetsTemplate
<<get>> + annotationLabel
<<get>> + findings
<<get>> + snippetsFilterTemplate
<<get>> + allegationsSnippetProvider
<<get>> + findingsSnippetProvider
<<get>> + annotationLabel
<<get>> + snippetsPage
<<get>> + case-timeline
<<get>> + caseTree
<<get>> + chartDirective
<<get>> + casesearch
<<get>> + advancedSearch
<<get>> + sideSearch
<<post>> + *root*(folderNum, caseNum): htmlDocument
<<post>> + getSavedSearches(): SearchDefinition[]
<<post>> + getDocumentHeaders(disabilityCaseId): DocumentHeader[]
<<post>> + getCase(folderNumber, CaseNumber): Case
<<post>> + getDocuments(disabilityCaseId): Document[]
<<post>> + getSections(documentId): Section[]
<<post>> + getAnnotations(disabilityCaseId): ViewerAnnotation[]
<<post>> +searchcases(firstName, lastName, ssn, folderNum, caseNum, pageSize): SearchInfo
<<post>> +search()

Shared Components

<<Portable Data Object>>
SearchDefinition

+ id: int
+ name: String
+ concepts: String[]

<<Portable Data Object>>
DocumentHeader

+ documentId: UUID
+ type: String
+ fileName: String
+ custodian: Custodian
+ origin: String
+ fileSize: int
+ labels: String
+ dateReceived: Date
+ dmaDocumentId: String
+ dmaDocumentType: String
+ totalAnnotations: int
+ diseaseDisorderTotal: int
+ signSymptomTotal: int
+ medicationTotal: int
+ procedureTotal: int
+ anatomicSiteTotal: int
+ sections: SectionHeader[]

<<Portable Data Object>>
Custodian

+ name: String

<<Portable Data Object>>
SectionHeader

+ documentId: UUID
+ index: int
+ title: String
+ fileSize: int

See Database for Properties

<<Portable Data Object>>
ViewerAnnotation

<<Portable Data Object>>
Section

<<Portable Data Object>>
Document

<<Portable Data Object>>
Case

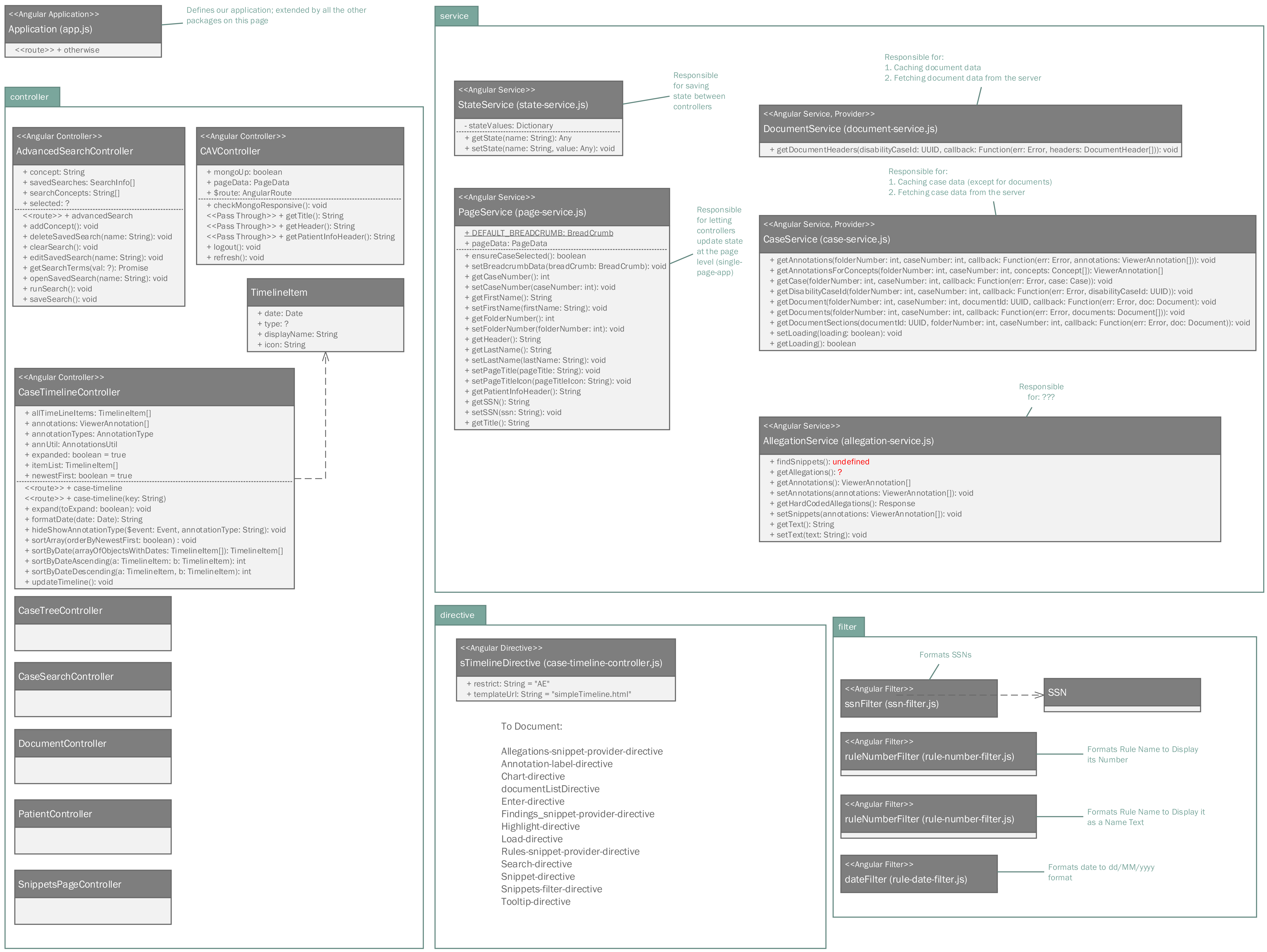
<<Portable Data Object>>
SearchInfo

+ count: int
+ docs: Case[]

Limited to patient first & last name, and case details SSN, folder number and case number

# NLP Viewer (CAV) Project

June 7, 2017



<<Angular Application>>
Application (app.js)
<<route>> + otherwise

Defines our application; extended by all the other packages on this page

## controller

<<Angular Controller>>
AdvancedSearchController
+ concept: String
+ savedSearches: SearchInfo[]
+ searchConcepts: String[]
+ selected: ?
<<route>> + advancedSearch
+ addConcept(): void
+ deleteSavedSearch(name: String): void
+ clearSearch(): void
+ editSavedSearch(name: String): void
+ getSearchTerms(val: ?): Promise
+ openSavedSearch(name: String): void
+ runSearch(): void
+ saveSearch(): void

<<Angular Controller>>
CAVController
+ mongoUp: boolean
+ pageData: PageData
+ $route: AngularRoute
+ checkMongoResponsive(): void
<<Pass Through>> + getTitle(): String
<<Pass Through>> + getHeader(): String
<<Pass Through>> + getPatientInfoHeader(): String
+ logout(): void
+ refresh(): void

TimelineItem
+ date: Date
+ type: ?
+ displayName: String
+ icon: String

<<Angular Controller>>
CaseTimelineController
+ allTimeLineItems: TimelineItem[]
+ annotations: ViewerAnnotation[]
+ annotationTypes: AnnotationType
+ annUtil: AnnotationsUtil
+ expanded: boolean = true
+ itemList: TimelineItem[]
+ newestFirst: boolean = true
<<route>> + case-timeline
<<route>> + case-timeline(key: String)
+ expand(toExpand: boolean): void
+ formatDate(date: Date): String
+ hideShowAnnotationType($event: Event, annotationType: String): void
+ sortArray(orderByNewestFirst: boolean) : void
+ sortByDate(arrayOfObjectsWithDates: TimelineItem[]): TimelineItem[]
+ sortByDateAscending(a: TimelineItem: b: TimelineItem): int
+ sortByDateDescending(a: TimelineItem, b: TimelineItem): int
+ updateTimeline(): void

CaseTreeController

CaseSearchController

DocumentController

PatientController

SnippetsPageController

## service

<<Angular Service>>
StateService (state-service.js)
- stateValues: Dictionary
+ getState(name: String): Any
+ setState(name: String, value: Any): void

Responsible for saving state between controllers

<<Angular Service>>
PageService (page-service.js)
+ DEFAULT_BREADCRUMB: BreadCrumb
+ pageData: PageData
+ ensureCaseSelected(): boolean
+ setBreadcrumbData(breadCrumb: BreadCrumb): void
+ getCaseNumber(): int
+ setCaseNumber(caseNumber: int): void
+ getFirstName(): String
+ setFirstName(firstName: String): void
+ getFolderNumber(): int
+ setFolderNumber(folderNumber: int): void
+ getHeader(): String
+ getLastName(): String
+ setLastName(lastName: String): void
+ setPageTitle(pageTitle: String): void
+ setPageTitleIcon(pageTitleIcon: String): void
+ getPatientInfoHeader(): String
+ getSSN(): String
+ setSSN(ssn: String): void
+ getTitle(): String

Responsible for letting controllers update state at the page level (single-page-app)

Responsible for:
1. Caching document data
2. Fetching document data from the server

<<Angular Service, Provider>>
DocumentService (document-service.js)
+ getDocumentHeaders(disabilityCaseId: UUID, callback: Function(err: Error, headers: DocumentHeader[])): void

Responsible for:
1. Caching case data (except for documents)
2. Fetching case data from the server

<<Angular Service, Provider>>
CaseService (case-service.js)
+ getAnnotations(folderNumber: int, caseNumber: int, callback: Function(err: Error, annotations: ViewerAnnotation[])): void
+ getAnnotationsForConcepts(folderNumber: int, caseNumber: int, concepts: Concept[]): ViewerAnnotation[]
+ getCase(folderNumber: int, caseNumber: int, callback: Function(err: Error, case: Case)): void
+ getDisabilityCaseId(folderNumber: int, caseNumber: int, callback: Function(err: Error, disabilityCaseId: UUID)): void
+ getDocument(folderNumber: int, caseNumber: int, documentId: UUID, callback: Function(err: Error, doc: Document): void
+ getDocuments(folderNumber: int, caseNumber: int, callback: Function(err: Error, documents: Document[])): void
+ getDocumentSections(documentId: UUID, folderNumber: int, caseNumber: int, callback: Function(err: Error, doc: Document)): void
+ setLoading(loading: boolean): void
+ getLoading(): boolean

Responsible for: ???

<<Angular Service>>
AllegationService (allegation-service.js)
+ findSnippets(): undefined
+ getAllegations(): ?
+ getAnnotations(): ViewerAnnotation[]
+ setAnnotations(annotations: ViewerAnnotation[]): void
+ getHardCodedAllegations(): Response
+ setSnippets(annotations: ViewerAnnotation[]): void
+ getText(): String
+ setText(text: String): void

## directive

<<Angular Directive>>
sTimelineDirective (case-timeline-controller.js)
+ restrict: String = "AE"
+ templateUrl: String = "simpleTimeline.html"

To Document:

Allegations-snippet-provider-directive
Annotation-label-directive
Chart-directive
documentListDirective
Enter-directive
Findings_snippet-provider-directive
Highlight-directive
Load-directive
Rules-snippet-provider-directive
Search-directive
Snippet-directive
Snippets-filter-directive
Tooltip-directive

## filter

Formats SSNs

<<Angular Filter>>
ssnFilter (ssn-filter.js)

SSN

<<Angular Filter>>
ruleNumberFilter (rule-number-filter.js)

Formats Rule Name to Display its Number

<<Angular Filter>>
ruleNumberFilter (rule-number-filter.js)

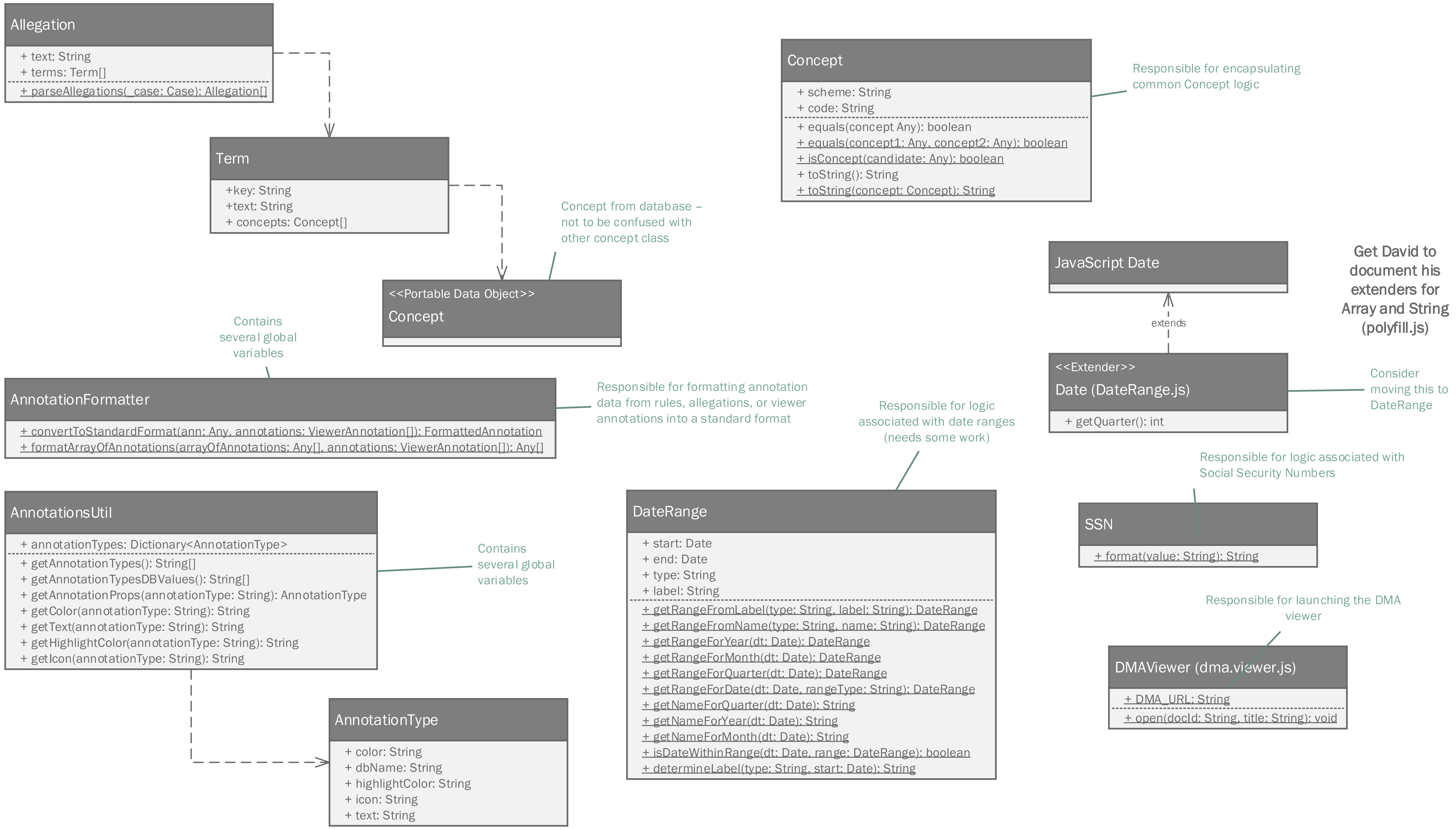
Formats Rule Name to Display it as a Name Text

<<Angular Filter>>
dateFilter (rule-date-filter.js)

Formats date to dd/MM/yyyy format

# NLP Viewer (CAV) Project

June 7, 2017



Allegation
+ text: String
+ terms: Term[]
+ parseAllegations(_case: Case): Allegation[]

Term
+key: String
+text: String
+ concepts: Concept[]

Concept from database – not to be confused with other concept class

<<Portable Data Object>>
Concept

Contains several global variables

AnnotationFormatter
+ convertToStandardFormat(ann: Any, annotations: ViewerAnnotation[]): FormattedAnnotation
+ formatArrayOfAnnotations(arrayOfAnnotations: Any[], annotations: ViewerAnnotation[]): Any[]

Responsible for formatting annotation data from rules, allegations, or viewer annotations into a standard format

AnnotationsUtil
+ annotationTypes: Dictionary<AnnotationType>
+ getAnnotationTypes(): String[]
+ getAnnotationTypesDBValues(): String[]
+ getAnnotationProps(annotationType: String): AnnotationType
+ getColor(annotationType: String): String
+ getText(annotationType: String): String
+ getHighlightColor(annotationType: String): String
+ getIcon(annotationType: String): String

Contains several global variables

AnnotationType
+ color: String
+ dbName: String
+ highlightColor: String
+ icon: String
+ text: String

Concept
+ scheme: String
+ code: String
+ equals(concept Any): boolean
+ equals(concept1: Any, concept2: Any): boolean
+ isConcept(candidate: Any): boolean
+ toString(): String
+ toString(concept: Concept): String

Responsible for encapsulating common Concept logic

JavaScript Date

extends

<<Extender>>
Date (DateRange.js)
+ getQuarter(): int

Consider moving this to DateRange

Get David to document his extenders for Array and String (polyfill.js)

Responsible for logic associated with date ranges (needs some work)

DateRange
+ start: Date
+ end: Date
+ type: String
+ label: String
+ getRangeFromLabel(type: String, label: String): DateRange
+ getRangeFromName(type: String, name: String): DateRange
+ getRangeForYear(dt: Date): DateRange
+ getRangeForMonth(dt: Date): DateRange
+ getRangeForQuarter(dt: Date): DateRange
+ getRangeForDate(dt: Date, rangeType: String): DateRange
+ getNameForQuarter(dt: Date): String
+ getNameForYear(dt: Date): String
+ getNameForMonth(dt: Date): String
+ isDateWithinRange(dt: Date, range: DateRange): boolean
+ determineLabel(type: String, start: Date): String

Responsible for logic associated with Social Security Numbers

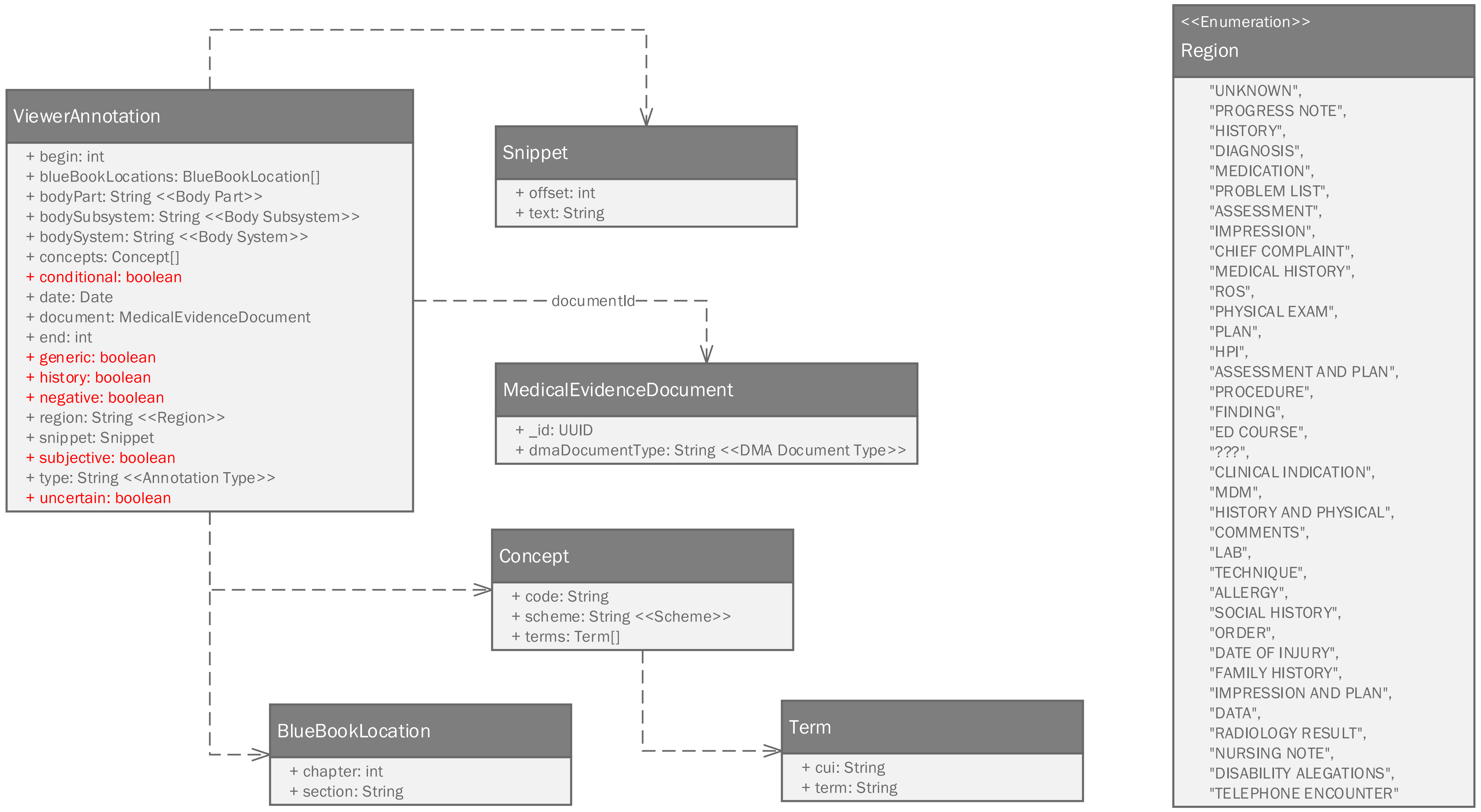
SSN
+ format(value: String): String

Responsible for launching the DMA viewer

DMAViewer (dma.viewer.js)
+ DMA_URL: String
+ open(docId: String, title: String): void

# NLP Viewer (CAV) Project

June 7, 2017



ViewerAnnotation

- + begin: int
- + blueBookLocations: BlueBookLocation[]
- + bodyPart: String <<Body Part>>
- + bodySubsystem: String <<Body Subsystem>>
- + bodySystem: String <<Body System>>
- + concepts: Concept[]
- + conditional: boolean
- + date: Date
- + document: MedicalEvidenceDocument
- + end: int
- + generic: boolean
- + history: boolean
- + negative: boolean
- + region: String <<Region>>
- + snippet: Snippet
- + subjective: boolean
- + type: String <<Annotation Type>>
- + uncertain: boolean

Snippet

- + offset: int
- + text: String

MedicalEvidenceDocument

- + _id: UUID
- + dmaDocumentType: String <<DMA Document Type>>

Concept

- + code: String
- + scheme: String <<Scheme>>
- + terms: Term[]

BlueBookLocation

- + chapter: int
- + section: String

Term

- + cui: String
- + term: String

<<Enumeration>>
Region

"UNKNOWN",
"PROGRESS NOTE",
"HISTORY",
"DIAGNOSIS",
"MEDICATION",
"PROBLEM LIST",
"ASSESSMENT",
"IMPRESSION",
"CHIEF COMPLAINT",
"MEDICAL HISTORY",
"ROS",
"PHYSICAL EXAM",
"PLAN",
"HPI",
"ASSESSMENT AND PLAN",
"PROCEDURE",
"FINDING",
"ED COURSE",
"???",
"CLINICAL INDICATION",
"MDM",
"HISTORY AND PHYSICAL",
"COMMENTS",
"LAB",
"TECHNIQUE",
"ALLERGY",
"SOCIAL HISTORY",
"ORDER",
"DATE OF INJURY",
"FAMILY HISTORY",
"IMPRESSION AND PLAN",
"DATA",
"RADIOLOGY RESULT",
"NURSING NOTE",
"DISABILITY ALEGATIONS",
"TELEPHONE ENCOUNTER"