VISCERAL Annotation Ticketing Framework

Markus Krenn
Medical University Of Vienna

1 Introduction

One goal of the VISCERAL project is to create large datasets that contain expert annotations (organ segmentations, landmarks and lesion annotations) of high quality in medical imaging data. For this purpose a ticketing framework has been developed that allows the management of different annotation types, the distribution of annotation tickets to multiple annotators and to perform a quality check of annotations to ensure consistent annotation quality across different annotators.

This documents gives an overview of the ticketing system and describes its main components. Installation guidelines and the sources are publicly available at `https://github.com/Visceral-Project/annotationTicketingFramework`. In the following sections we describe the typical life cycle of an annotation ticket and all components of the ticketing framework in detail.

2 Annotation ticket life cycle

The typical life cycle of an annotation ticket within the VISCERAL project can be outlined as follows:

1. In a first step a ticket is created. An annotation ticket is defined by the volume to be annotated, the annotation type (segmentation, lesion or landmark) and the annotator.

2. The annotator receives the ticket through a web-interface, performs annotation and submits the annotated data through the web-interface.

3. Depending on the type of annotation, an automated quality check is performed to detect common annotation errors, such as empty label volumes, incorrect file extentions or wrong naming of landmarks.

4. If the annotation passes the automated quality check the ticket is assigned to a Quality Check (QC) annotator, otherwise it is reassigned for annotation.

5. The QC annotator receives the QC ticket trough the web-interface, performs the QC and submits the QC result (including textual feedback if the QC is negative) through the web-interface.

6. The ticket reaches its final state if the QC is positive, otherwise the annotator receives textual feedback and the ticket is reassigned for annotation.

3 The VISCERAL Ticketing Framework

The VISCERAL ticketing framework is designed to monitor and manage the full life cycle of an annotation ticket, to provide an interface for annotators and QC team members for ticket submission and consists of three main components:

1. **Ticketing Data base:** A MySQL data base which stores information of volumes to annotate, annotators, annotation types and tickets and their states (pending, submitted, QC passed, QC failed).

2. **Backend:** A backend implemented in Matlab, to manage volumes, annotators, ticket types and annotation tickets. The backend is used to distribute tickets, to perform automated quality check and to distribute QC tickets of submitted annotations.

3. **Frontend:** A web-interface that is used by annotators and QC team members to receive their assigned tickets and to submit annotations and QC results.

Figure 1 provides an overview of the ticketing system implemented within the VISCERAL project.

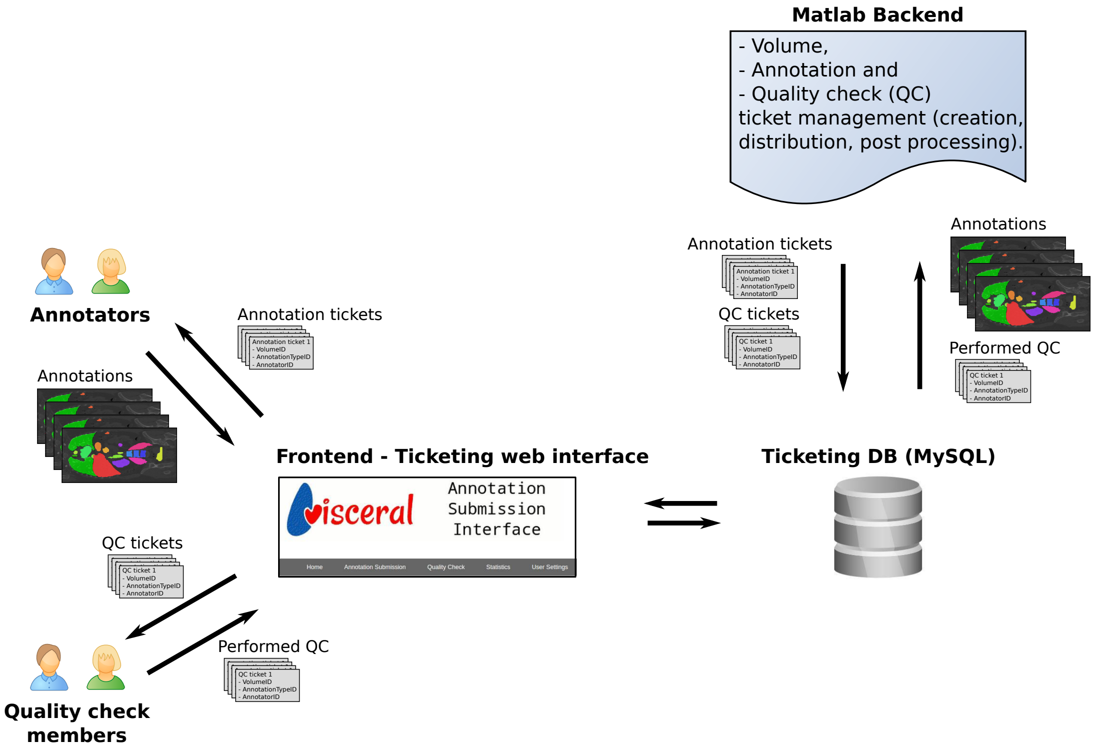


Figure 1: Work flow overview of the VISCERAL Ticketing Framework.

Ticketing System Data Base The Data Base (DB) of the ticketing system is created by SQL scripts provided in the ticketing repository. All relevant information is stored in five DB tables:

- **Annotator:** Identified by an *AnnotatorID*, holds next to contact information, name and password (for login) a flag indicating if the annotator is currently available and an additional flag if the annotator is considered as QC team member.

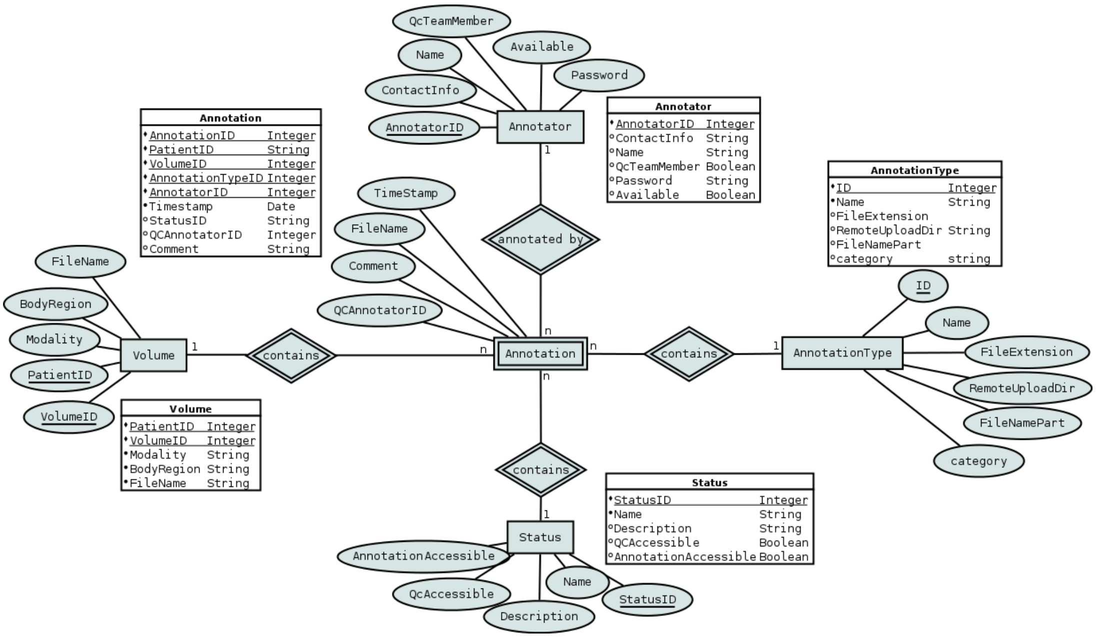
- **Volume:** A volume is identified by its *PatientID* and *VolumeID*. Additionally the modality, bodyregion and the filename is stored.

- **AnnotationType:** Entries in this table define which tpyes of annotations can be managed by the ticketing system. Each entry is identified by its *AnnotationTypeID*. Additionally the name, the file extension of the submitted files, the remote upload directory, the category (segmentation, landmarks,...) and an optional string describing the files prefix are stored. Exemplary entries of this table are created within the given SQL scripts.

- **Status:** Defines all states a ticket can have during its life cycle. A status is identified by its *StatusID* and stores its name and description as well as to which type of annotators (QC and normal annotators) the status option is available in the ticketing web interface. Default entries of this table are created within the given SQL scripts.

- **Annotation:** This table represents an annotation ticket. An annotation entry is identified by its *PatientID* and *VolumeID*, the *AnnotationTypeID*, the *StatusID* and the *AnnotatorID* of the annotator a ticket is assigned to. Additionally the filename, a timestamp, the ID of the annotator who performs the QC of the ticket and a QC comment is stored for each ticket.

Figure 2 illustrates the ER diagram of the resulting data base.



Annotation	
♦AnnotationID	Integer
♦PatientID	String
♦VolumeID	Integer
♦AnnotationTypeID	Integer
♦AnnotatorID	Integer
•Timestamp	Date
°StatusID	String
°QCAnnotatorID	Integer
°Comment	String

Annotator	
♦AnnotatorID	Integer
°ContactInfo	String
°Name	String
°QcTeamMember	Boolean
°Password	String
°Available	Boolean

AnnotationType	
♦ID	Integer
•Name	String
°FileExtension	
°RemoteUploadDir	String
°FileNamePart	
°category	string

Volume	
♦PatientID	Integer
♦VolumeID	Integer
•Modality	String
•BodyRegion	String
•FileName	String

Status	
♦StatusID	Integer
•Name	String
°Description	String
°QCAccessible	Boolean
°AnnotationAccessible	Boolean

Figure 2: Entity Relation diagram of the VISCERAL Ticketing System DB.

Backend The backend of the framework is implemented in Matlab. It provides getter and setter functions for all tables of the database and can thus be used to manage

- the distribution of annotation tickets
- different annotators and their availability
- subjects to be annotated (volumes, images,...)
- different annotation types, such as segmentations or landmarks and
- to perform automated quality checks.

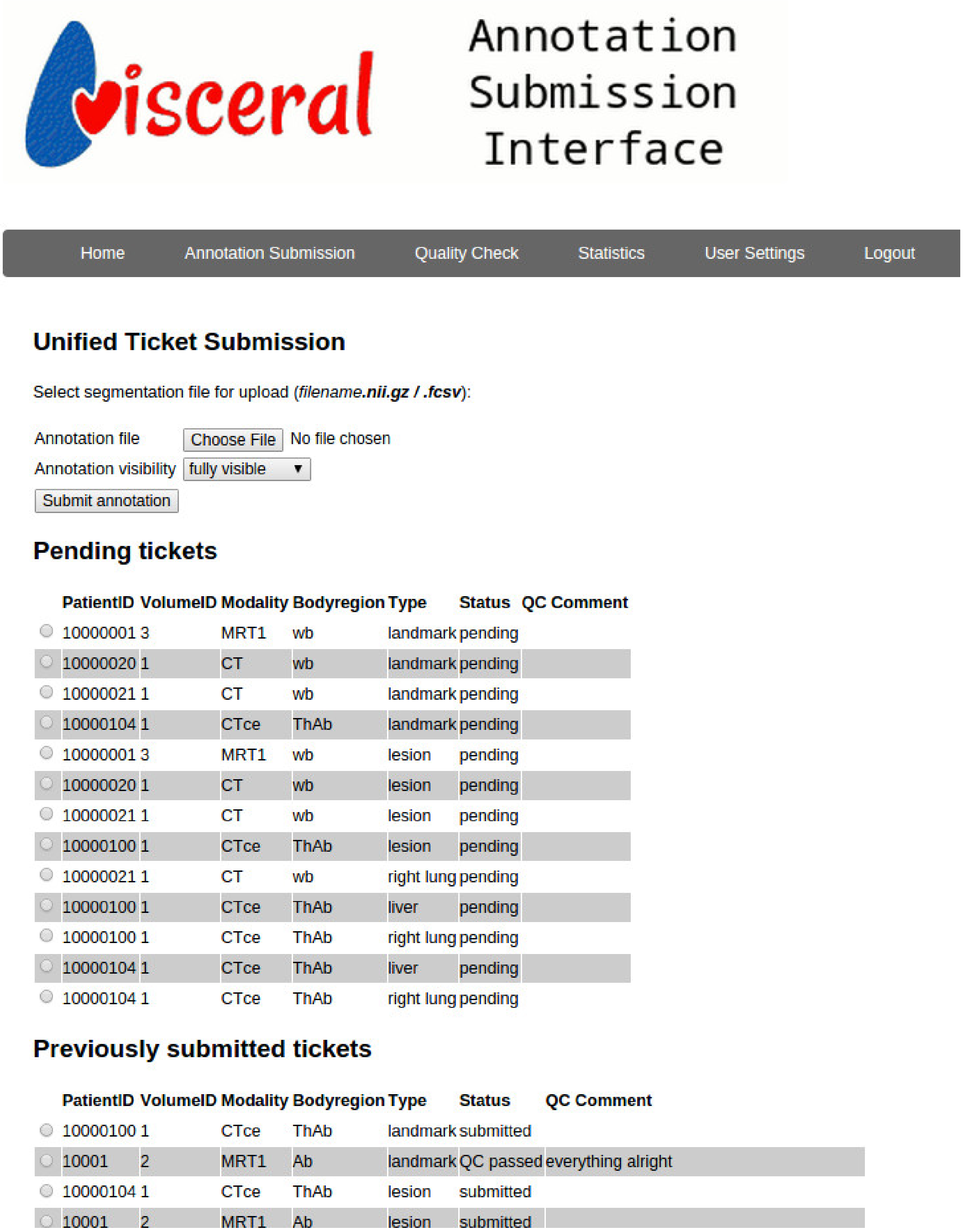
The sources in the repository include a tutorial script that shows the usage of the data base getter and setter functions provided.

Frontend The VISCERAL ticketing frontend serves as interface between annotators, QC team members and the data base. Each annotator and QC team member. Each annotator and QC team member is able to login to the ticketing page. After login, the page "Annotation Submission" lists all annotation tickets that are assigned to an annotator. Here an annotator can choose a ticket and upload his annotation. The annotation file is then checked on the correct file extension and uploaded to a SFTP server. Figure 3 shows a screen shot of the Annotation Submission page.

After a ticket has been annotated it is assigned to a QC team member to be checked on its quality. A quality check member can see all his tickets assigned on the page "Quality Check". If the annotation is of poor quality, the QC team member sets the status of the ticket to *QC failed* and gives a short comment to provide feedback for the original annotator. Figure 4 shows a screen shot of the quality check page.

Additionally the sub page statistics shows the number of distributed, pending, annotated, qc failed and qc passed tickets for each annotation type. A screen this sub page is shown in Figure 5.

A demo version of the web-interface including exemplary test data is available at `https://www.cir.meduniwien.ac.at/visceral/tickets/demo/tickets/login.php`. Login information is given on request from **Markus Krenn** for security reasons.



Unified Ticket Submission

Select segmentation file for upload (*filename.**nii.gz / .fcsv***):

Annotation file Choose File No file chosen

Annotation visibility fully visible ▼

Submit annotation

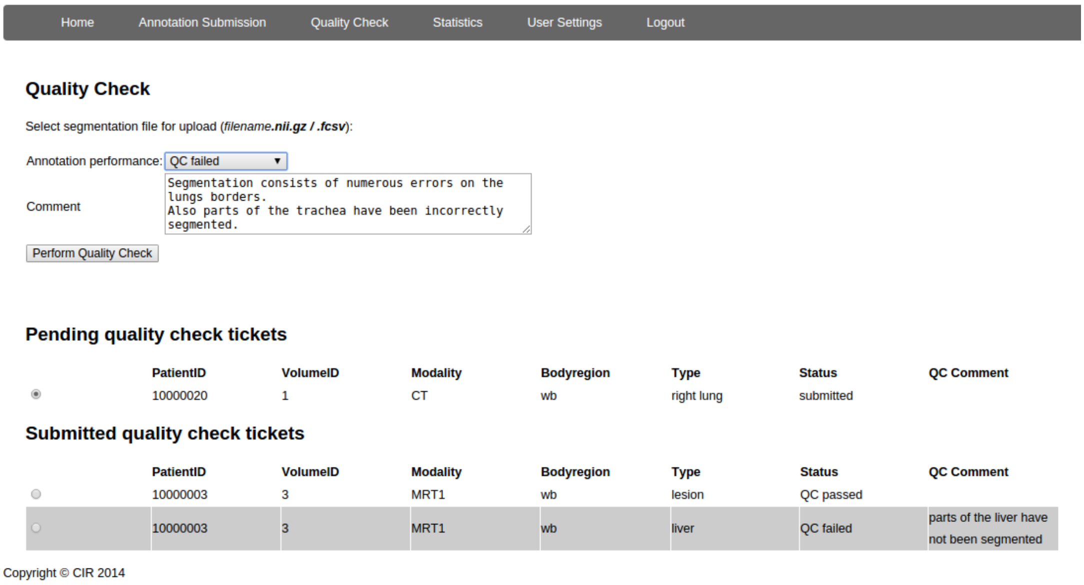
Pending tickets

	PatientID	VolumeID	Modality	Bodyregion	Type	Status	QC Comment
○	10000001	3	MRT1	wb	landmark	pending	
○	10000020	1	CT	wb	landmark	pending	
○	10000021	1	CT	wb	landmark	pending	
○	10000104	1	CTce	ThAb	landmark	pending	
○	10000001	3	MRT1	wb	lesion	pending	
○	10000020	1	CT	wb	lesion	pending	
○	10000021	1	CT	wb	lesion	pending	
○	10000100	1	CTce	ThAb	lesion	pending	
○	10000021	1	CT	wb	right lung	pending	
○	10000100	1	CTce	ThAb	liver	pending	
○	10000100	1	CTce	ThAb	right lung	pending	
○	10000104	1	CTce	ThAb	liver	pending	
○	10000104	1	CTce	ThAb	right lung	pending	

Previously submitted tickets

	PatientID	VolumeID	Modality	Bodyregion	Type	Status	QC Comment
○	10000100	1	CTce	ThAb	landmark	submitted	
○	10001	2	MRT1	Ab	landmark	QC passed	everything alright
○	10000104	1	CTce	ThAb	lesion	submitted	
○	10001	2	MRT1	Ab	lesion	submitted	

Figure 3: Annotation submission of the ticketing web-interface.



Quality Check

Select segmentation file for upload (*filename****.nii.gz / .fcsv***):

Annotation performance: QC failed ▾

Comment

```
Segmentation consists of numerous errors on the
lungs borders.
Also parts of the trachea have been incorrectly
segmented.
```

Pending quality check tickets

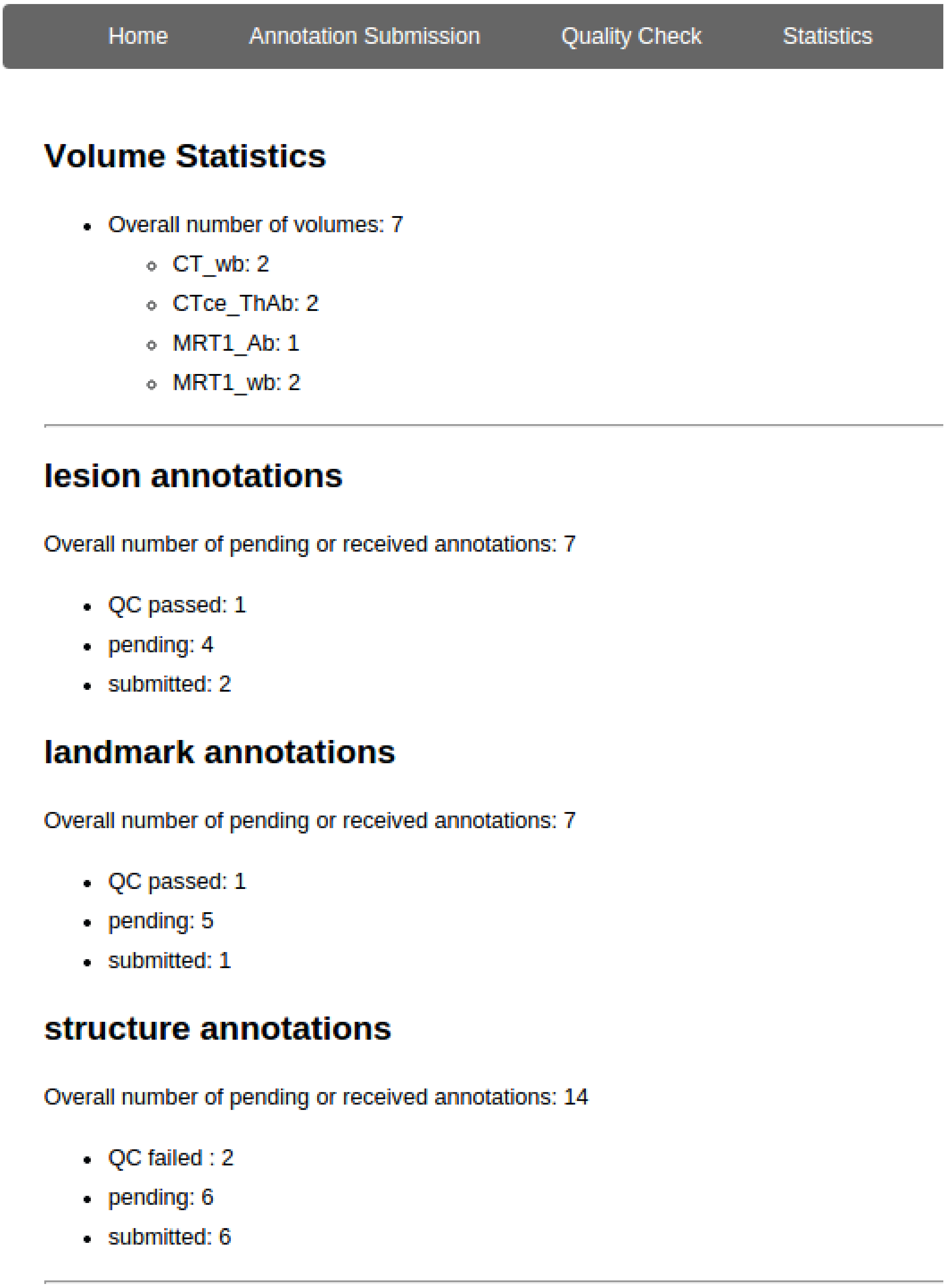
	PatientID	VolumeID	Modality	Bodyregion	Type	Status	QC Comment
●	10000020	1	CT	wb	right lung	submitted	

Submitted quality check tickets

	PatientID	VolumeID	Modality	Bodyregion	Type	Status	QC Comment
○	10000003	3	MRT1	wb	lesion	QC passed	
○	10000003	3	MRT1	wb	liver	QC failed	parts of the liver have not been segmented

Copyright © CIR 2014

Figure 4: Quality check page of the ticketing web-interface.



Volume Statistics

- Overall number of volumes: 7
  - CT_wb: 2
  - CTce_ThAb: 2
  - MRT1_Ab: 1
  - MRT1_wb: 2

lesion annotations

Overall number of pending or received annotations: 7

- QC passed: 1
- pending: 4
- submitted: 2

landmark annotations

Overall number of pending or received annotations: 7

- QC passed: 1
- pending: 5
- submitted: 1

structure annotations

Overall number of pending or received annotations: 14

- QC failed : 2
- pending: 6
- submitted: 6

Figure 5: Statistic page of the ticketing web-interface.