

Progetto Basi di Dati A.A. 2017-2018

Alessandro Marchetti

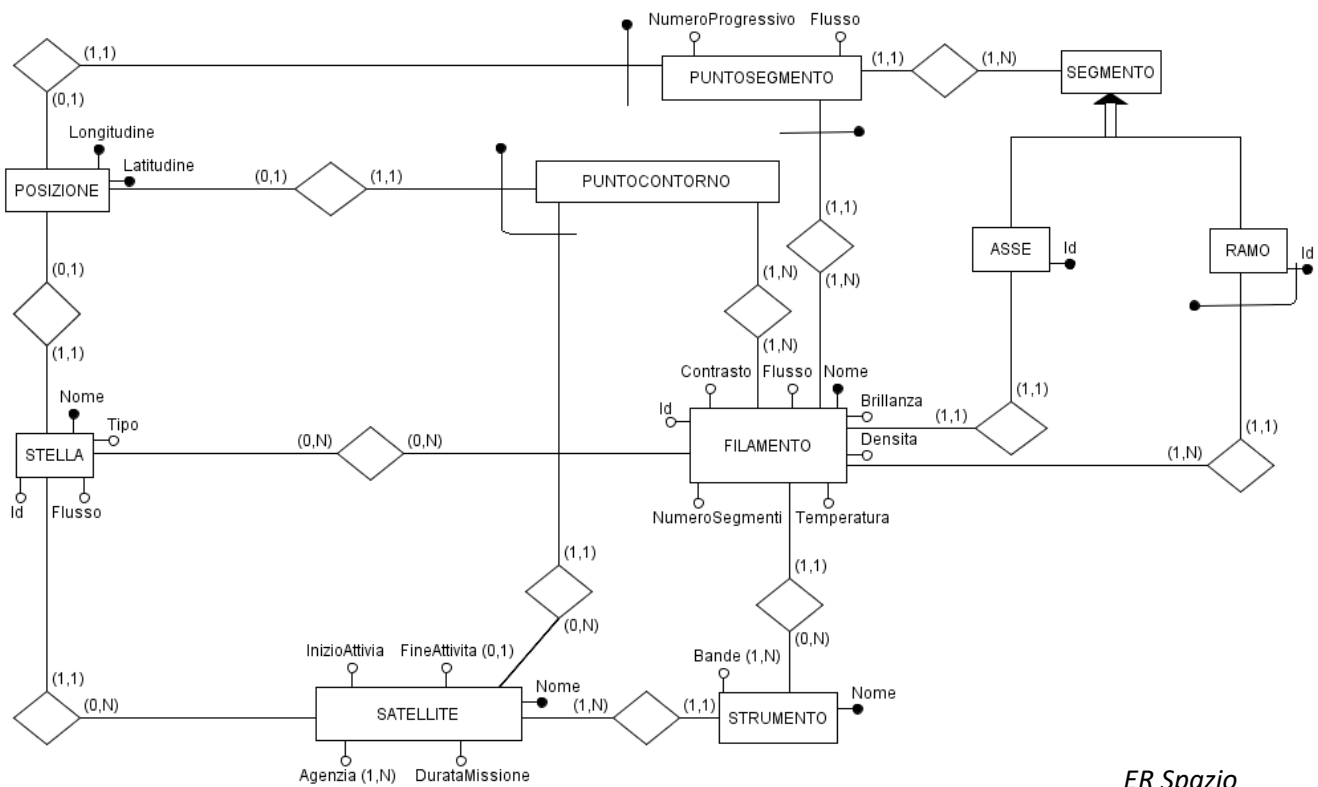
Federico Viglietta

Tommaso Villa

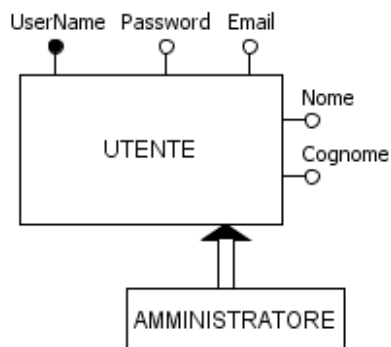
Relazione

Progettazione concettuale

Schema E-R



ER Spazio



ER Utenti

Dizionario dei dati-Tabella delle Entità

Entità	Descrizione	Attributi	Identificatore
Satellite	Oggetto orbitante impiegato per ottenere informazioni su stelle e filamenti. Possiede uno o più strumenti.	Nome, Inizio attività, Termine attività, Durata missione, Agenzia spaziale	Nome
Strumento	Apparecchio a disposizione di un satellite per effettuare misurazioni su stelle e filamenti. Copre una o più bande di frequenza.	Nome, Banda	Nome, Banda
Filamento (Struttura estesa)	Formazione gassosa con pattern filamentoso e elevata brillantezza. Può contenere stelle (in formazione) al suo interno. Possiede un perimetro(contorno). Possiede uno scheletro costituito da segmenti.	Id, Nome, Flusso totale, Densità media, Temperatura media, Ellitticità, Contrasto, Brillanza, Numero segmenti	Nome
Posizione galattica	Coppia di coordinate galattiche (longitudine e latitudine).	Longitudine galattica, Latitudine galattica	Longitudine galattica, Latitudine galattica
Punto del Contorno (Punto del Perimetro)	Posizione galattica in cui un satellite individua un punto appartenente al contorno di un filamento.		Posizione galattica, Satellite
Punto del Segmento (Posizione dello Scheletro)	Posizione galattica in cui uno strumento individua un punto appartenente allo scheletro di un filamento.	Numero progressivo, Flusso misurato	Posizione galattica, Filamento
Stella	Oggetto luminoso puntiforme. Può trovarsi dentro il perimetro di uno o più filamenti.	Id, Nome, Flusso, Classificazione	Nome
Segmento	Insieme di punti dello scheletro di un filamento. Può essere un Asse o un Ramo.	Id	
Asse (Spina Dorsale)	Segmento principale dello scheletro di un filamento.	Id	Id, Filamento
Ramo	Segmento secondario dello scheletro di un filamento.	Id	Id, Filamento

Dizionario dei dati-Tabella delle Associazioni

Relazione	Descrizione	Entità coinvolte (cardinalità)	Attributi
Strumento-Satellite	Associa ad uno strumento il satellite che lo ha a bordo.	Strumento (1,1) Satellite (1,n)	
Filamento-Strumento	Associa ad un filamento lo strumento con il quale è stato osservato.	Filamento (1,1) Strumento (0,n)	
Asse-Filamento	Associa ad un asse il filamento a cui appartiene.	Asse (1,1) Filamento (1,1)	
Ramo-Filamento	Associa ad un ramo il filamento a cui appartiene.	Ramo (1,1) Filamento (1,n)	
Stella-Filamento	Associa una stella e un filamento tali che la stella è dentro il bordo del filamento.	Stella (0,n) Filamento (0,n)	
Stella-Satellite	Associa ad una stella il satellite con la quale è stata osservata.	Stella (1,1) Satellite (0,n)	
Stella-Posizione galattica	Associa ad una stella la posizione galattica in cui si trova.	Stella (1,1) Posizione (0,1)	
Posizione galattica-Punto segmento	Associa ad un punto del segmento la posizione galattica in cui si trova.	Punto segmento (1,1) Posizione galattica (0,n)	
Posizione galattica-Punto contorno	Associa ad un punto del contorno la posizione galattica in cui si trova.	Punto contorno (1,1) Posizione galattica (0,n)	
Punto contorno-Satellite	Associa ad un punto del contorno il satellite con il quale è stato rilevato.	Punto contorno (1,1) Satellite (0,n)	
Punto segmento-Filamento	Associa ad un punto del segmento il filamento a cui appartiene.	Punto segmento (1,1) Filamento (1,n)	
Punto contorno-Filamento	Associa un punto del contorno al filamento a cui appartiene.	Punto contorno (1,n) Filamento (1,n)	
Punto segmento-Segmento	Associa un punto del segmento al segmento a cui appartiene.	Punto segmento (1,1) Segmento (1,n)	

Business Rules

- I punti dello scheletro di un filamento NON DEVONO essere punti del contorno dello stesso filamento.
- La latitudine del centroide di una figura SI OTTIENE calcolando la media delle latitudini dei suoi punti.
- La longitudine del centroide di una figura SI OTTIENE calcolando la media delle longitudini dei suoi punti.
- La distanza tra due punti SI OTTIENE valutando la *distanza euclidea* tra gli stessi.
- L'estensione longitudinale di una figura SI OTTIENE calcolando la distanza tra il minimo e il massimo delle posizioni longitudinali dei suoi punti.
- L'estensione latitudinale di una figura SI OTTIENE calcolando la distanza tra il minimo e il massimo delle posizioni latitudinale dei suoi punti.
- La percentuale di brillantezza di un filamento SI OTTIENE dal contrasto.
- La percentuale di brillantezza di un filamento DEVE essere $[100(\text{Contrasto} - 1)]$.
- Una stella che appartiene a un filamento DEVE verificare la formula specificata nel requisito REQ-FN-9.
- Un estremo di un segmento DEVE essere il punto del segmento con numero progressivo massimo o minimo.

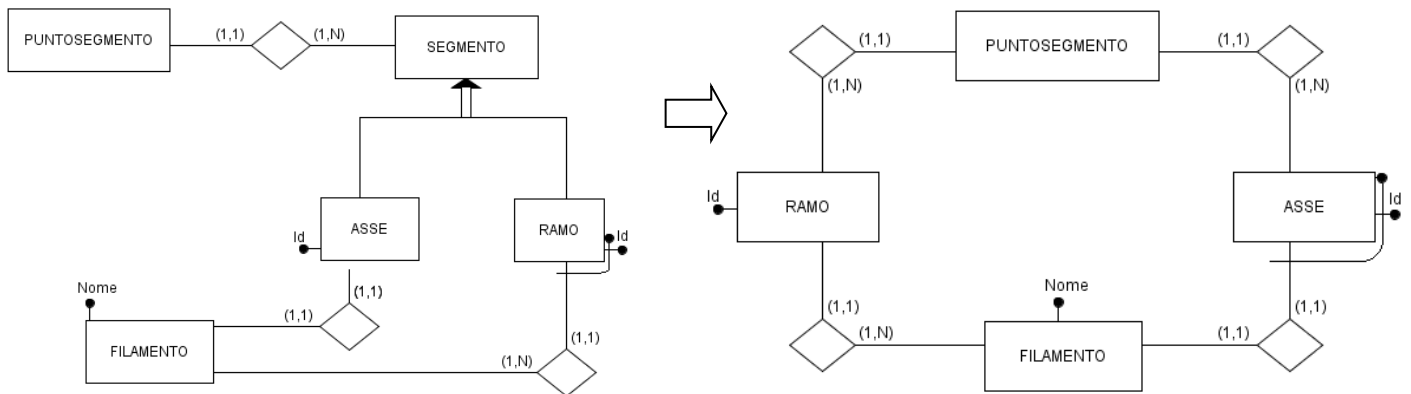
Progettazione Logica

Eliminazione delle generalizzazioni

Per quanto concerne la generalizzazione relativa alle entità SEGMENTO, ASSE e RAMO le possibilità di ristrutturazione sono:

- (1) Accorpamento di ASSE e RAMO in SEGMENTO: la soluzione richiede di aggiungere un attributo Tipo a SEGMENTO e un'associazione da SEGMENTO a FILAMENTO per effettuare l'identificazione esterna dell'entità SEGMENTO;
- (2) Accorpamento di SEGMENTO in ASSE e RAMO: si osservi innanzitutto che la soluzione è attuabile in quanto la generalizzazione in oggetto è totale. Essa richiede di replicare per ASSE e RAMO l'associazione SEGMENTO-PUNTOSEGMENTO presente nello schema concettuale;
- (3) Sostituzione della generalizzazione con due associazioni: la soluzione è sintatticamente valida ma non lo è semanticamente. Infatti, RAMO e ASSE hanno un proprio identificativo, per cui introdurre le associazioni in luogo della generalizzazione farebbe perder traccia della relazione *is-a-kind-of* che sussiste tra le entità figlie e l'entità padre.

Pertanto, è emerso che le soluzioni di ristrutturazione più adatte sono la (1) e la (2). D'altra parte, visto che entrambe aggiungono un'associazione e visto che l'entità SEGMENTO è priva di attributi si è scelto di attuare la soluzione (2), come illustrato nella figura seguente.



Accorpamento/Partizionamento

- L'attributo multi-valore Agenzie dell'entità SATELLITE è stato tradotto introducendo l'entità AGENZIA (con attributo identificativo Nome) e l'associazione molti a molti SATELLITE-AGENZIA.
- L'attributo multi-valore Bande invece non è stato reificato ma si è scelto di sostituirlo con l'attributo semplice di tipo stringa ElencoBande; infatti, le bande non sono concepibili come entità.

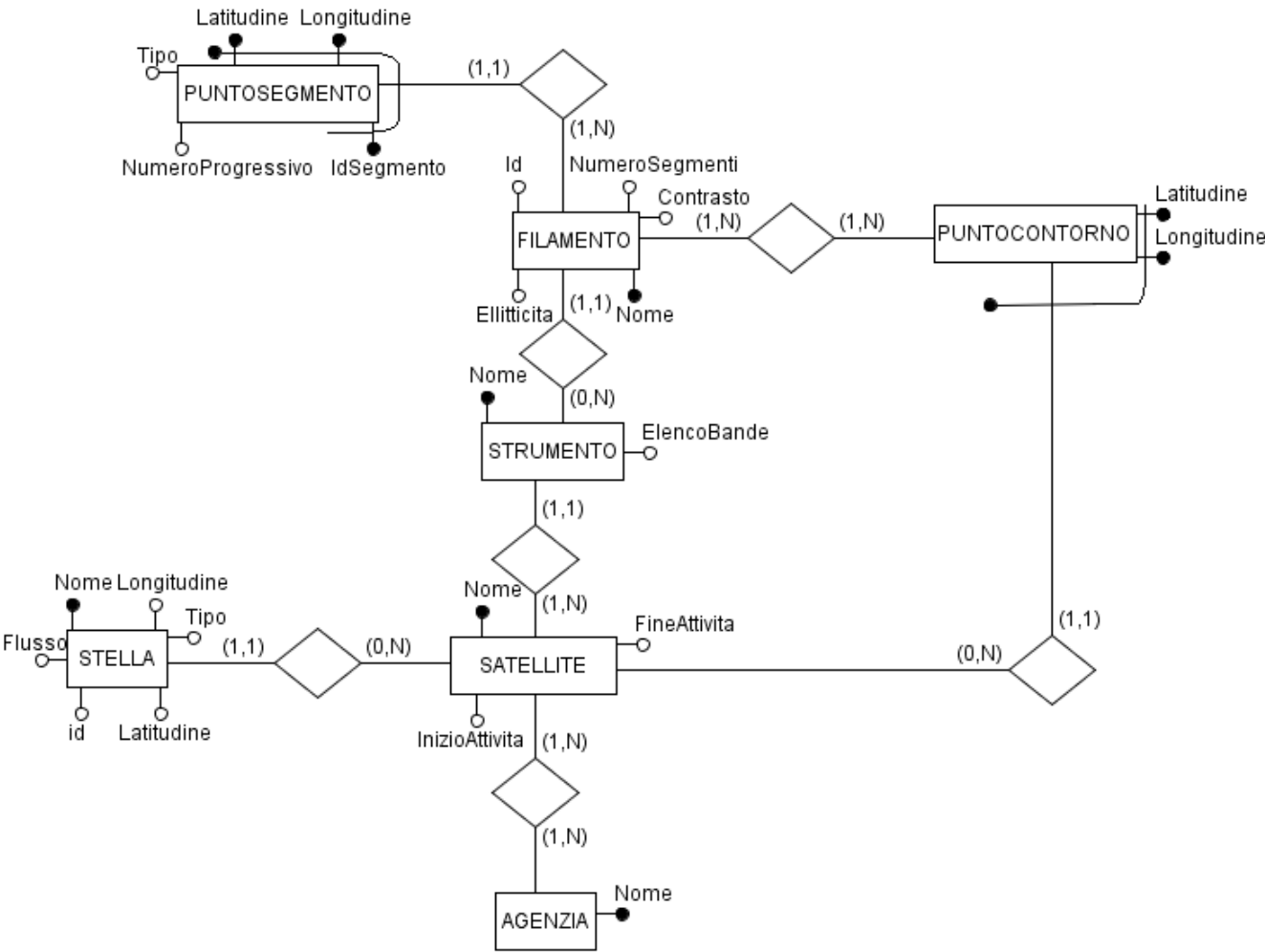
Analisi delle ridondanze

- L'attributo Brillanza dell'entità FILAMENTO è ridondante perché può essere ricavato dall'attributo Contrasto applicando la formula indicata nel requisito REQ-FN-6 delle specifiche. Lo stesso si può dire viceversa per l'attributo Contrasto. Uno dei due attributi può quindi essere rimosso dallo schema concettuale. La scelta è ricaduta sull'attributo Brillanza per i seguenti motivi: vista la formula di REQ-FN-6 calcolare Brillanza on-demand non impatta eccessivamente sui tempi d'attesa dell'utente; inoltre, questa scelta alleggerisce l'importazione dei file richiesta da REQ-FN-3/4, mentre mantenere un attributo Brillanza imporrebbe in fase di import la conversione Contrasto->Brillanza per ogni filamento inserito.
- L'associazione STELLA-FILAMENTO è ridondante perché l'appartenenza di una stella ad un filamento può essere verificata applicando la formula indicata nel requisito REQ-FN-9. Nonostante essa possa tornare utile nei requisiti REQ-FN-9 e REQ-FN-10 si è scelto di rimuoverla per evitare di appesantire la fase di importazione dei file.
- L'attributo DurataMissione dell'entità SATELLITE è stato rimosso in quanto ricavabile dagli attributi InizioAttività e FineAttività della stessa entità.
- L'asse principale di un filamento può essere ricavato considerando le istanze di PUNTOSEGMENTO di tipo "S" che partecipano con esso all'associazione PUNTOSEGMENTO-FILAMENTO. Analogamente, si possono ottenere i rami del filamento tenendo presente che i punti di un ramo avranno in comune il valore di IdSegmento. Per questo, le entità ASSE e RAMO sono state rimosse.
- L'attributo NumeroSegmenti dell'entità FILAMENTO è ridondante perché può essere ricavato contando i rami del filamento, ottenibili come indicato al punto precedente. Tuttavia, poiché la conoscenza del numero di segmenti di un filamento è necessaria in ben 2 requisiti (REQ-FN-5 e REQ-FN-7), visto l'impatto limitato sull'importazione dei file, si è scelto di mantenere l'attributo.
- Si è scelto di rimuovere l'entità POSIZIONEGALATTICA aggiungendo gli attributi Longitudine, Latitudine alle entità PUNTOSEGMENTO, PUNTOCONTORNO e STELLA. Difatti, si è ritenuto che la posizione galattica non fosse concepibile come entità. Si tratta in ogni caso di una ridondanza "indolore": i campi Latitudine e Longitudine sarebbero comunque stati necessari nelle tabelle PUNTOSEGMENTO e PUNTOCONTORNO per via dell'identificazione esterna da essi fornita.
- Infine, sono stati rimossi i seguenti attributi perché non coinvolti in nessuna delle operazioni riportate nel file di specifiche: Flusso di PUNTOSEGMENTO, Flusso di FILAMENTO, Densità di FILAMENTO, Temperatura di FILAMENTO.

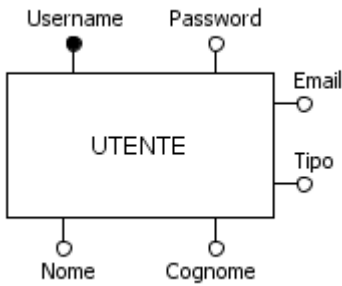
Scelta degli identificatori principali

- L'entità STELLA può essere identificata attraverso Latitudine, Longitudine e SATELLITE (esternamente) oppure tramite Nome. Per praticità si è scelto di usare come identificatore l'attributo Nome.
- Analogamente, si è scelto di identificare l'entità FILAMENTO mediante l'attributo Nome piuttosto che utilizzare Numero e STRUMENTO(esternamente).
- Nell'identificazione di PUNTOSEGMENTO è necessario anche l'attributo IdSegmento in quanto in un punto possono intersecarsi più rami.
- Nell'identificazione di PUNTOCONTORNO è necessario il riferimento a SATELLITE perché satelliti diversi potrebbero rilevare nella stessa posizione galattica punti del contorno appartenenti a filamenti diversi.

Schema E-R Ristrutturato



ER Ristrutturato Spazio



ER Ristrutturato Utenti

Schema Logico

AGENZIA (Nome)

SATELLITE (Nome, InizioAttivita, FineAttivita)

AGENZIA-SATELLITE (Satellite, Agenzia),

con vincolo di FK tra Satellite e SATELLITE.Nome e tra Agenzia e AGENZIA.Nome

STRUMENTO (Nome, ElencoBande, Satellite)

con vincolo di FK tra Satellite e SATELLITE.Nome

STELLA (Nome, Id, Latitudine, Longitudine, Flusso, Tipo, Satellite),

con vincolo di FK tra Satellite e SATELLITE.Nome

FILAMENTO (Nome, Elletticità, Contrasto, Id, NumeroSegmenti, Strumento),

con vincolo di FK tra Strumento e STRUMENTO.Nome

PUNTOCONTORNO (Latitudine, Longitudine, Satellite),

con vincolo di FK tra Satellite e SATELLITE.Nome

PUNTOSEGMENTO (Latitudine, Longitudine, Filamento, IdSegmento, NumeroProgressivo, Tipo),

con vincolo di FK tra Filamento e FILAMENTO.Nome

PUNTODELCONTORNO-FILAMENTO (Latitudine, Longitudine, Strumento, Filamento),

con vincolo di FK tra Latitudine, Longitudine, Satellite e PUNTOCONTORNO.Latitudine, PUNTOCONTORNO.Longitudine, PUNTOCONTORNO.Satellite e tra Filamento e FILAMENTO.Nome

UTENTE(Username, Password, Nome, Cognome, Email, Tipo)

Progettazione Fisica

Per migliorare le performance dell'applicazione sono stati aggiunti i seguenti indici:

- Indice "filamento_id_index" sulla colonna "id" della tabella FILAMENTO;
- Indice "ptcontorno_latitudine_index" sulla colonna "latitudine" della tabella PUNTOCONTORNO;
- Indice "ptcontorno_longitudine_index" sulla colonna "longitudine" della tabella PUNTOCONTORNO;
- Indice "ptcontornofilamento_latitudine_index" sulla colonna "latitudine" della tabella PUNTOCONTORNOFILAMENTO;
- Indice "ptcontornofilamento_longitudine_index" sulla colonna "longitudine" della tabella PUNTOCONTORNOFILAMENTO;
- Indice "ptcontornofilamento_filamento_index" sulla colonna "filamento" della tabella PUNTOCONTORNOFILAMENTO;
- Indice "ptsegmento_latitudine_index" sulla colonna "latitudine" della tabella PUNTOSEGMENTO;
- Indice "ptsegmento_longitudine_index" sulla colonna "longitudine" della tabella PUNTOSEGMENTO;
- Indice "ptsegmento_filamento_index" sulla colonna "filamento" della tabella PUNTOSEGMENTO;

Nota:

Come riportato nel paragrafo sulla Progettazione Concettuale è richiesto che:

- Due segmenti appartenenti a due filamenti diversi non abbiano punti in comune
- Due punti del segmento non si sovrappongono ai punti del contorno del filamento cui appartengono.

Per la gestione di tali vincoli sarebbe una valida soluzione introdurre apposite clausole **check** nella definizione delle tabelle PUNTOSEGMENTO e PUNTOCONTORNO. All'interno delle clausole sarebbe necessaria l'uso di query annidate. Poiché alcuni DBMS (e.g. PostgreSQL) non supportano query annidate nella definizione dei **check**, si è scelto di gestire i vincoli direttamente a livello applicativo.

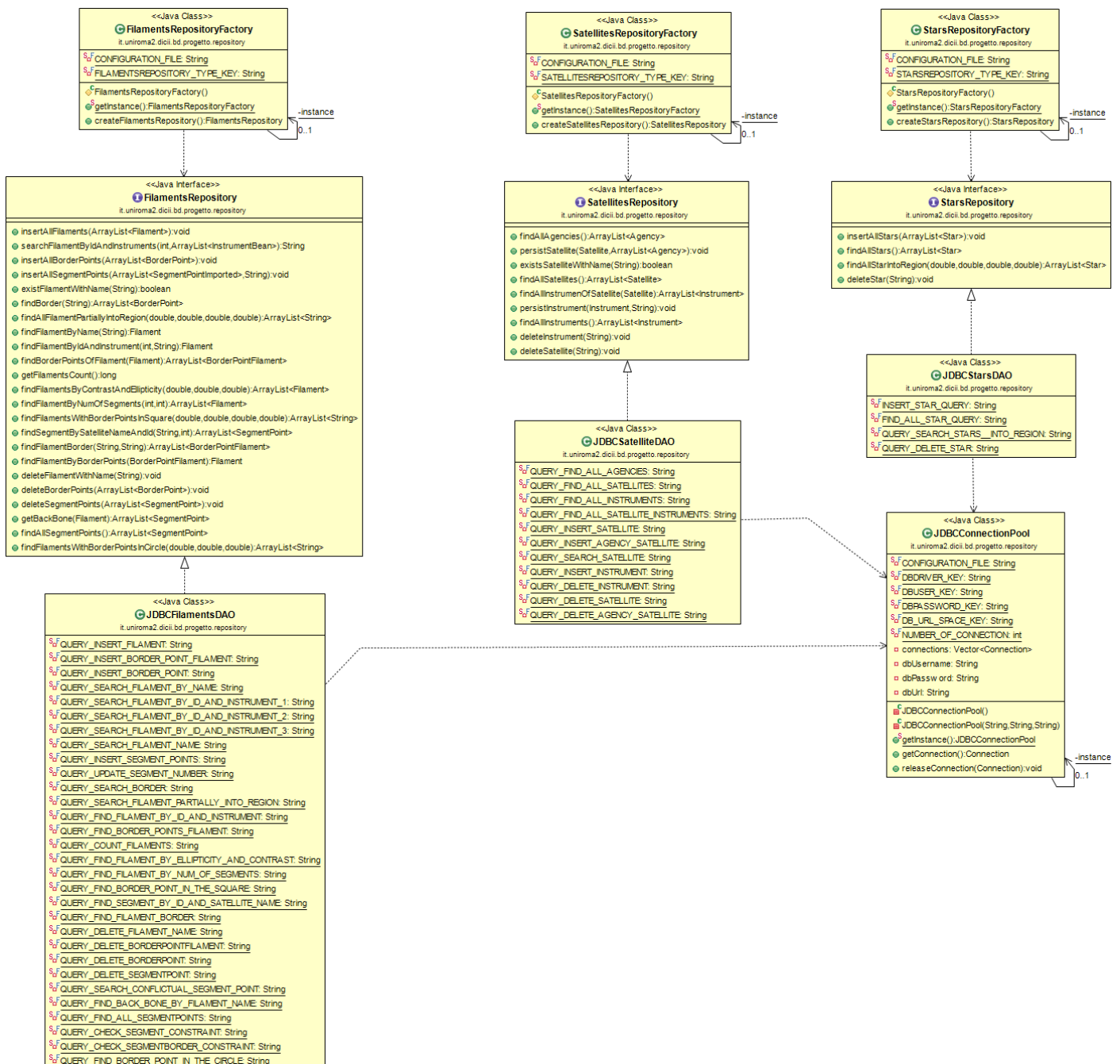
Design dell'applicazione

DCD (Design Class Diagrams)

Avvertenza: le seguenti immagini potrebbero non essere leggibili. Nel folder in allegato sono disponibili le immagini dei class diagram per una migliore risoluzione.

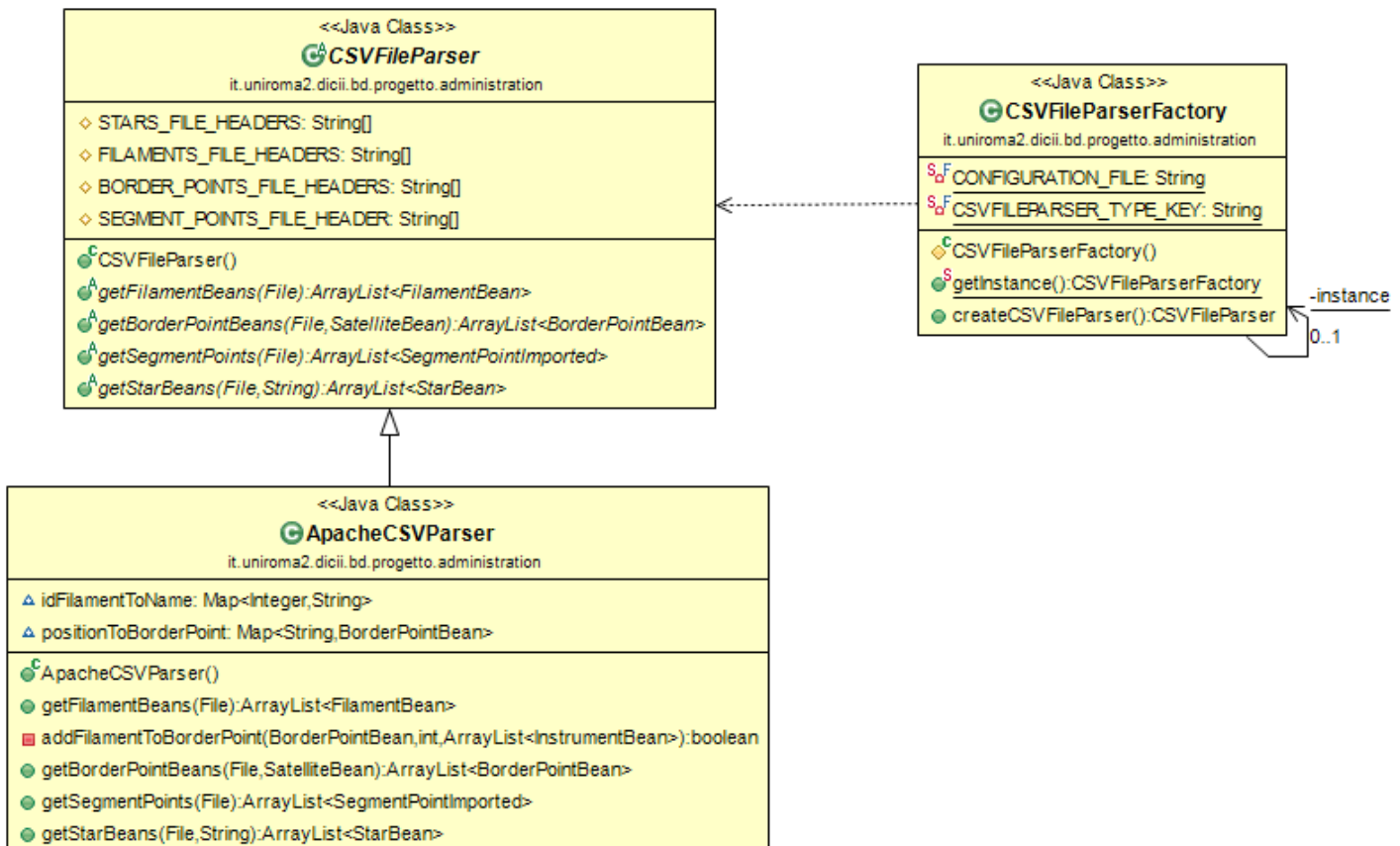
• Repository

Lo strato di persistenza viene gestito attraverso il pattern Repository. Ciascun oggetto repository simula la presenza in memoria di una collezione di entità (e.g. UsersRepository simula la presenza in memoria di una collezione di utenti). Ogni Repository è un'interfaccia che astrae dal modo particolare in cui la persistenza è gestita ad un livello più basso (JDBC, JPA, File...). Per questo la costruzione di un Repository avviene sempre tramite una Factory che istanzia l'implementazione in uso dell'interfaccia leggendone a runtime il tipo da un file di configurazione.



- **Parsing dei file**

L'interfaccia CSVFileParser mette a disposizione una serie di operazioni necessarie a parsare i file dei dati scientifici da importare. L'interfaccia nasconde all'utilizzatore la strategia di parsing concretamente adottata. Per questo l'oggetto CSVFileParser viene sempre costruito attraverso la classe CSVFileParserFactory, che ricava l'implementazione corrente dell'interfaccia da un file di configurazione e la istanzia.



- **Task**

Nei requisiti in cui è prevista un'interazione costosa con lo strato di persistenza, quest'ultima viene realizzata su un thread separato da quello che gestisce la GUI. Poiché nel progetto si utilizza la libreria JAVAFX per la gestione della GUI questo meccanismo viene realizzato attraverso oggetti di tipo Task.

- **JDBC: Gestione delle connessioni**

Con l'obiettivo di garantire maggiore coesione e riusabilità gli oggetti JDBCDAO delegano la gestione della connessione con il database ad una classe esterna: JDBCConnectionPool. Quando il JDBCDAO rilascia una connessione, la classe JDBCConnectionPool non la chiude, bensì la conserva per richieste di connessioni successive. Il riutilizzo delle connessioni riduce così l'overhead dato dall'apertura e dalla chiusura delle connessioni. (Si noti tuttavia che questo miglioramento si apprezzerrebbe maggiormente nello scenario di una web application.)

- **Window Manager**

La classe WindowManager gestisce la GUI in termini di transizioni tra menù, apertura di finestre...

- **Gestione utenti**

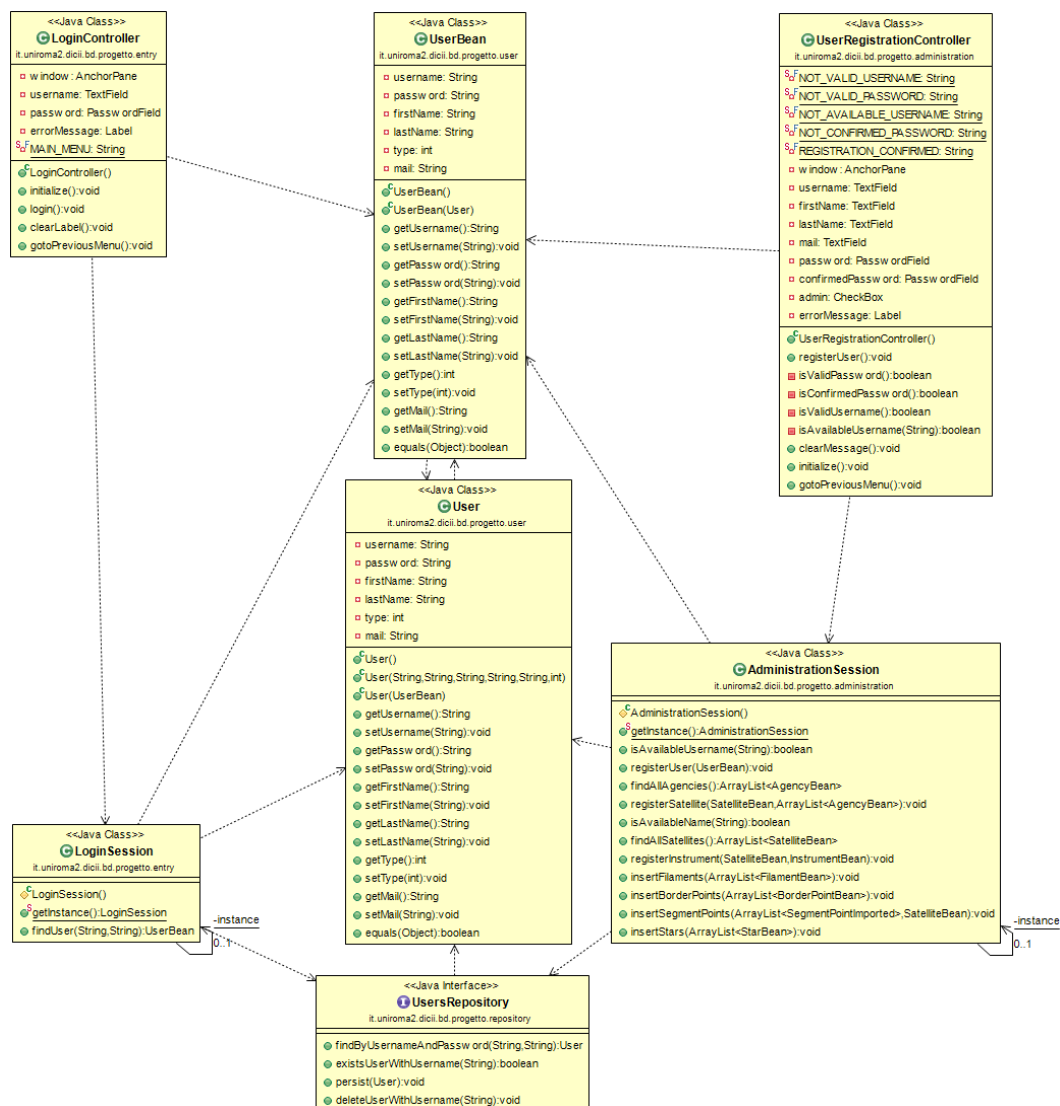
Il seguente class diagram illustra le classi coinvolte nell'implementazione dei requisiti 1, 2 e 3.2.

- **Login:**

- La classe LoginController passa all'oggetto AdministrationSession l'username e la password inseriti dall'utente;
- Il session richiede all'interfaccia UsersRepository l'utente (oggetto User) avente l'username e la password ricevuti dal controller;
- Ricevuto l'oggetto User, il session lo wrappa all'interno di uno UserBean che restituisce al controller;
- Se l'oggetto restituito dal session è null (i.e. non esiste un utente con l'username e la password indicati) il controller mostra un messaggio d'errore. Altrimenti, valorizza con l'oggetto UserBean ricevuto l'attributo statico della classe MainController (non presente nel class diagram) alla quale viene passato il controllo della GUI. La classe MainController utilizzerà l'attributo per riconoscere quando l'utente ha i permessi per accedere alle operazioni riservate all'amministratore.

- **Registrazione utente:**

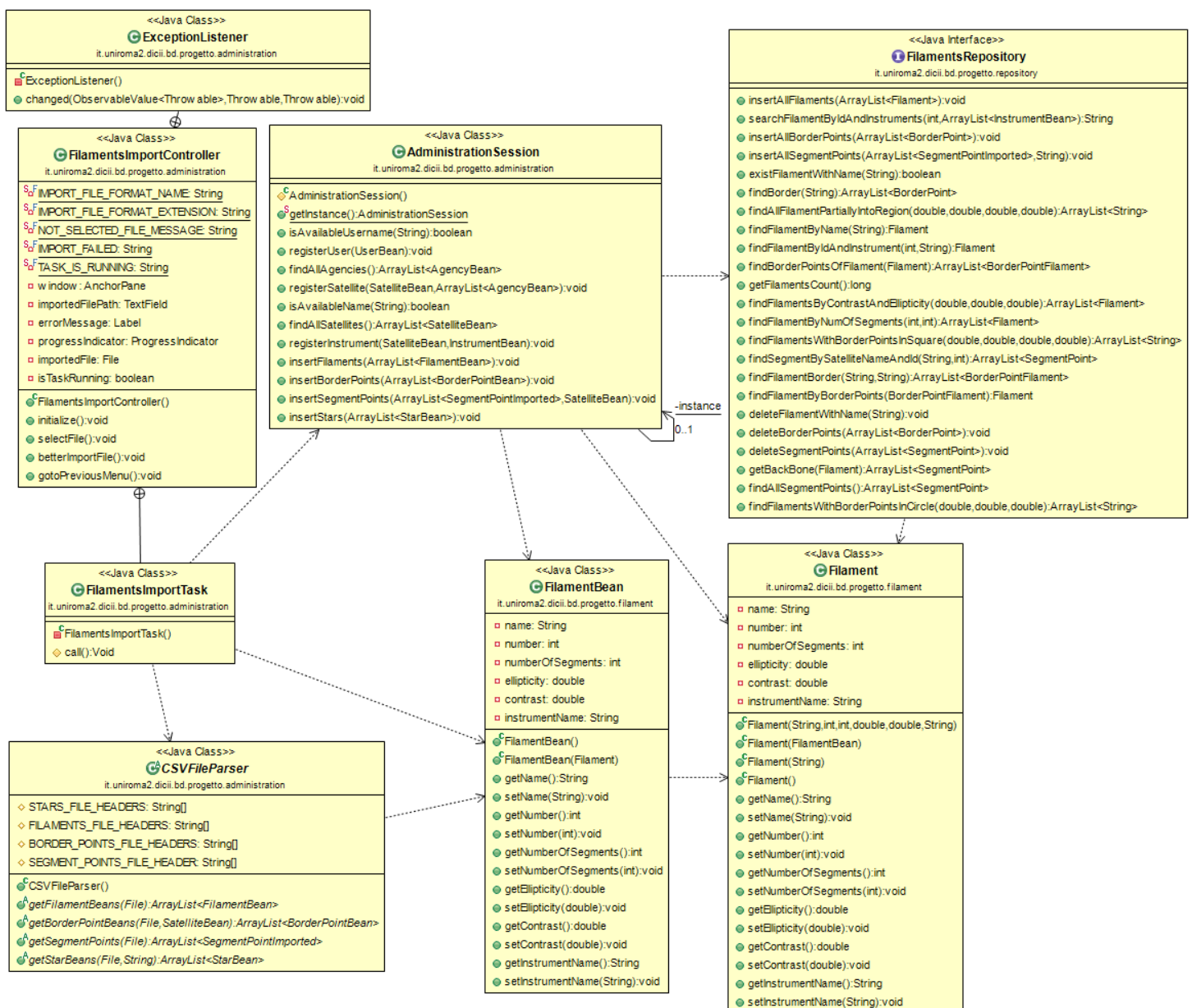
- La classe UserRegistrationController passa all'oggetto AdministrationSession l'oggetto UserBean costruito con i dati inseriti dall'utente;
- Il session richiede all'interfaccia UsersRepository il salvataggio dell'utente nello strato di persistenza.



• Importazione di un file

Il seguente class diagram illustra le classi coinvolte nell'implementazione dell'import dei filamenti, che è stato scelto come caso d'uso campione per la descrizione dell'import dei file (requisiti 3.1 e 4).

- La classe FilamentsImportController utilizza un oggetto CSVFileParser per parsare il file CSV contenente i dati dei filamenti da importare;
- L'oggetto CSVFileParser restituisce la lista dei filamenti parsati (oggetti di tipo FilamentBean);
- Il controller passa all'oggetto AdministrationSession la lista ricevuta dal parser;
- Il session richiede all'interfaccia FilamentsRepository il salvataggio dei filamenti nello strato di persistenza;
- Se i filamenti non rispettano i vincoli imposti dallo strato di persistenza, l'inserimento fallisce e viene sollevata un'eccezione (e.g. inserimento di filamento con chiave già presente nel database).



- **Inserimento di satelliti e strumenti**

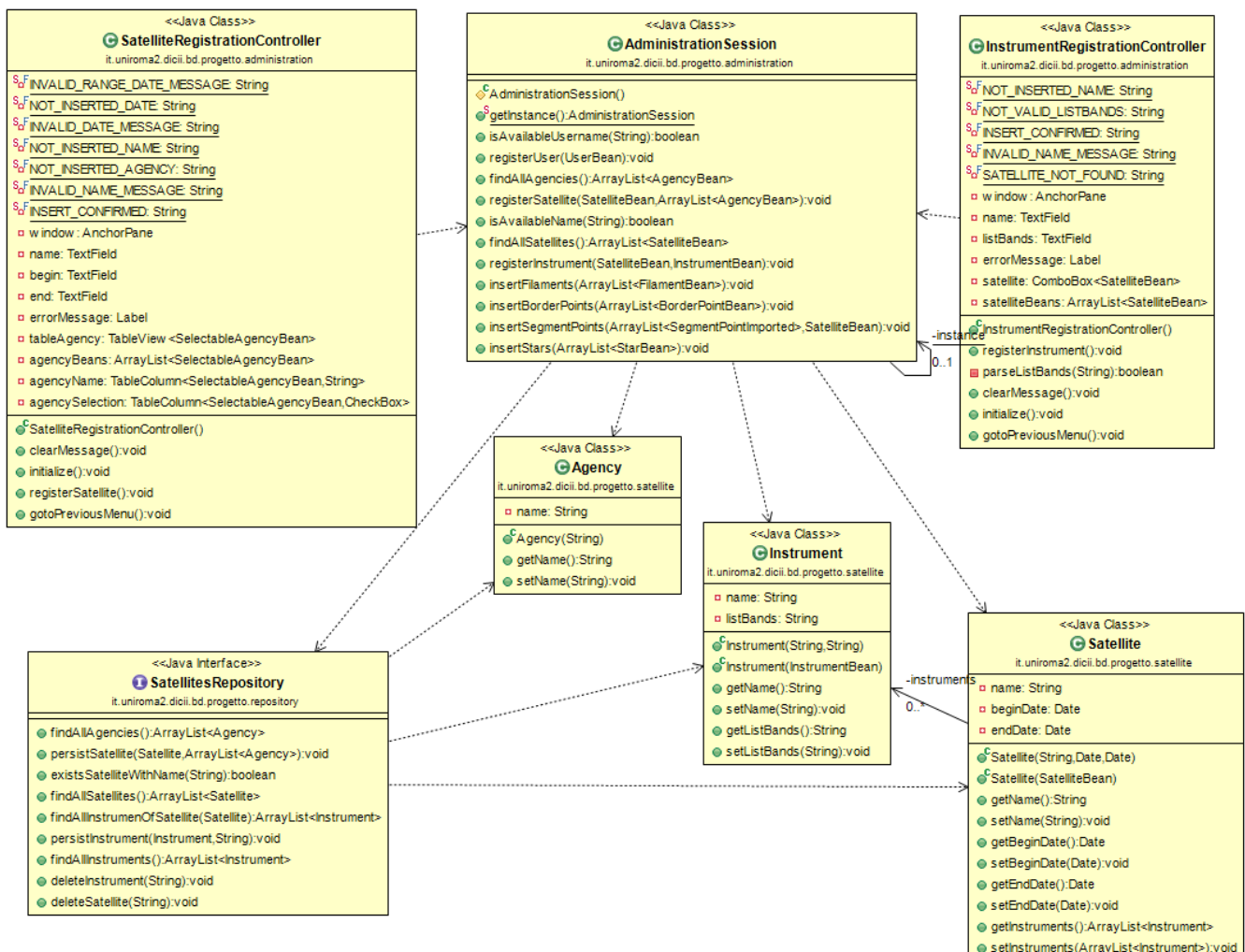
Il seguente class diagram illustra le classi coinvolte nell'implementazione dei requisiti funzionali 3.3 e 3.4.

- Inserimento satellite:

- La classe `SatelliteRegistrationController` costruisce un oggetto `SatelliteBean` contenente i dati del satellite da inserire (nome, data di inizio attività ed eventuale data di fine attività);
- Inoltre, costruisce la lista delle agenzie coinvolte nel progetto, che l'utente ha selezionato spuntando le checkbox in un'apposita tabella;
- L'oggetto `SatelliteBean` e la lista di agenzie vengono passati all'oggetto `AdministrationSession`;
- Il session richiede all'interfaccia `SatellitesRepository` il salvataggio del satellite nello strato di persistenza.

- Inserimento strumento:

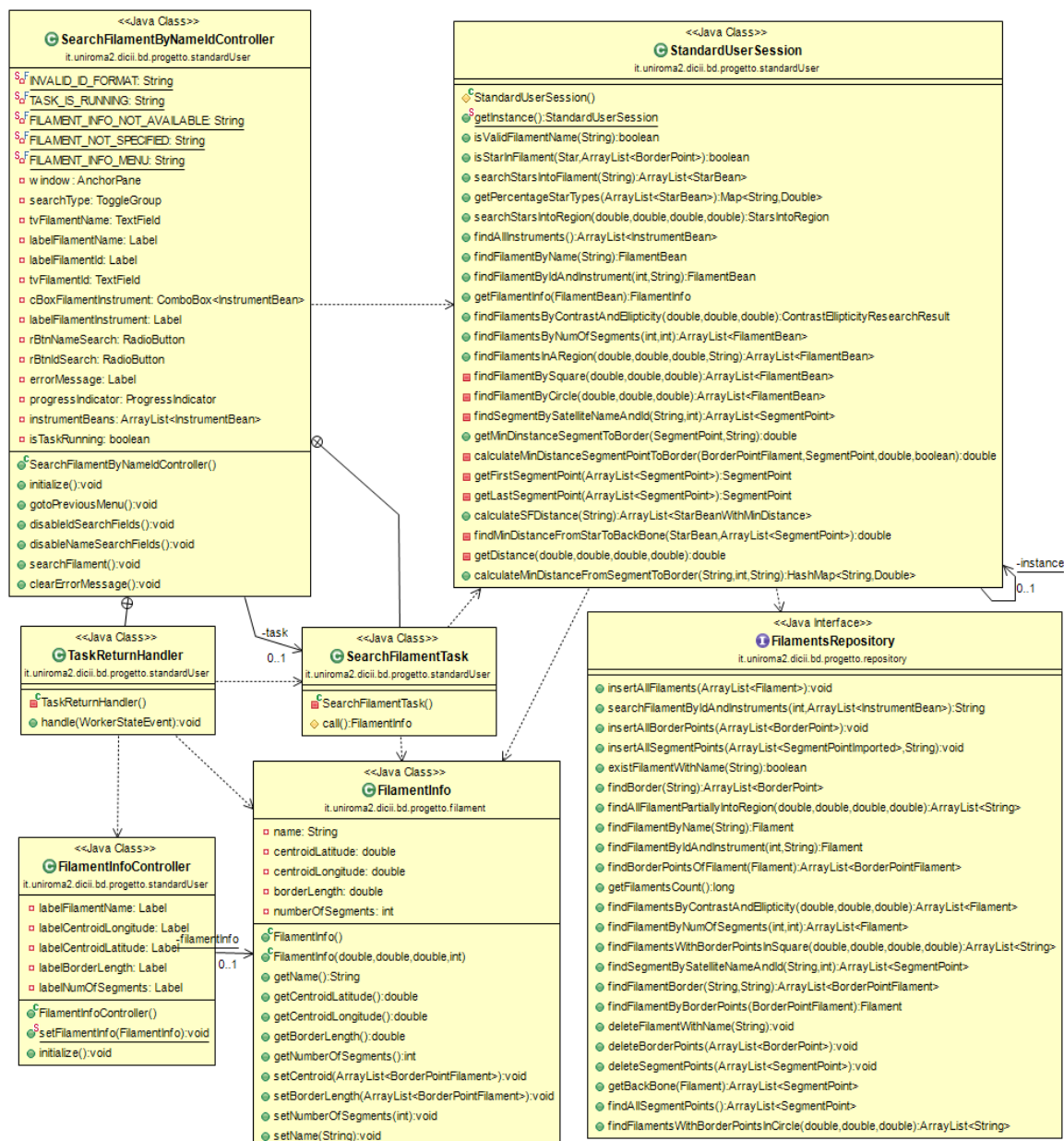
- In fase di inizializzazione la classe `InstrumentRegistrationController` popola una combo box con la lista dei satelliti presenti nello strato di persistenza;
- Il controller costruisce un oggetto `InstrumentBean` contenente i dati dello strumento da inserire (nome ed elenco bande). Inoltre, verifica che l'elenco delle bande sia stato inserito in modo sintatticamente corretto, segnalando con un messaggio d'errore eventuali problemi;
- Il controller verifica che non esista uno strumento con lo stesso nome fra quelli appartenenti al satellite selezionato, visualizzando in caso contrario un messaggio d'errore;
- Il controller passa all'oggetto `AdministrationSession` il satellite selezionato e l'oggetto `InstrumentBean`;
- Il session richiede all'interfaccia `SatellitesRepository` il salvataggio dello strumento nello strato di persistenza.



- **Recupero delle informazioni derivate di un filamento**

Il seguente class diagram illustra le classi coinvolte nell'implementazione del requisito funzionale 5.

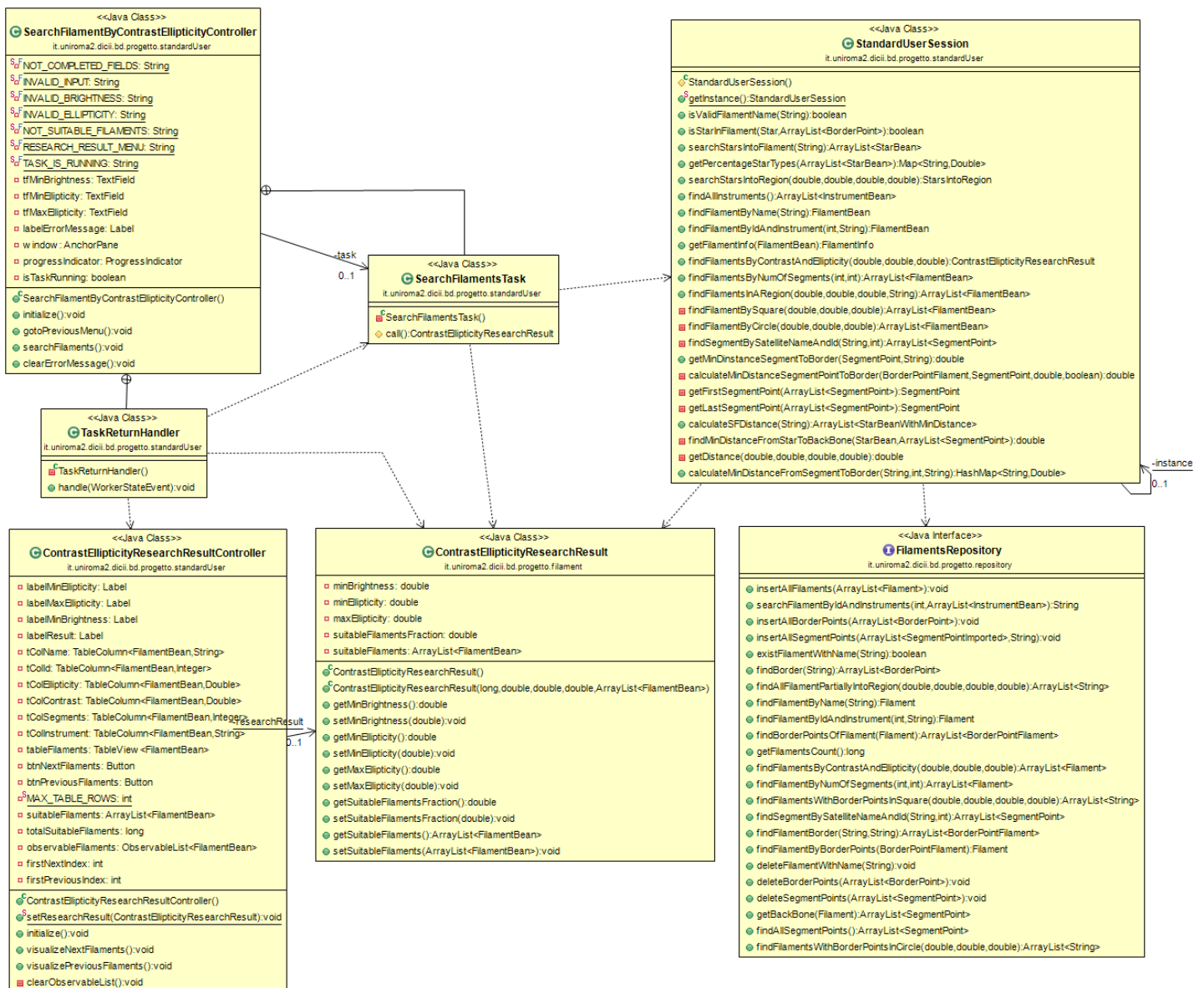
- La classe `SearchFilamentByNameIdController` richiede all'oggetto `StandardUserSession` il filamento avente il nome o l'identificativo inserito dall'utente;
- Il session utilizza l'interfaccia `FilamentsRepository` per effettuare la ricerca e ritorna al controller un oggetto `FilamentBean`;
- Se l'oggetto `FilamentBean` ricevuto è diverso da null (i.e. il filamento richiesto esiste) il controller delega al session il recupero delle sue informazioni derivate: coordinate del centroide, estensione del contorno e numero di segmenti;
- Il session ottiene da `FilamentsRepository` i punti appartenenti al contorno del filamento e con essi calcola l'estensione del bordo e le coordinate del centroide. Non è necessario invece calcolare direttamente il numero di segmenti, essendo questo uno degli attributi dell'oggetto `Filament` trovato precedentemente. Le informazioni raccolte vengono incapsulate in un apposito oggetto `FilamentInfo`, successivamente restituito al controller;
- `SearchFilamentByNameIdController` passa l'oggetto `FilamentInfo` ricevuto dal session alla classe `FilamentInfoController`, che apre una finestra contenente le informazioni derivate richieste.



- **Ricerca di un filamento per contrasto ed ellitticità**

Il seguente class diagram illustra l'implementazione del requisito funzionale 6.

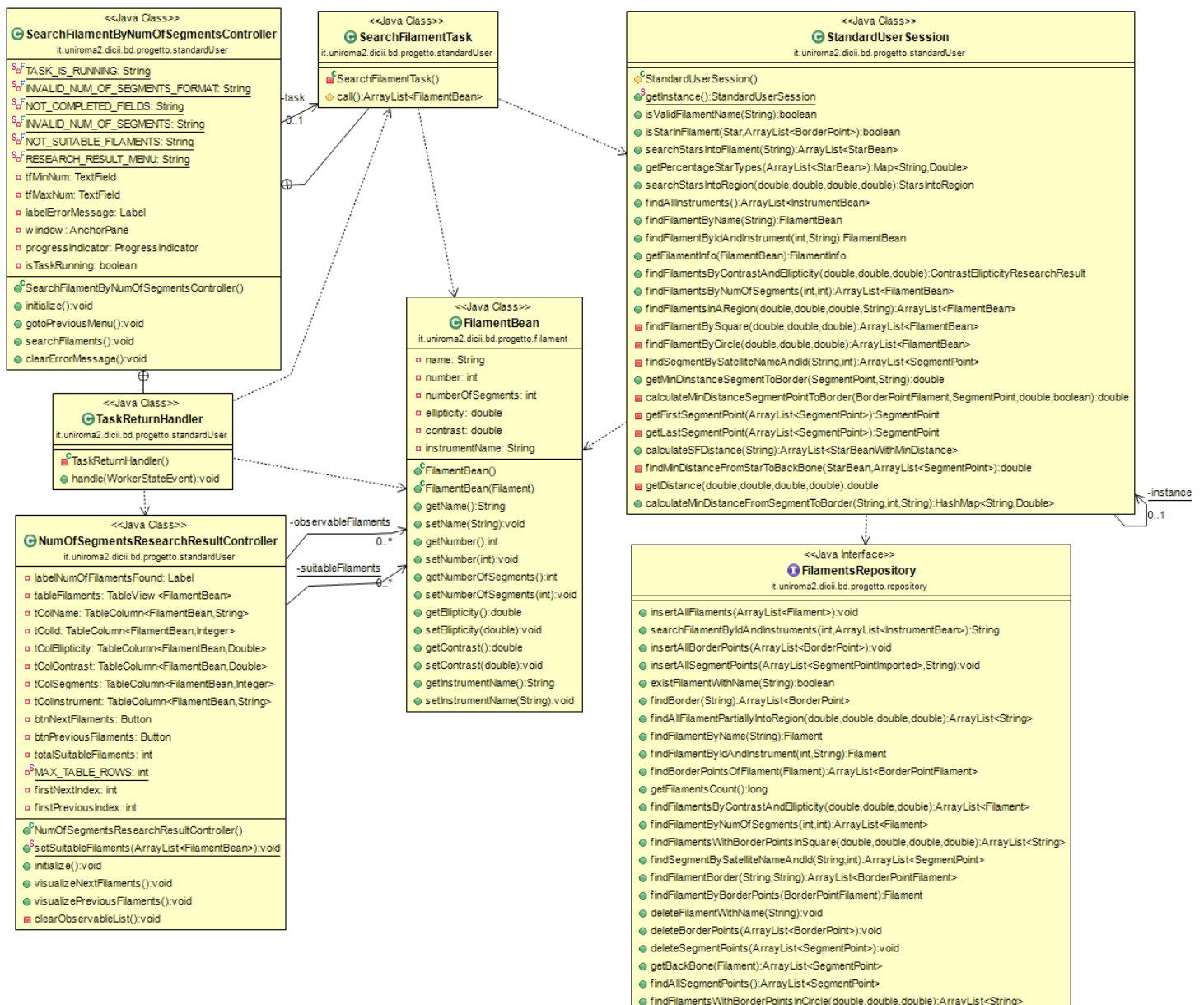
- La classe `SearchFilamentByContrastEllipticityController` parse i parametri di ricerca inseriti dall'utente: brillantezza minima (b), ellitticità minima (e) ed ellitticità massima (E);
- Il controller richiede quindi all'oggetto `StandardUserSession` di trovare i filamenti aventi una brillantezza maggiore di b e un'ellitticità compresa tra e ed E;
- Il session per prima cosa verifica che i parametri di ricerca siano semanticamente validi (sollevando in caso contrario l'eccezione `InvalidBrightnessError` oppure l'eccezione `InvalidEllipticityError`). Successivamente ottiene da `FilamentsRepository` i filamenti rispondenti ai criteri specificati dall'utente e il numero totale di filamenti memorizzati nello strato di persistenza. Con le informazioni ottenute costruisce un oggetto `ContrastEllipticityResearchResult` contenente la lista di filamenti trovati e la loro percentuale rispetto al totale. Infine, restituisce l'oggetto al controller;
- Il controller passa l'oggetto ritornato dal session all'oggetto ritornato dal session alla classe `ContrastEllipticityResearchResultController` la quale gestisce la visualizzazione dei filamenti trovati in un'apposita tabella.



- **Ricerca di un filamento per numero dei segmenti**

Il seguente class diagram illustra l'implementazione del requisito funzionale 7.

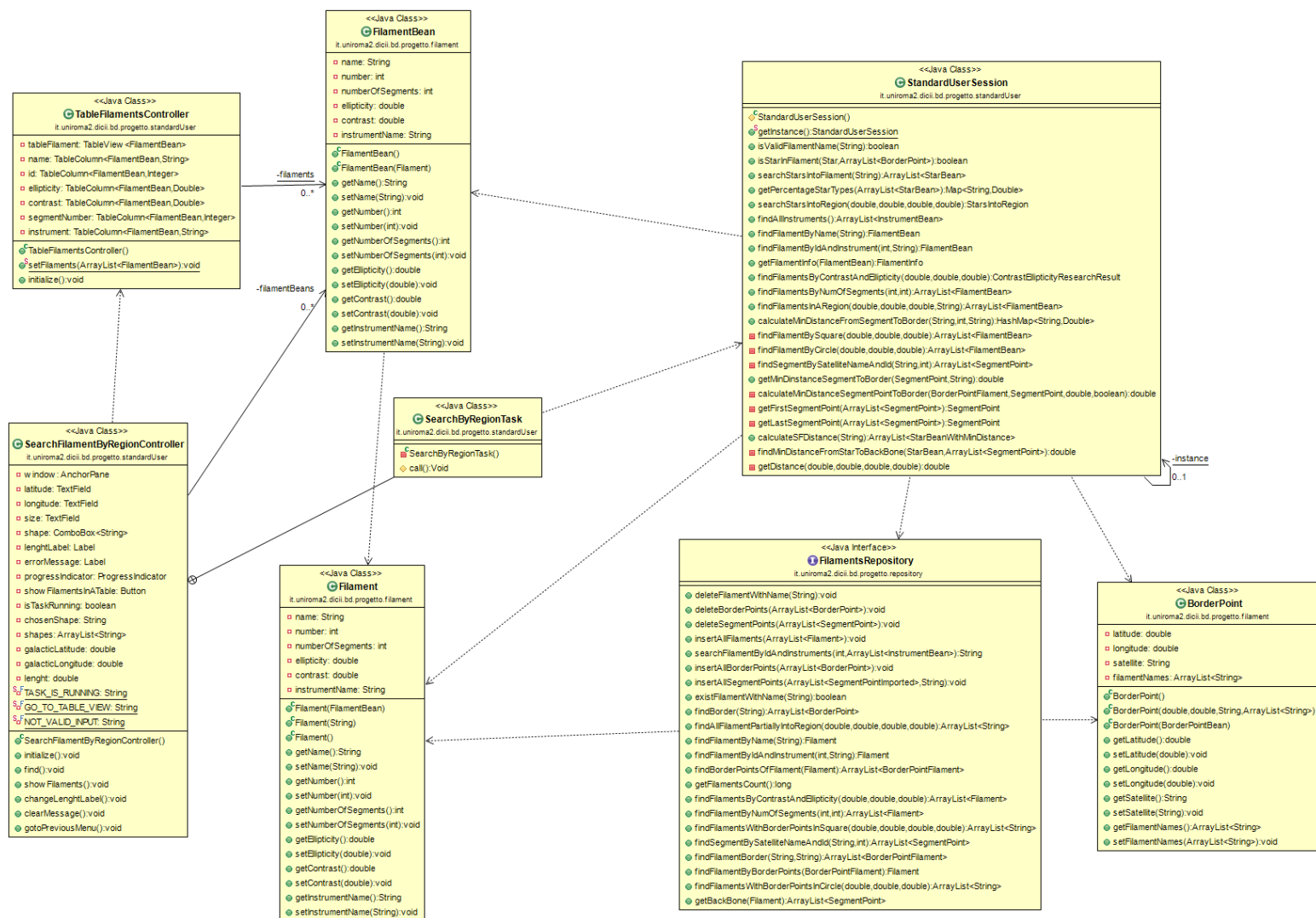
- La classe `SearchFilamentByNumOfSegmentsController` parse i parametri di ricerca inseriti dall'utente: numero minimo di segmenti (s) e numero massimo di segmenti (S);
- Il controller richiede quindi all'oggetto `StandardUserSession` di trovare i filamenti aventi un numero di segmenti compreso tra s e S;
- Il session per prima cosa verifica che i parametri di ricerca siano semanticamente validi (sollevando in caso contrario l'eccezione `InvalidNumOfSegmentsError`). Successivamente ottiene da `FilamentsRepository` i filamenti rispondenti ai criteri specificati dall'utente e li restituisce al controller;
- Il controller passa i filamenti trovati a `NumOfSegmentsResearchResultController` che li visualizza in formato tabellare.



- **Ricerca filamenti dentro una regione quadrata o circolare**

Il seguente class diagram illustra le classi coinvolte nell'implementazione del requisito funzionale 8.

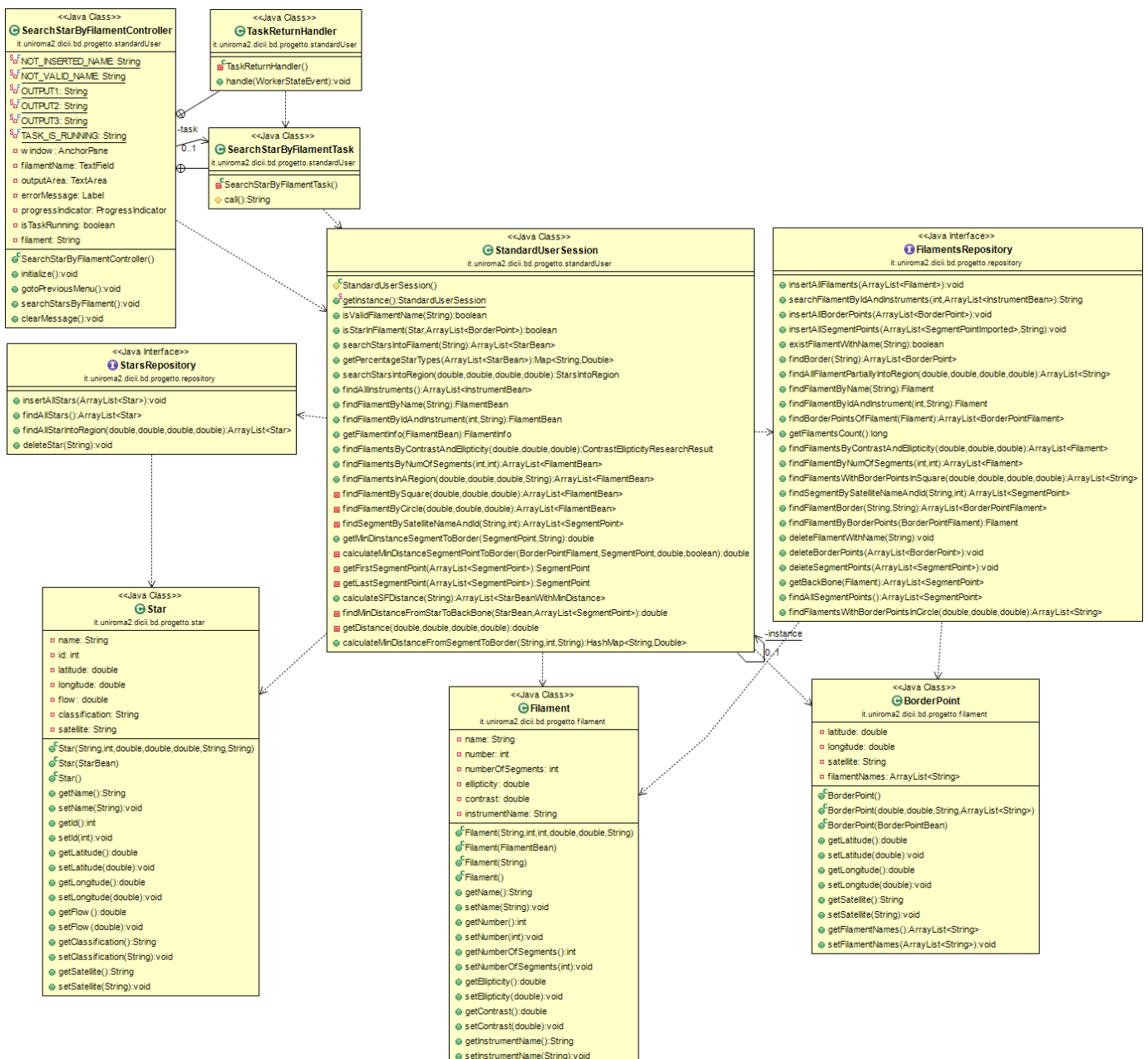
- La classe SearchByFilamentController passa allo StandardUserSession il centroide, la forma e la lunghezza del lato/raggio;
- Nel caso del quadrato, lo StandardUserSession rielabora i dati inseriti dall'utente per ricavare i vertici della regione;
- Lo StandardUserSession richiede all'interfaccia FilamentsRepository i nomi dei filamenti che non hanno punti del contorno esterni all'area d'interesse;
- Lo StandardUserSession utilizza l'interfaccia FilamentsRepository per cercare i filamenti con le caratteristiche sopra indicate passando i loro nomi;
- La classe SearchByFilamentController passa alla classe TableFilamentsController i filamenti ricevuti dallo StandardUserSession;
- TableFilamentsController mostra i filamenti all'utente.



• Ricerca stelle dentro un filamento

Il seguente class diagram illustra le classi coinvolte nell'implementazione del requisito funzionale 9.

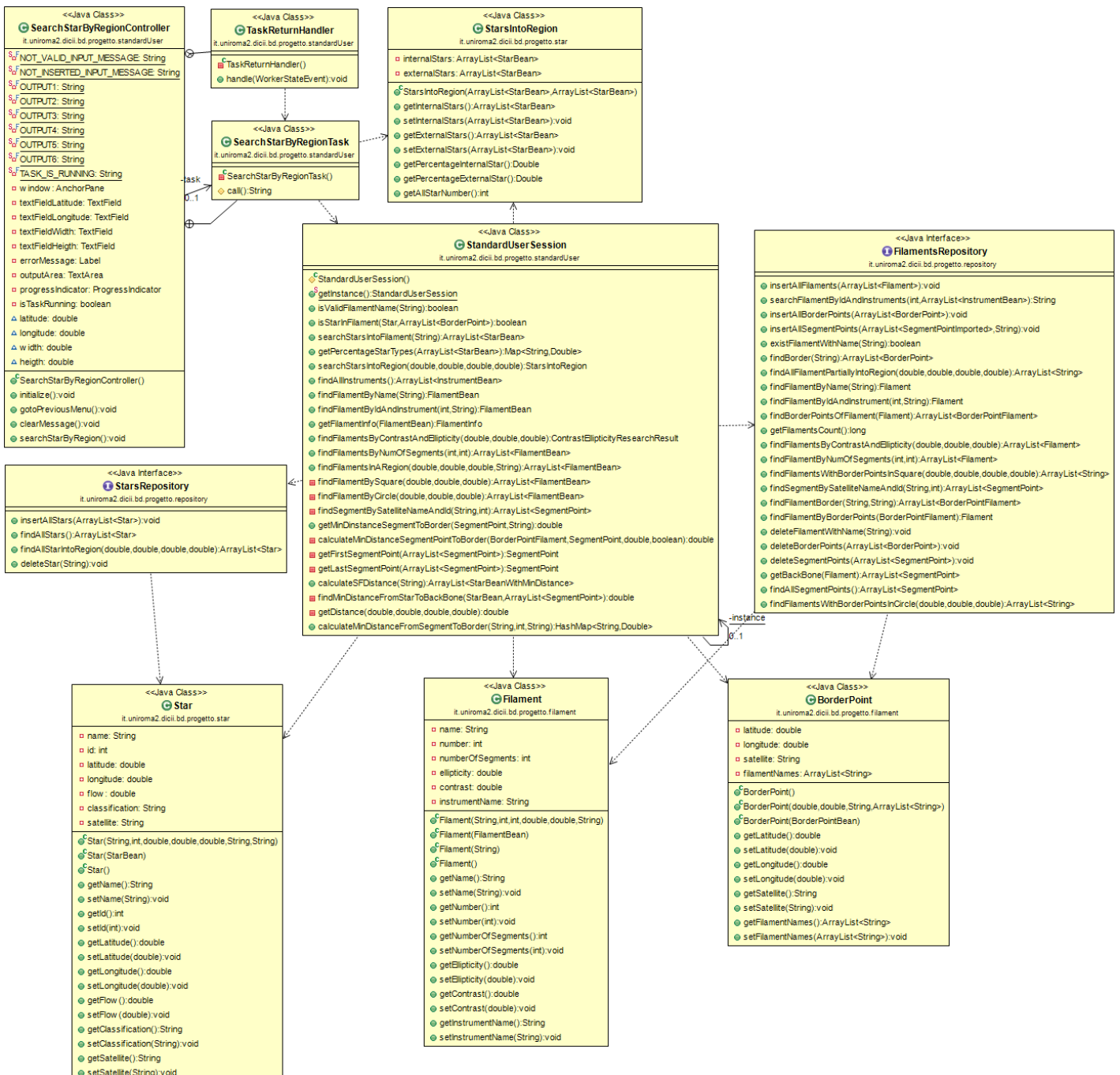
- La classe `SearchStarByFilamentController` passa allo `StandardUserSession` il nome del filamento all'interno del quale ricercare le stelle;
- Lo `StandardUserSession` richiede all'interfaccia `FilamentsRepository` i punti che costituiscono il contorno del filamento (`BorderPoint`) e all'interfaccia `StarsRepository` le stelle (`Star`) presenti in persistenza. Per ogni stella verifica l'appartenenza al filamento e, se interna ad esso, la inserisce in una lista da restituire al controller;
- Il controller può procedere con una stampa delle stelle trovate e successivamente richiede al session il calcolo delle percentuali per ogni tipo di stella;
- Il controller mostra all'utente il risultato della computazione del session.



• Ricerca stelle dentro una regione

Il seguente class diagram illustra le classi coinvolte nell'implementazione del requisito funzionale 10.

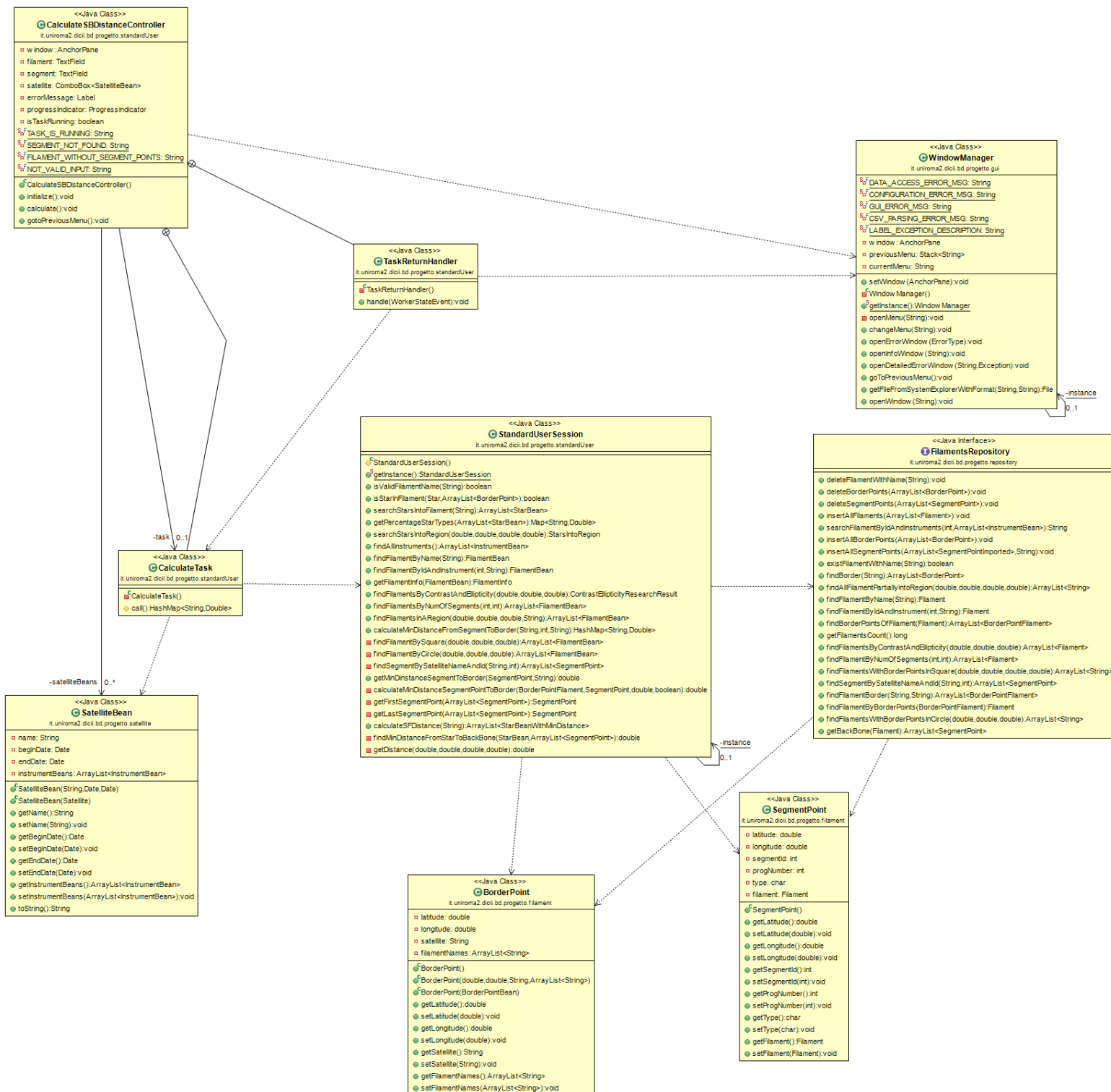
- La classe `SearchStarByRegionController` passa allo `StandardUserSession` il centroide e le dimensioni della regione all'interno della quale ricercare le stelle;
- Il session utilizza l'interfaccia `StarRepository` per ricercare tutte le stelle contenute nella regione e l'interfaccia `FilamentRepository` per ricercare il bordo di tutti i filamenti aventi almeno un punto del contorno interno alla regione;
- Per ogni stella si verifica l'appartenenza ad almeno uno dei filamenti interni (o parzialmente tali) alla regione. Sulla base di tale controllo le stelle contenute nella regione vengono suddivise in due liste corrispondenti all'elenco di stelle interne ed esterne ai filamenti;
- Per ciascuna delle due liste il session calcola la percentuale per ogni tipo di stella e ritorna il risultato di tale computazione al controller;
- Il controller mostra all'utente il numero di stelle interne ed esterne trovate nella regione e la percentuale calcolata per ogni tipo di stella.



- **Calcolo della distanza tra gli estremi di un segmento di un filamento ed il suo contorno**

Il seguente class diagram illustra le classi coinvolte nell'implementazione del requisito funzionale 11.

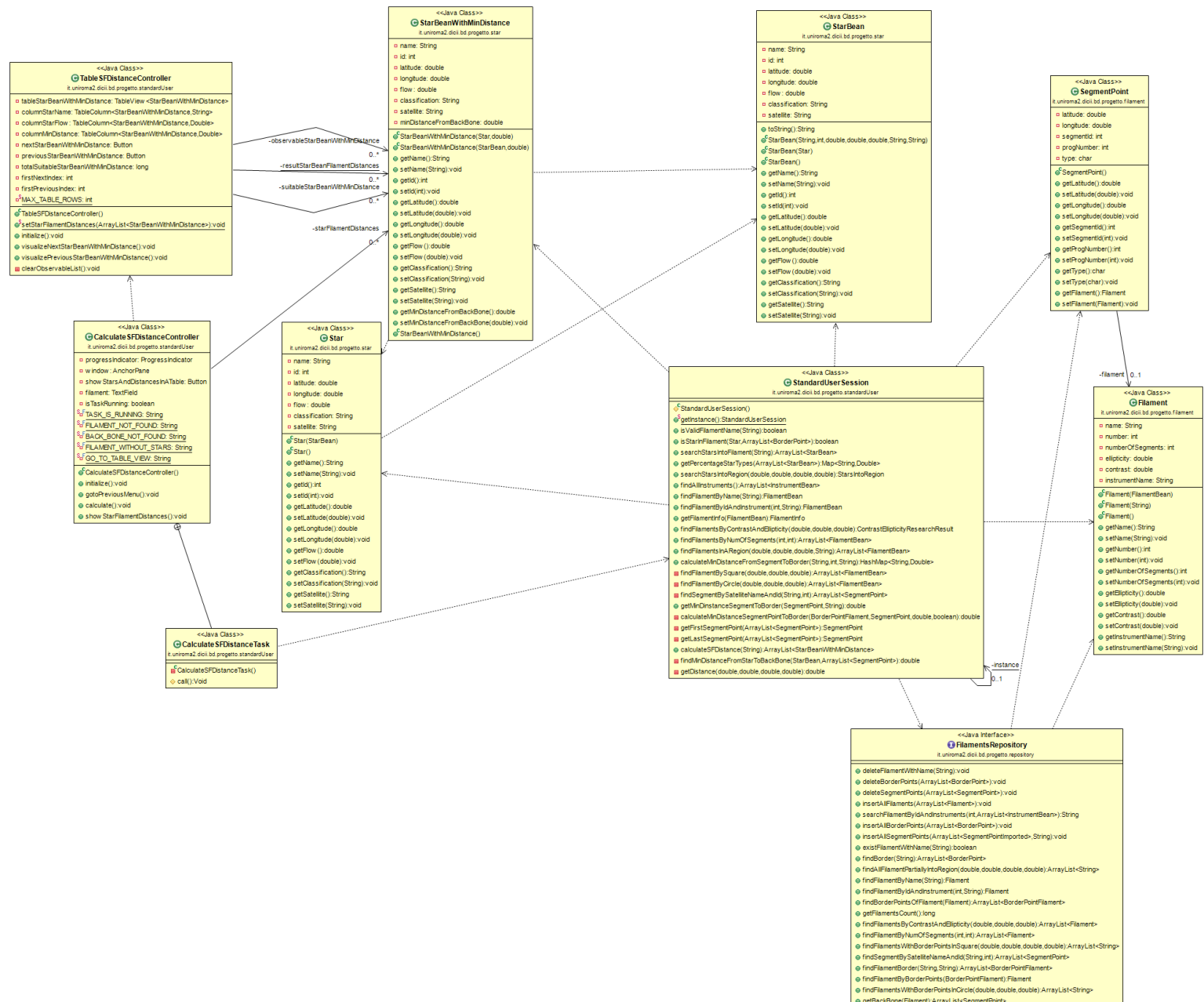
- La classe CalculateSBDistanceController passa allo StandardUserSession il nome del filamento, il nome del satellite e l'id del segmento;
- Lo StandardUserSession utilizza l'interfaccia FilamentRepository per cercare il filamento, poi ottiene tutti i punti del segmento con l'id inserito dall'utente ed i suoi estremi tramite il numero progressivo;
- Lo StandardUserSession utilizza l'interfaccia FilamentRepository per cercare i punti del contorno e calcola la distanza minima tra il più vicino di essi e gli estremi del segmento;
- La classe CalculateSBDistanceController riceve dallo StandardUserSession le due distanze minime e richiede all'oggetto WindowManager di aprire una finestra che le visualizzi.



- **Calcolo della distanza tra le stelle contenute in un filamento ed il suo scheletro**

Il seguente class diagram illustra le classi coinvolte nell'implementazione del requisito funzionale 12.

- La classe CalculateSFDistanceController passa allo StandardUserSession il nome del filamento in cui cercare le stelle;
- Lo StandardUserSession utilizza l'interfaccia FilamentRepository per cercare il filamento, poi cerca le stelle al suo interno sfruttando le funzionalità descritte nella sezione "Ricerca stelle dentro un filamento";
- Lo StandardUserSession utilizza l'interfaccia FilamentRepository per cercare lo scheletro del filamento, calcola la distanza minima tra esso ed ogni stella e istanzia un oggetto StarBeanWithMinDistance che ha come attributi gli attributi della stella e la rispettiva distanza minima dallo scheletro;
- La classe CalculateSFDistanceController riceve tutte le stelle con le rispettive distanze minime dallo scheletro e le passa alla classe TableSFDistanceController che mostra il risultato all'utente.



Testing

Il testing dell'applicazione è stato realizzato con il seguente approccio:

- È stata realizzata una copia del database vuota dedicata esclusivamente al testing
- Per ogni requisito funzionale è stata definita una classe di test **JUnit**
- Ogni classe contiene:
 - Un metodo di setup che popola il database con gli input significativi per il caso di test;
 - Uno o più metodi di test che confrontano valori attuali e valori attesi;
 - Un metodo di cleanup che rimuove dal database i dati inseriti nel setup ed eventualmente quelli inseriti durante l'esecuzione dei metodi di test.

Per i dettagli sui casi di test specifici, vedere i commenti presenti nel codice dell'applicazione.