ECE 1896 : SENIOR DESIGN

DESIGN SPECIFICATIONS

TRACT RATER APPLICATION

UNIVERSITY OF PITTSBURGH SCHNEIDER RESEARCH LAB

Written By: Deirdre Sweeney
Advised By: Walt Schneider
Date: February 25, 2015

Problem Definition and Background

The University of Pittsburgh Schneider Research Lab is a neuro-imaging research lab within the University of Pittsburgh. They are in need of a graphical user interface to enable researcher to input qualitative metrics on advanced diffusion MRI imaging. The current problem is that the research assistants must complete and save their analysis in a machine readable document by hand which is too time consuming and error-prone. This project is to create a Tract Rater Application that will be a graphical interface to enter data and save the data in a machine readable document. The Tract Rater Application will be utilized by the University of Pittsburgh Schneider Research Lab. The Users of the Tract Rater Application are scientists and research assistants within the University of Pittsburgh Schneider Research Lab. The Tract Rater Application will potentially be used by clients outside of the University of Pittsburgh Schneider Research Lab in the future.

The University of Pittsburgh Schneider Research Lab has created a process called High-Definition Fiber Tractography (HDFT). “ High-definition Fiber Tracking is a novel combination of processing, reconstruction, and tractography methods that can track white matter fibers from cortex, through complex fiber crossings, to cortical and subcortical targets with subvoxel resolution.” [2]. This HDFT approach “provides an accurate reconstruction of white matter fiber tracts with unprecedented detail.” [2]. HDFT uses diffusion magnetic resonance imaging (MRI) to scan a brain. [3]. Because diffusion MRI is a three-dimensional process, there is plenty of data that is collected from these brain scans. At the University of Pittsburgh Schneider Research Lab, scientists and research assistants take the raw diffusion MRI brain scans, analyze all of the required data that is necessary for an HDFT report and hand write all of this data in a machine readable document (XML) to store the information in a database and to export the information to a mobile application. The process of analyzing all of the required data and hand writing it in a machine readable document is very tedious, time consuming, and prone to error. The object of the Tract Rater Application is to streamline this process to make it more convenient, intuitive, and reliable.

The Tract Rater Application will provide a user-friendly graphical interface to enter all required and optional information about a brain subject. It will then collect all of the data and save a machine readable document herein called the XML Rater Document. This XML Rater Document will be pushed through the pipeline process. The Tract Rater Application will allow the user to enter information about the overall brain scan and about each specific tract within the brain. The Tract Rater Application will take about 4 months to code and test and it will meet all of the requirements and expectations discussed throughout this paper.

Preliminary Work and Design Approaches

As a graphical interface application, there are parts of the Tract Rater Application that will be universal throughout each method that was considered. The primary universal property is that the software design of the Tract Rater Application will follow a Model-View-Controller architecture. A Model-View-Controller architecture divides the software application into three interconnected parts. As shown in Figure 1, the Model-View-Controller model is used to effectively communicate with each other and with the user.

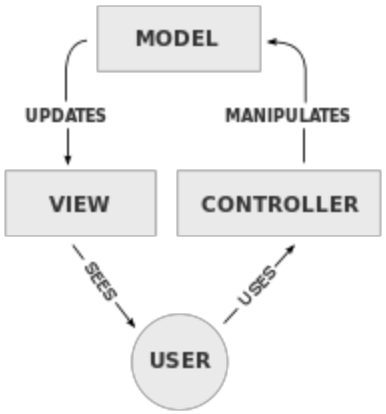


Figure 1: Model-View-Controller [9]

The Controller sends commands and updates to the Model. The Model notifies the View and Controller to changes of state. The View takes information from the Model and displays it for the User. This is the underlying architecture for the entire project.

There were three different design approaches that were researched, analyzed, and prototyped for the Tract Rater Application. These methods are described below along with pros, cons, and a description of the prototypes for each given method.

Method 1: Google Web Toolkit (GWT)

Google Web Toolkit is a development toolkit that is used to build and optimize complex browser-based applications. It relies on three different development tools: [8]

1. GWT SDK, which contains the Java API libraries, compiler, and development server.
2. An Eclipse Plugin that provides IDE support for web project.
3. A Google Chrome plugin that provides development support within the web browser.

Pros of GWT:

- All code is written in Java and converted to Javascript through the compiler. This is nice because I have a lot of experience in writing Java and minimal experience writing Javascript.

Cons of GWT:

- The Google Chrome plugin is not supported in Linux.
- All code for GWT must be written and compiled through the Eclipse IDE. This means that every computer that runs the Tract Rater Application must also have Eclipse installed.
- Web server is required.

Prototype:

This prototype was a simple Login Manager. It allowed you to enter a username and password. When you clicked “Sign In”, a pop-up box would appear that displayed your username and password. The prototype application is shown in Figure 2.

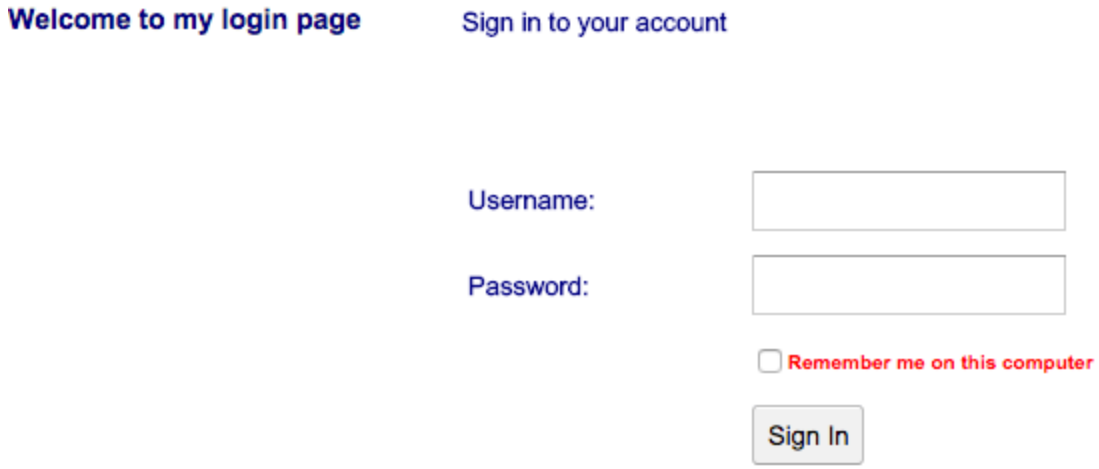


Figure 2: Initial Prototype for Google Web Toolkit - Login Manager

Method 2: AngularJS

AngularJS is a popular Javascript framework that is comparable to JQuery. It can be completely client-side or have a client / server architecture. It is dependent on Node.js. [4]

Pros of AngularJS:

- There are plenty of tools to provide ease and support for AngularJS applications. These include Yeoman tools, which provide various support for scaffolding, installing dependencies, and building/compiling/serving the application. [6]
- I have access to an online course devoted completely to the AngularJS framework.
- Uses Bootstrap.css which is what the Design Images are based on.

Cons of AngularJS:

- I have minimal experience in writing Javascript and no experience with the AngularJS framework.
- Web server is recommended, but not required.

Prototype:

This prototype was a simple task list. You could add tasks, delete tasks, and reorder tasks. The tasks were also persistent. The prototype application is shown in Figure 3.

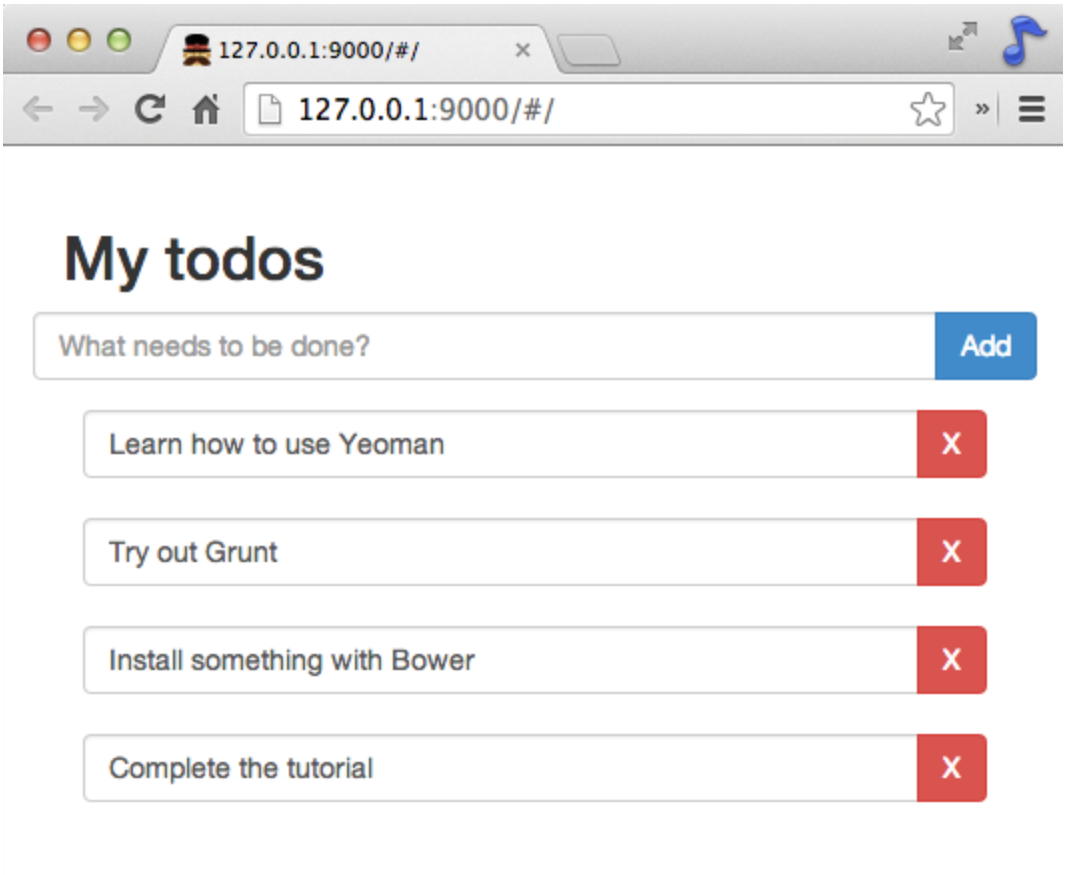


Figure 3: Initial Prototype for AngularJS - Todo List

Method 3: PyQt

PyQt is a python module that binds to the Qt cross-platform UI and application framework. It is dependent on Python and on Qt Designer. [7]

Pros of PyQt:

- PyQt4 is already installed on many of the University of Pittsburgh Schneider Research Lab computers.
- I have experience in writing Python.
- No web server is needed.

Cons of PyQt:

- PyQt4 will be unsupported at the end of 2015.
- I do not have experience with using the PyQt module.

Prototype:

This prototype was not a tutorial. I created an offline application that allowed the user to type anything, hit Enter, and see the text displayed in the text area. It also allowed you to pick a file and see the filepath of the file displayed in the text area. An example of the application is displayed in Figure 4.

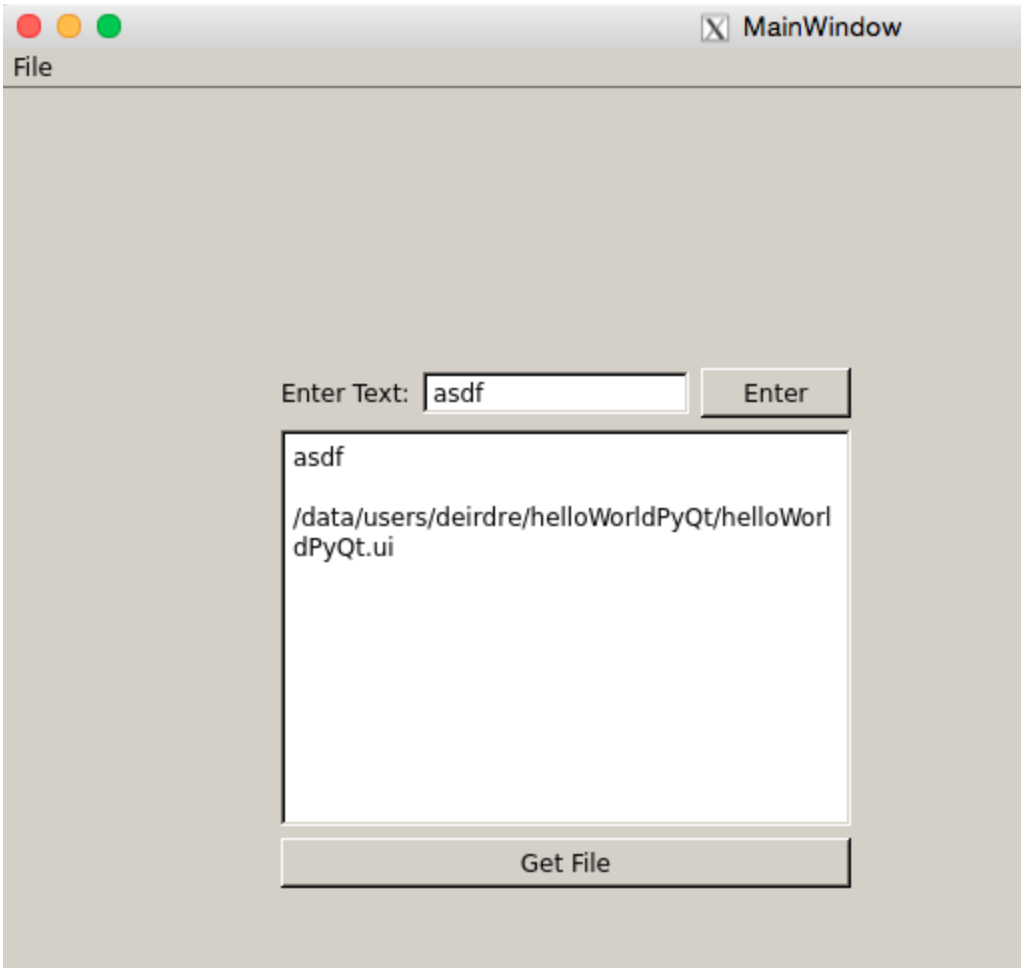


Figure 4: Initial Prototype of PyQt

Decision Matrix

The decision matrix below describes the major categories that were involved in the decision making process. Each category is marked with a number of points. These points determine the priority of the category towards the overall decision. The categories are as follows:

Development and Environment	(5 points)
Testing	(4 points)
Deployment and Platform Feasibility	(4 points)
Maintenance	(3 points)
Personal Preference	(3 points)

For each different potential method, the categories are evaluated as follows:

High	(3 points)
Medium	(2 points)
Low	(1 point)

The points from the method decision are multiplied by the points for each category to create the total number of points. The lower the total, the better the method. This total provides a valid comparison between the three methods.

	Development and Environment (5)	Testing (4)	Deployment and Platform Feasibility (4)	Maintainance (3)	Personal Preference (3)	Total
GWT	**Medium** - Dependent on Eclipse - No longer supported by Google - Plenty of documentation for GWT API	**Low** - JUnit - Plenty of Java testing frameworks available	**High** - Must be deployed through Eclipse. - Google Development Plugin not available on Linux.	**High** - No longer be supported in upcoming years	**High** - GWT was difficult to prototype in and is not very useful in the future.	**44**
Angular JS	**Low** - Lots of support from Yeoman - Robust AngularJS API	**Medium** - Never used Jasmine test framework or Karma test runner, but is supposed to be very robust.	**Low** - Can be a standalone application (completely client-side) - Can use a simple Apache or Node.js server	**Low** - AngularJS is expanding. - Supported worldwide.	**Low** - I am very interested in learning the AngularJS framework and believe it will be valuable in the future.	**23**
PyQt	**Medium** - There will be a version update at the end of the year.	**High** - Not much support for testing PyQt framework.	**Low** - No server needed. - Many machines already have PyQt installed.	**Medium** - PyQt4 will be unsupported starting in 2016.	**Medium** - PyQt is an interesting framework but not as useful as AngularJS when moving forward.	**38**

There are 3 major components of this project that were provided by the University of Pittsburgh Schneider Research Lab Technical Team. These components are as follows:

1. *XML Rater Document*

   The XML Rater Document is the current working XML document that is used by the University of Pittsburgh Schneider Research Lab. This is the document that the Tract Rater Application will be loading, editing, and saving. The current version of the XML Rater Document is attached at the end of this paper. This document is subjected to minor changes throughout the rest of the project.
2. *XML Parser*

   The XML Parser is a parser for the XML Rater Document. It is meant to take the current XML Rater Document and convert it to a JavaScript object and vice versa. It will be utilized by the Tract Rater Application as the final step to save the XML Rater Document. The purpose of the XML Parser is to have a clean separation between the XML Rater Document and the Tract Rater Application.
3. *Design Images*

   The Design Images describe the look and feel of the current front-end design for the Tract Rater Application. They are attached at the end of this paper. These Design Images are subjected to minor changes throughout the rest of the project depending on the needs of the University of Pittsburgh Schneider Research Lab and the changes of the XML Rater Document.

Selected Design Approach

As you can see from the Decision Matrix above, the selected design approach was AngularJS. There are multiple reasons for this, as described below.

1. The biggest problem with AngularJS is that I have never used the AngularJS framework. However, this is problem is offset by the fact that I have access to an online course. There is a major learning curve that I am overcoming, but I am confident that I can have the project completed by the desired end date. Also, I am unfamiliar with the PyQt and GWT frameworks as well, so a learning curve could not be avoided.
2. The ability to run the the application as a completely client-side, standalone application or as a client-server web application is very profound. This allows the University of Pittsburgh Schneider Research Lab to decide which is best for their needs currently and in the future.
3. It is the most stable choice for the future because the AngularJS framework has been growing in size, popularity, and applications.

Minimum Standard for Project Completion

Main Objective of Project: The main objective of this project is to enable the user to conveniently, intuitively, quickly, and robustly analyze brain tracts and save all of the data in a machine readable manner. The Tract Rater Application will accomplish this. The Tract Rater Application has the following requirements:

Platform Requirements: *These requirements are related to the platform and the environment of the completed Tract Rater Application.*

Minimum Goals:

1. The Tract Rater Application shall run in Google Chrome.
2. The Tract Rater Application shall run on Mac and Linux.
3. The Tract Rater Application Documentation shall provide a list of pros and cons that include feasibility, possibilities, and security issues between using a web server and having a completely client-side application.

Stretch Goals:

1. The Tract Rater Application shall run in Firefox and Safari.
2. The Tract Rater Application shall run on Windows.
3. The Tract Rater Application shall run as a standalone application in a robust and secure manner.
4. The Client shall be provided with an Administrator Manual.

Application Requirements: *These requirements are related to the layout, interface, and implementation of the Tract Rater Application.*

Minimum Goals:

1. The Tract Rater Application shall contain widgets and fields to input all pertinent information to the XML Rater Document.
2. The Tract Rater Application shall import the XML Rater Document.
3. The Tract Rater Application shall be written in javascript using the angularJS framework.
4. The Tract Rater Application shall export and save the XML Rater Document.
5. The Tract Rater Application shall provide authentication with a username but not a password.
6. The User shall be able to cancel an incomplete rating and erase all previous edits.

Stretch Goals:

1. The Tract Rater Application shall provide authentication with a username and a password.
2. The User shall be able to review a previously rater subject in a non-editable manner.
3. The Tract Rater Application shall display the brain images of the subject's current tract for the User.

4. The Tract Rater Application shall display the brain images of the control's current tract for the User.
5. The Tract Rater Application shall indicate the overall progress of the user's analysis.

Software Engineering Requirements: *These requirements are related to the software practices and management of the Tract Rater Application.*

Minimum Goals:

1. The Tract Rater Application shall adhere to a Model-View-Controller architecture.
2. The Tract Rater Application shall adhere to API for the XML Parser.
3. The code for the Tract Rater Application shall have good documentation and comments.
4. The Tract Rater Application shall be version controlled through SVN.
5. The updates and issues concerning development will be communicated using the Clients Zoho Project Manager accounts.

Usability Requirements: *These requirements are related to the overall usability, comfortability, and ease of the Tract Rater Application.*

Minimum Goals:

1. The Tract Rater Application shall be user friendly and have an easy and intuitive layout.
2. The Tract Rater Application shall look similar to the Design Images.
3. The Tract Rater Application shall have a quick start up time.
4. The Tract Rater Application shall shield the User from the XML Rater Document format details such that the format and XML document can change without the Users notice.

Stretch Goals:

1. The Client shall be provided with a User Manual.

Testing Requirements: *These requirements are related to the testing of the Tract Rater Application.*

Minimum Goals:

1. The Tract Rater Application shall be tested using the Jasmine testing framework and the Karma test runner. [5]
2. The Tract Rater Application shall have exhaustive testing that checks for correct input syntax on user fields, graceful exits, correct xml values, unit tests, and test cases.
3. The Tract Rater Application shall be tested and surveyed by at least one experienced user and at least one inexperienced user. Feedback, comments, and concerns will be collected.

Non-Technical Constraints

Economic Constraints:

This project is completely software based and all of the software is free. There is no budget for this project. Therefore, there are no economic constraints for this project.

Environmental and Sustainability Constraints:

The Sustainability Subcommittee of PUPC has described some aspects that the University of Pittsburgh should do to be more sustainable.

Firstly, they say “Foster environmentally responsible purchasing practices by striving to balance short and long-term, maintenance, life cycle, and environmental costs in purchasing goods and services, with sound fiscal practices.” [1] For this project, no additional hardware was purchased in the making of the application.

Secondly, they say “Continue to identify, implement, communicate, and coordinate practices that preserve and promote efficient use and conservation of energy, water, and other resources and increase promotion of conservation efforts to the University community, so as to reduce resource consumption”. [1] This project will increase productivity and decrease confusion, so there is potentially less wasted time driving between labs.

Lastly, they say “Continue efforts to minimize solid waste production by fostering recycling and reuse.” [1] This project is completely software based, so a lot of work previously done on sheets of paper will now be done on a computer. This will save trees and promote reuse of products.

Manufacturability Constraints:

This project is completely software based. There is no manufacturing of goods or products. Therefore there are no manufacturability constraints.

Ethical, Health, and Safety Constraints:

The safety and reliability of software is very important, especially when embedded in heavy equipment or machinery. However, this particular project is not created to help users operate heavy machinery or endanger their health in any way. Therefore there is no health and safety constraint.

There are a variety of ethical issues that are related to this project because we are dealing with personal data and information from different medical subjects. There are confidentiality and security protocols that must be met. A lot of these protocols are focused on the XML Rater Document which is the final document for the storage of this information. This document follows all security and confidentiality protocols within the University of Pittsburgh Schneider Research Lab.

Some of these issues and protocols must be applied to the Tract Rater Application also. For example, confidentiality protocols are being upheld through the fact that the Tract Rater Application knows nothing about the demographics of the subject being analyzed. It only knows the subject by a unique numerical identifier that is given to the subject during initial processing. This unique numerical identifier is only used within the University of Pittsburgh Schneider Research Lab.

The security protocols are being met by the Tract Rater Application also. This is done through the requirements for authorization. This is another reason why it is important to provide the option for the project to be a standalone application. There are a lot of security risks that become issues if the application must be hosted on a web server.

Social and Political Constraints:

There are no social or political constraints for this project. The demographics of the subject being analyzed are completely anonymous to the Tract Rater Application, therefore there is no potential for bias or discrimination against race or gender.

Project Revisions

The only project revision since the Written Proposal is the addition of the XML Parser by the University of Pittsburgh Schneider Research Lab Technical Team. This XML Parser, as described above, is a parser between the XML Rater Document and a Javascript Object. The Javascript Object is a middle man between the XML Rater Document and the Javascript Objects used by the Tract Rater Application. The API for the Javascript Object was collaborated, discussed, and agreed upon by myself and the University of Pittsburgh Schneider Research Lab Technical Team.

Preliminary Results

There are many features that have already been implemented and many more that need to be. Some of the features that are already implemented include the following:

1. The Tract Rater Application can download information to the users Downloads folder.
    a. This works but is not the best option. Ideally this will be changed so the user can pick where the file will be downloaded.
2. The Tract Rater Application can read the contents of any machine readable file on the local computer.
3. The Tract Rater Application can display information for each potential Tract.
4. The Tract Rater Application can add and save ratings, metrics, graph abnormalities, and observations for each specific tract.
5. The Tract Rater Application can edit and save observations.
6. The Tract Rater Application has the option for “No Graph Abnormalities” which can be clicked to remove the Textbox.

Figure 5 shows the initial layout designs for the Tract Rater Application.

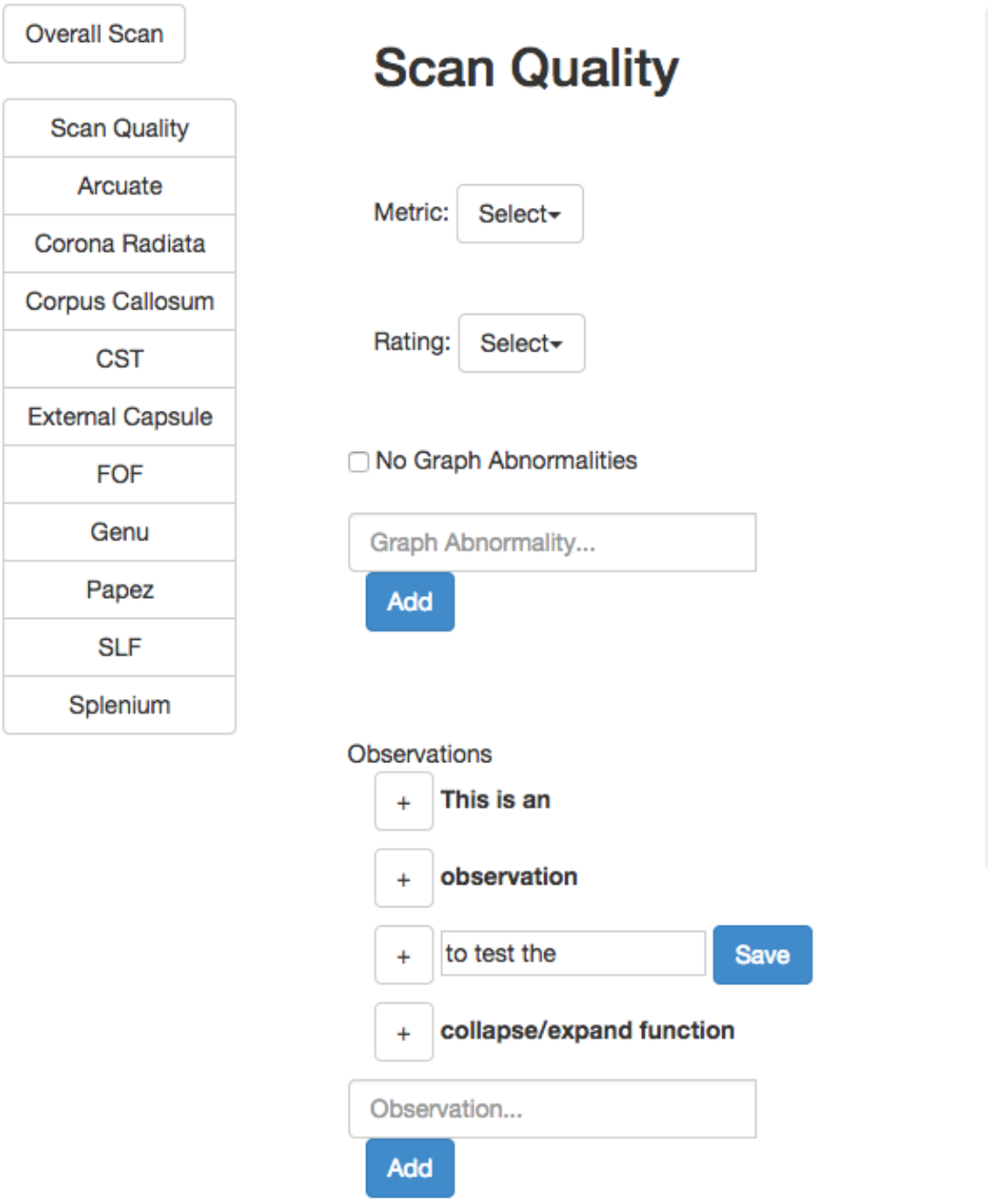


Figure 5: Current Look of the Tract Rater Application

References

[1] Favorini, Attilio. "Statement of Sustainability." *University of Pittsburgh* (2015): n. pag. Web.

[2] Fernandez-Miranda, Jan C., MD. "High-Definition Fiber Tractography of the Human Brain: Neuroanatomical Validation and Neurosurgical Applications." *Neurosurgery* (2012): n. pag. Print.

[3] Le Bihan, Denis, Md, PhD. "Diffusion Tensor Imaging: Concepts and Applications." *Journal of Magnetic Resonance Imaging* 13 (2001): 534-46. Print.

[4] Lerner, Ari. Ng-book: The Complete Book on AngularJS. S.I.: Fullstack.io :, 2013. Print.

[5] Ragonha, Paulo. Jasmine JavaScript Testing. Birmingham: Packt, 2013. Print.

[6] Spartley, Jonathan. Learning Yeoman Design, Implement, and Deliver a Successful Modern Web Application Project Using Three Powerful Tools in the Yeoman Workflow. Birmingham, UK: Packt Pub., 2014. Print.

[7] Summerfield, Mark. Rapid GUI Programming with Python and Qt: The Definitive Guide to PyQt Programming. Upper Saddle River, NJ: Prentice Hall, 2008. Print.

[8] Tacy, Adam. GWT in Action. 2nd ed. Shelter Island, NY: Manning Publications, 2013. Print.

[9] "MVC-Process" by RegisFrey - Own work. Licensed under Public Domain via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:MVC-Process.svg#mediaviewer/File:MVC-Process.svg

XML Rater Document

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE HDFTReportSummary SYSTEM "http://schweb1.lrdc.pitt.edu/schlab/projects/hdftmobile/report_summary.dtd">
<!-- Schneider Lab 2015 -->
<!-- caseID is the data json subjid field (used in HDFT pipeline processing)
  observation_date is the date the rater made the tract observations.  Format YYYY-MMM-DD  eg 2014-Dec-21→

<HDFTReportSummary version="2.0" origin="Schneider HDFT Lab, University of Pittsburgh"  caseID="zillah"
observation_date="2014-Jan-15">

  <!-- each of the elements inside provenance contains text that identifies who performed that processing step.   text can be
initials or other identifying info and more that 1 person can be identified, eg wb  or will lauren etc -->
<provenance>
   <observations></observations>
   <ROI></ROI>
   <trimming></trimming>
   <report_compilation></report_compilation>
</provenance>

<!-- scan element pertains across all tracts in the scan.  Comments, data_acquistion, and scan_quality are free text fields.
Comments are currently not used in CaseReport or iPad.-->
<scan>
   <comments></comments>
   <data_acquisition>All imaging data was acquired at the University of Pittsburgh Medical Center Presbyterian Hospital
Magnetic Resonance Research Center on a Siemens 3T Tim Trio system using a 32-channel head coil as part of an ongoing
research study by Drs. Walter Schneider and David Okonkwo. White matter tract analysis was performed in a qualitative and
quantitative fashion on 17 major fiber pathways of the brain. Overall, image acquisition and tractography quality was adequate,
indicating a valid scan.</data_acquisition>
   <scan_quality></scan_quality>
</scan>

<!-- tract element is repeated for each tract processed.-->
<tract tractID="CoronaRadiata" long_name="Corona Radiata" short_name="Corona Radiata" >
<!--        report_config pertains to formatting the pdf CaseReport
            third_view takes values:  SAG COR AX
            show_observations, show_bullets, show_graph_abnormatities, take values 0 or 1
-->
   <report_config>
      <show_observations>1</show_observations>
      <show_bullets>1</show_bullets>="1"
      <show_graph_abnormatities>1</show_graph_abnormatities>
      <third_view></third_view>
   </report_config>

<!--        trk_hierarchy indicates relations between tracts; always use tractID to
            identify a tract.
            parent_trk is a single tractID of parent tract
            child_trk is a single tractID of a child tract; the <child_trk> element can be
            repeated for as many child trks as needed.
```

```
-->
   <trk_hierarchy>
      <parent_trk></parent_trk>
      <child_trk></child_trk>
      <child_trk></child_trk>
   </trk_hierarchy>

<!-- summary is a short free text field, typical entries include Normal Range, Moderate Concerns, Possible Concerns→
   <summary>Normal Range</summary>
            <!-- rating takes values 0-5 -->
   <rating>2 </rating>

<!-- observation is a free text field.  attribute bullet_string is a short summary of the observation text.  The <observation> element can be repeated.  If it is repeated the CaseReport may make each element a separate bullet point in the report (But it could also concatenate observations into a paragraph.) -->
   <observation   bullet_string="sparse">This tract is a little bit sparse in anterior region.</observation>
   <observation   bullet_string="obs2"> This is another observation. </observation>

<!-- graph element can be repeated multiple times. One <graph> element pertains to all 4 projections (x,y,z,vol) for a given metric. Note this element is tract specific but the graph metric should be the same across all tracts.   GUI can make that happen by putting editable field for <graph_metric> outside the tract panes and reflecting the choice in non-editable box in each tract pane.-->
   <graph>
      <graph_metric> allfibs voxelscontacted anat</graph_metric>

      <!-- <graph_abnormality> can be repeated as many times as needed.  Need to find out if it makes sense to have multiple abnormalities for the same proj.  In future can add a <comment> element if needed and maybe a zscore element.  →
      <graph_abnormality>
         <!-- bullet and proj entries need to be dropdowns -->
         <bullet>high/low</bullet>
         <proj>X</proj>
     </graph_abnormality>
   </graph>
</tract>

<tract tractID="CST">
   <summary>Normal Range</summary>
   <observation>This tract falls within normal range compared to healthy controls.</observation>
</tract>

<tract tractID="Genu">
   <summary>Normal Range</summary>
   <observation>This tract falls within normal range compared to healthy controls.</observation>
   <trk_hierarchy>
      <parent_trk>CorpusCallosum</parent_trk>
   </trk_hierarchy>
</tract>

<tract tractID="CCBody">
   <summary>Normal Range</summary>
   <observation>This tract falls within normal range compared to healthy controls.</observation>
   <trk_hierarchy>
      <parent_trk>CorpusCallosum</parent_trk>
   </trk_hierarchy>
</tract>

<tract tractID="Splenium">
```

<summary>Possible Concerns</summary>
<observation>The right side tapetum projections into the temporal lobe are sparse compared to the left side and compared to healthy controls. However, imaging of the tapetum is relatively less certain than other tracts in the brain.</observation>
<trk_hierarchy>
<parent_trk>CorpusCallosum</parent_trk>
</trk_hierarchy>
</tract>

<tract tractID="CorpusCallosum">
<summary>Possible Concerns</summary>
<observation>The genu falls within normal range. The body falls within normal range. In the splenium, the right side tapetum projections into the temporal lobe are sparse compared to the left side and compared to healthy controls. However, imaging of the tapetum is relatively less certain than other tracts in the brain.</observation>
<trk_hierarchy>
<child_trk>CCBody</child_trk>
<child_trk>Genu</child_trk>
<child_trk>Splenium</child_trk>
</trk_hierarchy>
</tract>

<tract tractID="Cingulum">
<summary>Normal Range</summary>
<observation>This tract falls within normal range compared to healthy controls.</observation>
</tract>

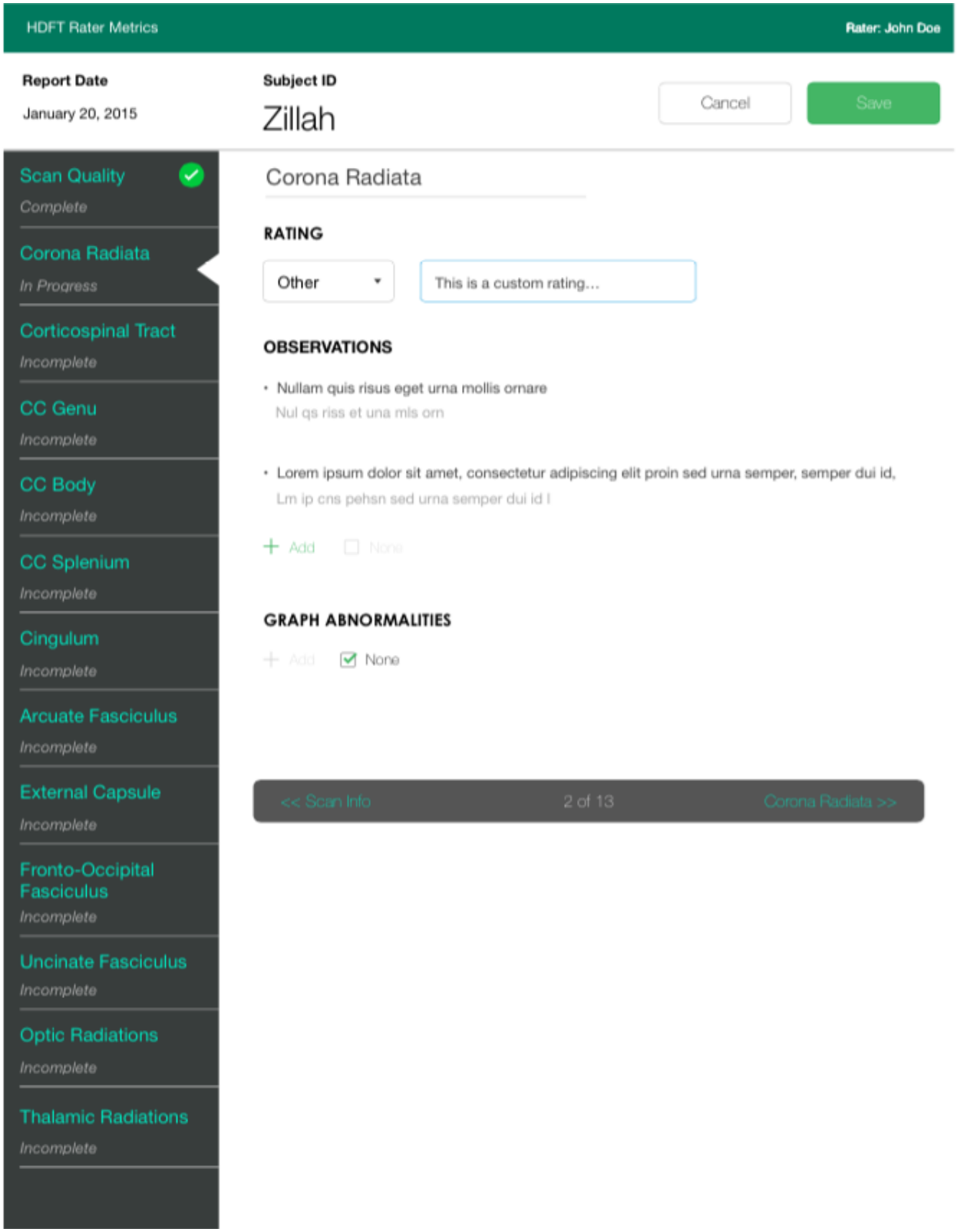
<tract tractID="Arcuate">
<summary>Normal Range</summary>
<observation>This tract falls within normal range compared to healthy controls.</observation>
</tract>

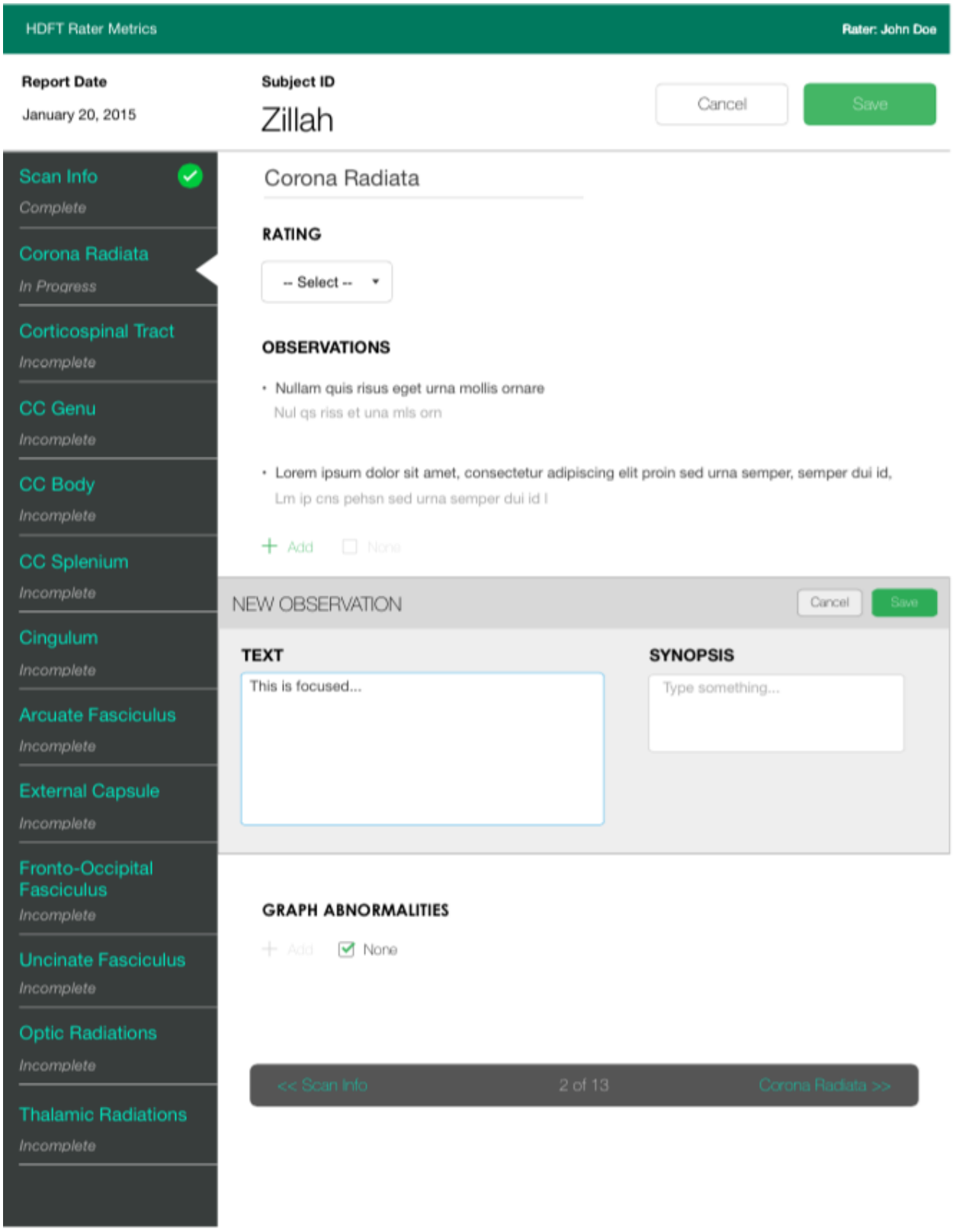
<tract tractID="ExternalCapsule">
<summary>Possible Concerns</summary>
<observation>Right side fanning projections to the superior premotor area in the frontal lobe are sparse compared to the left side. However, imaging of the external capsule is relatively less certain than other tracts in the brain.</observation>
</tract>

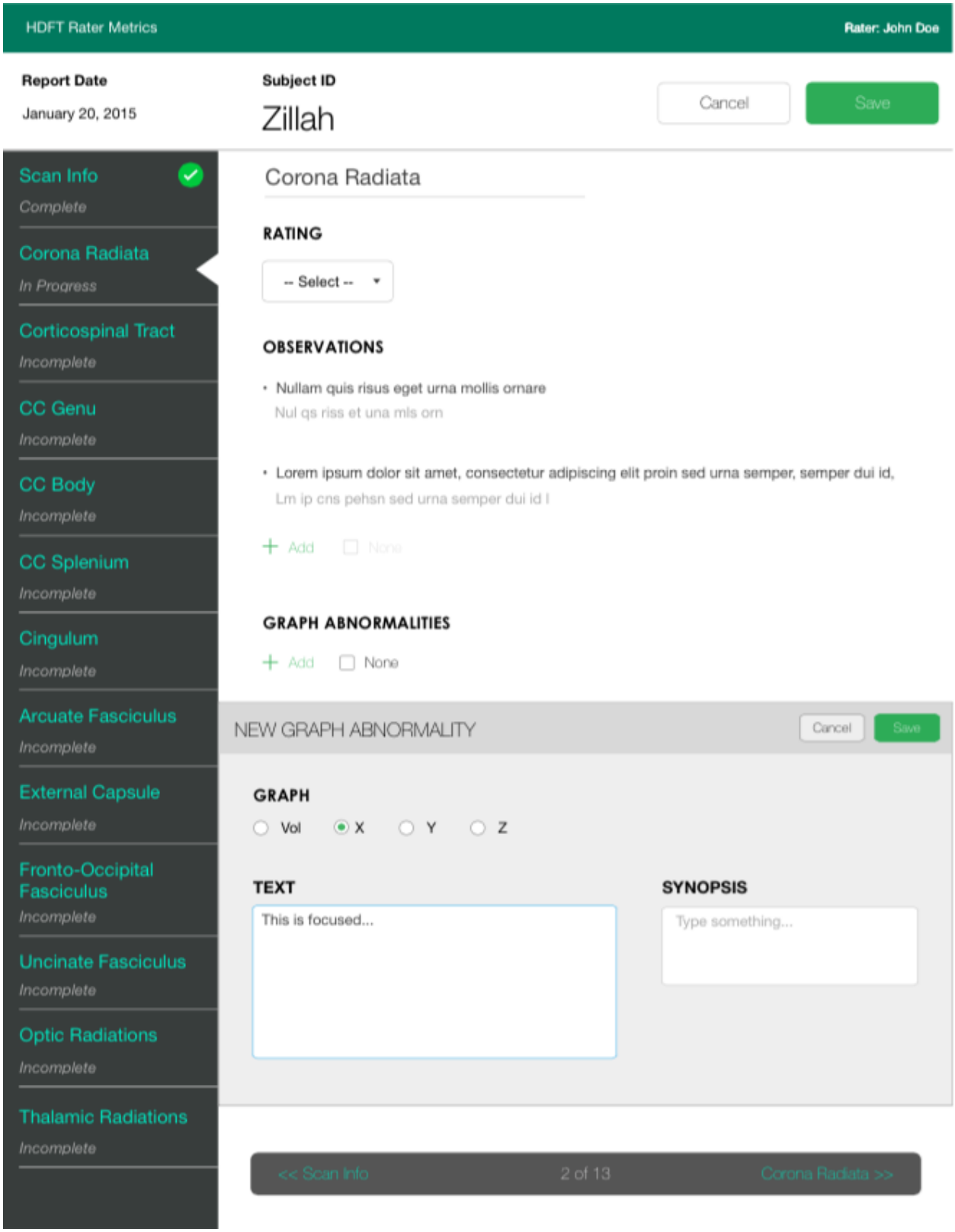
<tract tractID="FOF">
<summary>Possible Concerns</summary>
<observation>Right side frontal projections are sparse compared to the left side.</observation>
</tract>

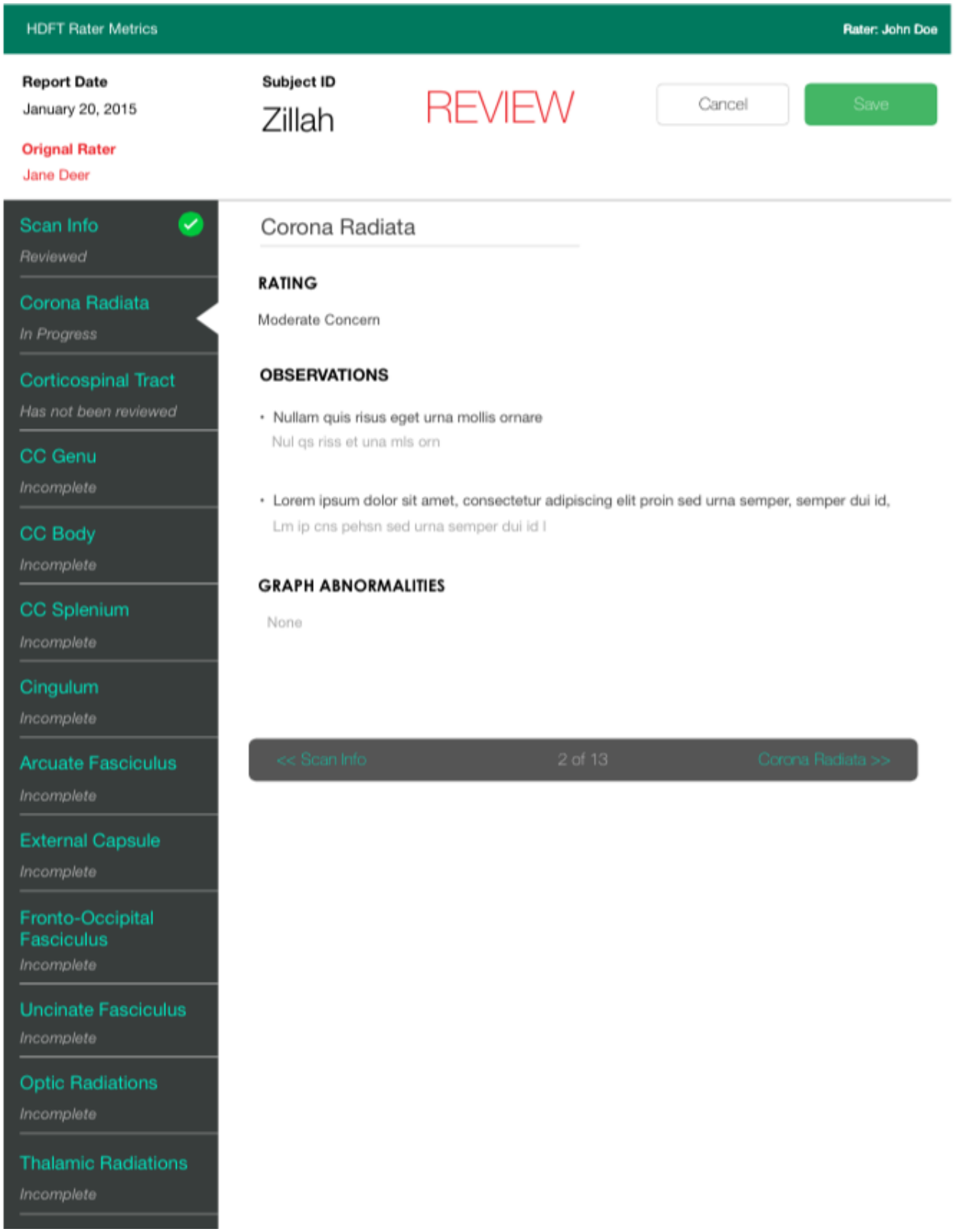
</HDFTReportSummary>

Design Images




HDFT Rater Metrics
Rater: John Doe
Report Date
January 20, 2015
Subject ID
Zillah
Cancel
Save
Scan Info
Complete
Corona Radiata
In Progress
Corticospinal Tract
Incomplete
CC Genu
Incomplete
CC Body
Incomplete
CC Splenium
Incomplete
Cingulum
Incomplete
Arcuate Fasciculus
Incomplete
External Capsule
Incomplete
Fronto-Occipital Fasciculus
Incomplete
Uncinate Fasciculus
Incomplete
Optic Radiations
Incomplete
Thalamic Radiations
Incomplete
Corona Radiata
RATING
-- Select --
OBSERVATIONS
Nullam quis risus eget urna mollis ornare
Nul qs riss et una mls orn
Lorem ipsum dolor sit amet, consectetur adipiscing elit proin sed urna semper, semper dui id,
Lm ip cns pehsn sed urna semper dui id l
Add
None
NEW OBSERVATION
Cancel
Save
TEXT
This is focused...
SYNOPSIS
Type something...
GRAPH ABNORMALITIES
Add
None
<< Scan Info
2 of 13
Corona Radiata >>


HDFT Rater Metrics
Rater: John Doe
Report Date
January 20, 2015
Subject ID
Zillah
Cancel
Save
Scan Info
Complete
Corona Radiata
In Progress
Corticospinal Tract
Incomplete
CC Genu
Incomplete
CC Body
Incomplete
CC Splenium
Incomplete
Cingulum
Incomplete
Arcuate Fasciculus
Incomplete
External Capsule
Incomplete
Fronto-Occipital Fasciculus
Incomplete
Uncinate Fasciculus
Incomplete
Optic Radiations
Incomplete
Thalamic Radiations
Incomplete
Corona Radiata
RATING
-- Select --
OBSERVATIONS
Nullam quis risus eget urna mollis ornare
Nul qs riss et una mls orn
Lorem ipsum dolor sit amet, consectetur adipiscing elit proin sed urna semper, semper dui id,
Lm ip cns pehsn sed urna semper dui id l
Add
None
GRAPH ABNORMALITIES
Add
None
NEW GRAPH ABNORMALITY
Cancel
Save
GRAPH
Vol
X
Y
Z
TEXT
This is focused...
SYNOPSIS
Type something...
<< Scan Info
2 of 13
Corona Radiata >>


HDFT Rater Metrics
Rater: John Doe
Report Date
January 20, 2015
Orignal Rater
Jane Deer
Subject ID
Zillah
REVIEW
Cancel
Save
Scan Info
Reviewed
Corona Radiata
In Progress
Corticospinal Tract
Has not been reviewed
CC Genu
Incomplete
CC Body
Incomplete
CC Splenium
Incomplete
Cingulum
Incomplete
Arcuate Fasciculus
Incomplete
External Capsule
Incomplete
Fronto-Occipital Fasciculus
Incomplete
Uncinate Fasciculus
Incomplete
Optic Radiations
Incomplete
Thalamic Radiations
Incomplete
Corona Radiata
RATING
Moderate Concern
OBSERVATIONS
Nullam quis risus eget urna mollis ornare
Nul qs riss et una mls orn
Lorem ipsum dolor sit amet, consectetur adipiscing elit proin sed urna semper, semper dui id,
Lm ip cns pehsn sed urna semper dui id l
GRAPH ABNORMALITIES
None
<< Scan Info
2 of 13
Corona Radiata >>