#### 筑波大学大学院博士課程

#### システム情報工学研究科特定課題研究報告書

海底コアCTスキャンデ - 夕可視化・情報共有を可能とするクラウドサ - ビスの構築- ユーザ管理機能の開発とAndroid端末向けクライアントアプリケーションのテスト -

城﨑亮

修士(工学)

(コンピュータサイエンス専攻)

指導教員 田中二郎

2014年3月

先行研究 [1] では海底コア試料と呼ばれる深海底掘削サンプルの X 線 CT スキャンされたデー タ (コアデータ) を WEB ブラウザ , またはタブレット端末から閲覧することができるクラウド サービスを公開中 [6] であるが,既存システムでは閲覧が機能の中心であり,利用者同士が情 報を提供するということが難しいという問題がある.そこで本プロジェクトでは,コアデー タを閲覧する際にその閲覧状態を保存するブックマーク機能とコアデータの任意の場所にメ モを保存することが出来るアノテーション機能を開発する.ブックマーク機能では閲覧状態 を他の研究者と共有できるようになり,アノテーション機能ではコアデータに自分の意見を 添付することができるようになる.これらの機能により、研究者間の議論を行うことができ るようになり、これまでより、よい研究成果を生み出すことができると考えられる.本プロ ジェクトにおいて筆者は主に二つのことを行う.一つはユーザ管理機能の開発である.誰にで もメモを操作することを許可した場合,悪意のあるユーザが保存されたメモを改竄したり削 除するような不正操作が発生すると考えられる.ユーザ管理機能を追加することにより、メ モを操作する人を制限することができる、保存されたメモはメモの作成者しか編集や削除と いった操作をできないよう設定する.また,アノテーション機能においてメモを保存するこ とができるユーザを制限するとともに、メモを残した人を識別することができる、もう一つ はタブレット端末向けアプリケーションのテストである.PC ではなく主にタブレット端末を 多く利用する環境にあるユーザがいると顧客よりヒアリングしており、タブレット端末アプ リケーションが正しく動作する必要がある.このアプリケーションが正しく動作することを 検証するためにテストを行う.今回は,実装した機能が意図したとおりに動作しているかど うかを検証するため,全てのボタンやテキストに対し操作を行う.これにより,仕様書に沿っ た機能をユーザに提供する.これら二つのことにより、システムを悪意があるユーザから保 護する、ユーザに高品質なサービスを提供することが可能になる、

# 目次

第1章	はじめに	1
第 <b>2</b> 章 2.1	開発背景と解決すべき課題 統合国際深海掘削計画 (IODP) とコア試料	<b>3</b>
2.2	既存システムと解決すべき課題	3
2.3	議論	5
第3章	海底コア CT スキャン可視化・情報共有を可能にするクラウドサービス	7
3.1	提案する機能	7
	3.1.1 ブックマーク機能	7
	3.1.2 アノテーション機能	9
3.2	システム構成と開発計画	11
	3.2.1 既存システムの構成	11
	3.2.2 提案するシステム構成	13
	3.2.3 開発すべき項目とスコープ	15
第4章	ユーザ管理機能の開発	18
4.1	機能概要	18
4.2	全体構成と開発計画	19
4.3	設計	21
4.4	実装	21
	4.4.1 ユーザ登録機能	21
	4.4.2 ログイン機能	24
	4.4.3 ログアウト機能	24
	4.4.4 ユーザ管理画面	24
4.5	まとめ	26
第5章	Android 端末向けクライアントアプリケーションのテスト	27
5.1	テスト概要	27
5.2	テスト環境	31
5.3	結果と評価	32
笙 6 音	おわりに	34

	謝辞	35
	参考文献	36
付録A	単体テスト項目	38
付録B	画面定義書	41

# 図目次

3.1	ブックマーク機能導入前のイメージ	8
3.2	ブックマーク機能導入後のイメージ	9
3.3	アノテーションが付加されたコア画像	10
3.4	アノテーション検索画面	11
3.5	既存のシステム構成	12
3.6	本プロジェクトの開発後のシステム構成	13
3.7	アノテーションの取得	14
3.8	アノテーションの編集と削除	14
3.9	アノテーションの追加	15
3.10	メモの表示位置	16
3.11	担当箇所のスケジュール	16
4 1	ユーザ管理機能の構成	20
4.2	ユーザ登録画面	22
4.3	ハッシュ関数の実行結果	23
4.4	乱数とハッシュ関数の実行結果	23
4.5	低登録ユーザ管理画面	25
4.6	放豆豚ユーザ   管理画面	26
4.0	豆琢炉がユーク自注画面 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	20
5.1	サンプルプログラム	28
5.2	命令網羅のパターン	29
5.3	分岐網羅のパターン	30
5.4	バグ曲線	32

### 第1章 はじめに

独立行政法人海洋研究開発機構 (JAMSTEC) が所有する地球深部探査船「ちきゅう」は,上部マントルや巨大地震発生域への大深度掘削を可能にする世界初のライザー式科学掘削船であり,統合国際深海掘削計画 (IODP)[8] の主力船として深海底掘削探査を行っている[2].

IODP は日本と米国が主導する国際的な海洋科学掘削計画であり、震源物質調査や気候変動 史,地下生物圏の解明を目的としている「ちきゅう」は海底コア試料と呼ばれる,海底を掘 削して得られる柱状の深海底掘削サンプルを採取している.海底コア試料は,過去の地球環 境の変化だけでなく、地震の発生メカニズムや海底資源の発見などのため、世界中の研究者 により役立てられている「ちきゅう」の特徴として,船上に X 線 CT スキャナを搭載してお り,これは他の掘削船には搭載されていない「ちきゅう」が掘削した海底コア試料は,最初 に X 線 CT スキャナを用いてデジタルデータ化される . デジタルデータ化することにより , 海 底コア試料を破壊することなく、その内部を可視化することができる、デジタルデータ化さ れたコアデータは同センターの Virtual Core Library[7] にて公開されており, だれでも自由に 利用することができる.しかし,X 線 CT スキャナは医用であり,閲覧用のビューアも医用に 特化しているため,コアデータを見るには不向きな側面があるという課題があった.さらに, 既存のビューアは膨大なスライス画像を3次元再描画するため,端末に大きい負荷がかかり コアデータを閲覧するためには高性能な PC が必要であり, コアデータを活用できる人は限ら れていた.先行研究である "海底コア CT スキャンイメージ可視化のためのクラウドサービス 開発"[1] により Web ブラウザやタブレット端末から閲覧することができるようになり,先行 研究にて開発されたシステムは研究者に利用されている.現在のシステムではコアデータの 閲覧という単純な操作のみが可能である.さらに,研究者からはコアデータの閲覧状態を保 存するような機能や、コアデータにメモを保存する機能を希望されている、これらの機能を 追加することにより、他者と情報を共有し研究することができるようになる。

本プロジェクトでは研究者が希望しているコアデータの閲覧状態を保存するような機能や,コアデータにメモを保存する機能を実現するため,海底コア CT スキャン可視化・情報共有を可能にするクラウドサービス」を構築する.本システムの開発は先行研究である"海底コア CT スキャンイメージ可視化のためのクラウドサービス開発"をベースに新しい2つの機能,コアデータの閲覧状態を保存するブックマーク機能とコアデータにメモを保存するアノテーション機能の開発を行う.ブックマーク機能は,ユーザがコアデータを閲覧する際にその閲覧状態を保存し,その閲覧状態を他のユーザに共有するといった事ができる.アノテーション機能は,コアデータを閲覧中にコアデータの気になったポイントにメモを保存することができ,そのコアデータを見る他のユーザに知見や意見を共有することができる.

しかし,アノテーション機能だけでは,メモの編集や削除を誰でも行うことができ,あるユーザが残したメモを悪意がある人物により改竄されたり削除される,といった問題が発生する可能性がある.この問題を解決するため,ユーザ管理機能を追加しメモを保存した人にしか編集や削除といった操作をさせないようにした。また,同時に

- 誰がメモを残したかということがわかるメモと一緒にメモを残したユーザ名を表示する。
- 悪意があるユーザにメモを保存させない
   ユーザ登録したユーザしかメモを残すことができず,悪意があるユーザにはユーザ登録を許可しない。

といった効果も期待できる.このような,ユーザ登録画面からシステム運用者用のユーザ管理画面など,ユーザ管理にかかわる機能の開発を行う.ユーザ管理機能によりユーザ間の議論を活発にすることができ,かつ悪意のあるユーザからシステムを保護することができる.

タブレット端末向けアプリケーションのテストでは, Android 端末向けアプリケーションのテストを行う.実際にユーザが利用すると考えられる Android 端末のうちいくつかに開発したアプリケーションをインストールし, 仕様書に沿った動作をすることの検証を行う.アプリケーションのテストを行うことにより, 本プロジェクトが意図した機能を実装している高品質なサービスをユーザに提供することができる.

筆者が行ったユーザ管理機能の開発と Android 端末向けアプリケーションのテストにより, 地質研究者の議論を本システム上で実現することができるようになる.これにより,活発に意見交換が行われ,本システム上では議論ができなかった頃よりもよい研究成果が生まれることを期待できる.

本報告書は、全6章で構成されている.第2章では本研究の背景について述べる.第3章では本プロジェクトが開発した「海底コア CT スキャンデ - タ可視化・情報共有を可能とするクラウドサ - ビスの構築」について述べる.第4章ではユーザ管理機能の実装について述べる.第5章では Android 端末向けアプリケーションのテストについて述べる.第6章ではまとめを述べる.

## 第2章 開発背景と解決すべき課題

本章では本研究の背景となっている統合国際深海掘削計画や先行研究について述べる.

#### 2.1 統合国際深海掘削計画 (IODP) とコア試料

本プロジェクトが共同研究を行っている JAMSTEC は,海洋に関する基盤的研究開発,海洋に関する学術研究に関する協力等の業務を総合的に行っている.JAMSTEC の活動の中には IODP[8] がある.IODP とは日本と米国が主導する地球環境変動、震源物質調査や気候変動史,地下生物圏の解明を目的とした国際的な海洋科学掘削計画である.JAMSTEC が所有する地球深部探査船「ちきゅう」は IODP において主力船として地球探査を行っている「ちきゅう」は上部マントルや巨大地震発生域への大深度掘削を可能にする世界初のライザー式科学掘削船であり,深海底を数千メートル掘削し貴重なコア試料を採取している[2]. さらに「ちきゅう」が掘削したコア試料を調査することにより,地震の発生メカニズムは今まで考えられていたものとは異なることが発見された[3][4][5].

掘削されたコア試料は 1.5 メートル毎に切断しセクションと呼ばれるものに分割される.セクションには航海番号やセクション番号といった ID を割り当て,データベースに登録し管理する.掘削されたコア試料の実物は高知コアセンターにて保管される.保管する際にはコア試料を縦に半分に分ける.片方は研究に利用するために切り分けることがあるが,もう片方は手を加えずに保存される.コア試料は物質が変化しないように冷蔵庫にて保存される.

#### 2.2 既存システムと解決すべき課題

X 線 CT スキャンされたコアデータは DICOM フォーマットという形式のファイルに保存される.そのため,Virtual Core Library[7] にて公開されているデータも DICOM フォーマットであった.その際

- 1. X線 CT スキャナは主に医用に利用されており, DICOM フォーマットの画像を閲覧するためのビューアは医用に特化したものしか存在せず, コアデータを閲覧するには不向きな側面がある
- 2. DICOM フォーマットのビューアは膨大なスライス画像を 3 次元再描画するため, CPU やメモリに大きな負荷がかかり, 閲覧するためには高性能な PC が必要である

- 3. コアデータはファイルサイズが大きく,広い通信帯域を必要とする
- 4. コアデータはファイルサイズが大きく, そのファイルを保存するストレージが必要である

という課題が存在した. リモートサーバで MPEG に変換しビデオストリーミングを行うという研究 [9] はあったが, ストリーミングは通信するデータの量が大きく, サービスの質がネットワークの帯域によって左右されるという問題がある. そこで先行研究では, 次のことを行った.

- コアデータの画像処理を行うレンダリングサーバを構築する レンダリングサーバは, DICOM フォーマットの画像からコアデータの三次元構成を行 い, それを任意の平面に切り取ったコアデータの二次元画像を生成する.
- コアデータを閲覧することに適したインタフェースを開発する
  Web ブラウザと Android 端末からどのコアデータを描画して欲しいかというリクエストをレンダリングサーバへ送り,サーバは生成した二次元画像のみを Web ブラウザとAndroid 端末へ返す.

#### これにより,

- 1. コアデータを閲覧することに適したインタフェースを利用することができる
- 2. コアデータの三次元構成という端末に大きな負荷がかかる処理をユーザの端末で行わないため,ユーザの端末の性能を気にする必要がない
- 3. DICOM ファイルはサーバにあればよいので,ユーザがダウンロードする必要がなく, 保存するストレージも必要ない

といったように,端末の性能を気にすることなく,コアデータを閲覧しやすいインタフェースを利用することができるようになった.その結果,次のようなことができるようになった.

「ちきゅう」船上で,掘削したコア試料と過去に掘削したコアデータを比較しながら見る

これまではコアデータを閲覧するために高性能な PC や広い通信帯域を必要としていたため「ちきゅう」船上の設備だけでは過去のコアデータを閲覧することは難しかった.

● 学会や会議のような場所で,コアデータを見ながら議論をする コアデータを閲覧するためには高性能な PC が要求されるが,会議の場にそのような設 備を用意することが難しいという問題があった.

先行研究において,コアデータを閲覧することに適したビューアを低性能な PC から利用し,DICOM フォーマットの画像を閲覧することができるようになった.しかし,コアデータは閲覧する箇所が閲覧できる箇所が膨大であり,その中から直感的に部位を特定するという研究スタイルであるため,研究者からは

- コアデータの閲覧状態を保存する機能
- コアデータに対してメモを保存する機能

を新たに要望されている.これらの機能を提供できれば,複数の研究者が協調して研究を行うことができるようになる.

#### 2.3 議論

本章では,IODPとは何か,コア試料の利用価値,先行研究の成果,先行研究にて未解決の問題について述べた。本プロジェクトでは先行研究にて未解決の次の二つの問題を扱う。

- コアデータの閲覧状態を保存できない
  - コアデータを閲覧する際には,
    - 1. 閲覧したいコアデータの航海番号やセクション番号などの ID を指定する
    - 2. コアデータの回転や視点の移動
    - 3. CT 値に対する色付け

といった操作を行い,見たい部分に視点を合わせなければならない.X線CTスキャンした際にスキャンした物質の密度や元素番号に比例したX線CT値として保存される. CT値に対する色付けというのは,CT値に対してユーザが見やすいようにどの値にどの色を割り当てるかを設定することである.既存のシステムでは,

- 1. Web ブラウザを閉じる
- 2. Android アプリケーションを終了する
- 3. 他のコア画像を閲覧する

といった操作のいずれかを行うと,再びコアデータの ID を指定し上記の手順を繰り返さなければならない.さらに,それらの情報はどこにも保存されておらず,閲覧状態を一つ一つメモする,または記憶しなければならない,という問題がある.

● コアデータに対してメモを保存できない

既存の機能ではコア画像を閲覧中に気になる箇所を発見しても自分でその内容をメモする必要がある.誰かと共有するためには気になった内容を共有する前に,コアデータの閲覧状態,画像中のどのあたりなのか,ということを伝えければならず,他者との共有が難しいという問題がある.

本プロジェクトではこれらの問題を解決するために、

● コアデータの閲覧状態を保存する機能

#### • コアデータに対してメモを保存する機能

以上の二つの機能の開発を行う.コアデータの閲覧状態を保存する機能では,ユーザが閲覧するために行った操作情報のパラメータを保存することにより,レンダラを行っているサーバへ描画リクエストを送る際にそのパラメータを利用し閲覧状態を再現することができるようになる.コアデータにメモを保存する機能では,コアデータの任意の位置にメモを保存することができるようになり,コアデータを閲覧中にコアデータに対して自分の意見を残すことができるようになる.これらの機能により,開発したシステム上で世界中の研究者と議論を行うことができるようになり,研究者同士が互いに情報を共有することができるようになる.それにより,これまでよりもよい研究成果が生み出されることを期待できる.

# 第3章 海底コアCTスキャン可視化・情報共有 を可能にするクラウドサービス

本章では本プロジェクトにて開発する機能とその構成について述べる.

#### 3.1 提案する機能

本プロジェクトが開発する二つの大きな機能について述べる.一つはコアデータの閲覧状態を保存する機能であり,ブックマーク機能と呼んでいる.もう一つはコアデータに対してメモを保存する機能であり,アノテーション機能と呼んでいる.次に各機能の詳細について述べる.

#### 3.1.1 ブックマーク機能

ブックマーク機能は,コアデータを閲覧中にコアデータの ID や視点の移動, CT 値に対する色付けなどの操作情報を保存するすることができるようになる機能である.既存のシステムではコアデータを閲覧する際に

- 1. 閲覧したいコアデータの航海番号やセクション番号などの ID を指定する
- 2. コアデータの回転や視点の移動を行う
- 3. CT 値に対する色付けを行う

以上のような操作を行わなければならない.これは,コア画像を表示しているブラウザを閉じる,別のコア画像を表示する,といった操作を行った場合には,再度同じ操作を行わなければならない.そのため,コアデータを閲覧中に他の研究者にも同じ画面を共有したい,という場面において,自分が行った操作を1つずつ相手に伝えなければならない(図 3.1).図 3.1 ブックマーク機能が導入されていないシステムにて,コアデータの閲覧状態を共有しようとする場合の利用イメージである.二人のユーザがおり,左のユーザが見ているコアデータの閲覧状態を右のユーザに伝えようとしている.そのために,左のユーザは自分が行った操作を一つ一つ右のユーザに伝えている.このとき,左のユーザが間違ったパラメータを伝える,右のユーザが操作をミスする,といった場合には右のユーザは同じ閲覧状態を再現することはできない.また,誤ってブラウザを閉じた場合なども同様の操作を行う必要があるが,こ



図 3.1: ブックマーク機能導入前のイメージ

のとき自分が操作したパラメータを正確に覚えることは難しい.現在のシステムでは操作したパラメータをメモするなどすれば,同じ状態を復元することはできるが,閲覧状態のパラメータを全て毎回記録するのは多くの時間を費やし,記録ミスの可能性もあり困難であるという問題がある.

ブックマーク機能では,メモしたり記憶する必要なく,一度見た閲覧状態を再現することができるようになる.この機能を実装することにより,コアデータを閲覧中に次のようなことができるようになる.

#### ● 気になった部分を他者に共有する

自分が現在閲覧している状態を,同じ研究所の人,または世界中の研究者に共有することができる.断層など特殊な地層を発見した場合には,多くの研究者に見てもらうことにより,その地層に関する研究を促進させることを期待できる.

● 気になった閲覧状態をリストなどにまとめ、後で同じ状態のコアデータを見返す

「ちきゅう」が採掘したコアデータが Virtual Core Viewer に追加されたときに,一通り全部見て特に詳しく見たいものをブックマークすることができる.追加されたコア全体を短時間に把握することができ,その中から自分が興味があるものをあとでゆっくり見ることができる.また,コアデータを閲覧中だが,会議や電話など別の用事により,コアデータの閲覧を一時的に中断せざるを得ない場合にも利用することができる.

既存のシステムでは,閲覧状態を他社と共有したい場合にコアデータの番号や回転の角度,視点の移動など,大量の閲覧情報を相手に共有する必要があった.本プロジェクトでは,ボタンをクリックするだけで現在の閲覧状態を保持した URL を生成することができる.この URL を共有したい相手に送ることにより,相手は URL を開くだけで送信者と同じ閲覧状態を再現することができる(図 3.2).図 3.2 はブックマーク機能導入後の利用イメージである.二人のユーザがおり,左のユーザは自分が見ているコアデータの閲覧状態のブックマーク URL を右のユーザに伝えている.ブックマーク機能を利用した場合は URL を伝えるだけでよいという



図 3.2: ブックマーク機能導入後のイメージ

点がブックマーク機能導入前と異なる.また,URL はシステムが生成するため,ユーザの記録ミスに左右されず,閲覧状態を生成することができる.

#### 3.1.2 アノテーション機能

アノテーション機能は,コアデータを閲覧中にコアデータの任意の位置にメモを保存することができる機能である(図 3.3).図 3.3 はメモが保存されたコアデータを閲覧しているときの図である.円柱形のコアデータに対して星形のピンのようなものを刺している.星形のピンのようなものはメモが保存されていることを表しており,この星形の部分をクリックすると保存されたメモの内容を見ることができる.既存の機能ではコア画像を閲覧中に気になる箇所を発見しても自分でその内容をメモする必要がある.誰かと共有するためには気になった内容を共有する前に,コアデータの閲覧状態,画像中のどのあたりなのか,ということを伝えければならないという問題がある.

この問題を解決するためアノテーション機能では、コアデータの気になった箇所にメモを保存することができるようにする.アノテーション機能ではコアデータの表面や断面など任意の位置にメモを保存することができる.コアデータを回転させたりした場合にはメモはコアデータと一緒に回転し、どの視点から見てもメモがどこに付加されたものかがわかる.そのため、メモの位置を共有する際にはどのコアに付いているか、といった大まかな情報のみを伝えるだけで良い.前述した、ブックマーク機能を利用し、メモの位置を相手に伝えることもできる.また、アノテーションは特定のユーザに共有するだけでなく、世界中の研究者に意見を発信することができる.Virtual Core Viewer を利用しているユーザなら誰でも保存されたメモを閲覧することができるからである.これにより、システムを通して世界中の人と議論をすることができるようになる.また、保存されたメモは文章から検索することもできる.アノテーションの検索用に図 3.4 のようなページを用意した.図 3.4 のページは Web ブラウザから利用することができ、上には検索用のテキストエリア、その下に検索結果が表示

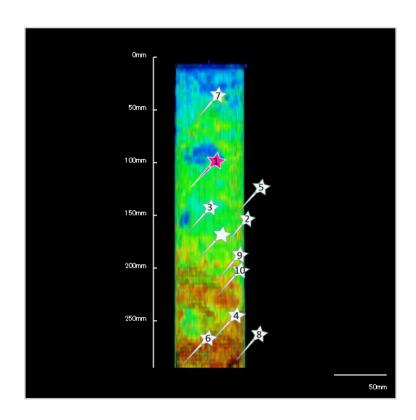


図 3.3: アノテーションが付加されたコア画像

されている.上のテキストエリアに検索したいワードを入力し,その横にあるボタンを押すことにより,

- 1. 検索ワードが含まれているアノテーションの本文
- 2. アノテーションの作成者
- 3. アノテーションが付加されているコアデータの ID

を検索することができる.例えば,'大震災'と入力し検索した場合は,アノテーションの本文に'大震災'が含まれるアノテーションのみが表示される.このとき,表示されているコアID は大震災に関係するコアであることが期待できる.これにより,他者が残したアノテーションから知見を広めることができる.

アノテーション機能ではコアデータに対してメモを保存することができるが,メモの内容を他者に編集されたり,削除されるという問題が発生する可能性がある.これは,ユーザに悪意がある場合や,誤操作により削除されるといった場合が考えられる.そのため,本プロジェクトではアノテーション機能の開発と同時に,ユーザ管理機能を追加し,保存されたメモに対する操作を制限する.



図 3.4: アノテーション検索画面

#### 3.2 システム構成と開発計画

次に本プロジェクトが開発するシステムの構成について述べる.最初に,既存システムの構成について述べる.次に,本プロジェクトが開発するシステムの構成について既存システムからの差分を述べる.

#### 3.2.1 既存システムの構成

既存システムの構成を図 3.5 に示す. 既存システムは図 3.5 の Virutual Core Viewer, ゲートウェイ,レンダラサーバの大きく分けて三つから構成される.

#### • クライアントアプリケーション

クライアントアプリケーションである Virtual Core Viewer は, Web ブラウザと Android 端末にて利用できる.

#### ゲートウェイ

ゲートウェイの中には検索 API 、検索モジュール 、レンダラ API 、レンダラモジュール 、検索用 DB が含まれる . 検索 API はクライアントアプリケーションからコア ID を検索する際に利用される . コア ID は航海番号やサイトナンバーといった識別子を複数組み合わせたものであり 、コア ID を指定すると 、その次に選択可能なサイトナンバーはいくつかに制限される . これをセクションナンバーという末尾の識別子まで繰り返す . その際にユーザからのリクエストを受け取る役割を持っている . 検索モジュールは検索 API から検索用のパラメータを受け取り DB からデータを取り出す役割を持っている . 検索用 DB にはコア ID と航海番号やサイトナンバーの対応が保存されている . レンダラ API はクライアントアプリケーションからコア ID やコアデータの回転角など描画に関するリクエストを受け取り 、レンダラモジュールはレンダラ API から描画に関するパラメータを受け取り 、レンダラサーバにあるレンダラを起動させ 、パラメータ通りの画像を生成させる .

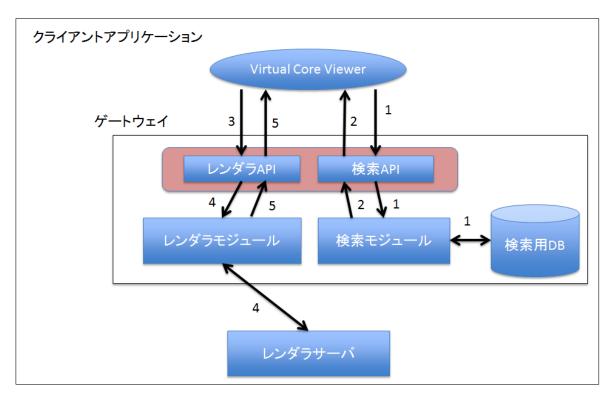


図 3.5: 既存のシステム構成

#### • レンダラサーバ

レンダラサーバは DICOM フォーマットの画像を三次元に再構築し,ユーザが見たい角度に回転させる,色付けを行う,といった端末に大きな負荷がかかる処理を行う.ユーザからのパラメータを受け取り,それをもとに生成した画像のみをレンダラモジュールに返す.

#### 既存システムでは

- 1. Virtual Core Viewer からゲートウェイにコアデータの検索リクエストを送る
- 2. ゲートウェイは Virtual Core Viewer にコアデータの ID を返す
- 3. Virtual Core Viewer はコアデータの ID をレンダラリクエストとしてレンダラ API に送る
- 4. レンダラ API はレンダラモジュールを通してレンダラサーバへ描画リクエストを送る
- 5. レンダラサーバは生成した二次元画像のみを Virtual Core Viewer へ返す.

という流れで,システムを実現しており,WebブラウザやAndroid端末から端末に大きな負荷をかけることなく,DICOMフォーマットの画像を閲覧することができる.

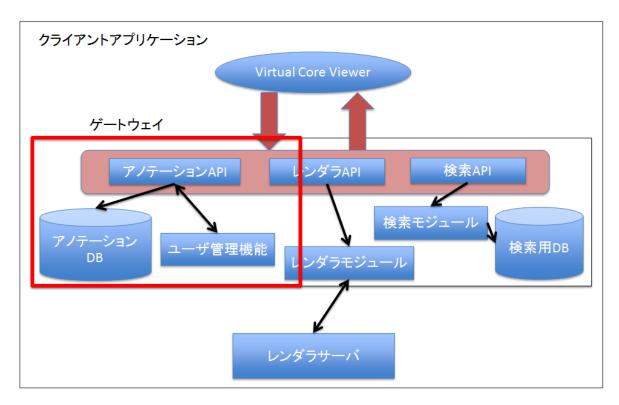


図 3.6: 本プロジェクトの開発後のシステム構成

#### 3.2.2 提案するシステム構成

次に,本プロジェクトのシステムの構成を図3.6に示す.図3.6の赤枠で囲まれた部分が本プロジェクトで新しく開発された部分であり,アノテーションAPI,アノテーションDB,ユーザ管理機能が含まれる.また,新機能の追加によりクライアントアプリケーションである Virtual Core Viewer も変更が行われている.既存システムからの差分は次の通りである.

#### ● アノテーション API

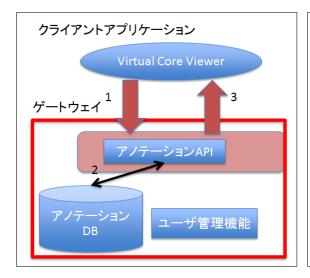
アノテーション API はユーザからコアデータの ID や閲覧情報を受け取り, Virtual Core Viewer に表示するベきアノテーションの取捨,アノテーションを表示,保存する位置の計算,などを行う.

#### ● アノテーション DB

アノテーション DB にはアノテーションの本文や座標, 作成者の名前といった情報が保存されている.

#### ● ユーザ管理機能

アノテーション機能を利用してメモを保存,編集,削除といった操作を行う際に,ユーザを確認するためにアノテーション API から参照される.



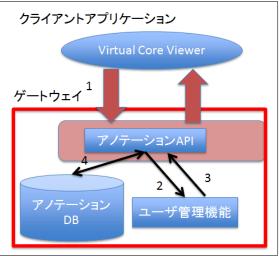


図 3.7: アノテーションの取得

図 3.8: アノテーションの編集と削除

既存システムと本プロジェクトが開発したシステムの大きな違いは,アノテーション機能であり,アノテーションの取得,編集,追加の流れを次に示す.最初にアノテーションの取得について図 3.7 に示す.図 3.7 は次のように動作する.

- 1. Virtual Core Viewer からアノテーション API にコア ID と閲覧状態を送る
- 2. アノテーション API はアノテーション DB から表示するアノテーションのみを抽出する
- 3. 適切な位置に表示されるように座標の計算を行い Virtual Core Viewer に送る

次にアノテーションの編集と削除について図3.8に示す.図3.8は次のように動作する.

- 1. Virtual Core Viewer からアノテーション API にコア ID と編集,削除したNアノテーションの ID を送る
- 2. アノテーション API は現在ログインしている人物が誰であるかをユーザ管理機能に問い合わせる
- 3. ログインしている自分物とメモを作成した人が一致していた場合には操作を許可する.
- 4. アノテーション API はアノテーション DB の内容を書き換える

最後にアノテーションの追加について図3.9に示す.

図 3.9 は次のように動作する.

1. Virtual Core Viewer からアノテーション API にコア ID と編集,追加したいアノテーションの本文を送る

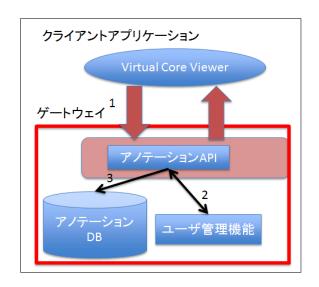


図 3.9: アノテーションの追加

- 2. アノテーション API は現在ログインしている人物が誰であるかをユーザ管理機能に問い合わせる
- 3. ログインしている自分物とアノテーションの本文をアノテーション DB に追加する

アノテーション API ではユーザの閲覧状態に合わせてアノテーションの表示位置を計算している. 例えば,正面から見た時はコアデータの中心にメモが表示されていたとしても,横に90 度回転させると中心から横にずれなければならない(図 3.10). 図 3.10 では左が回転させる前の状況であり,右が90 度回転させた後の状態である.最初は正面についていたメモは右側に移動している.

このなかのユーザ管理機能を筆者が担当した.詳しくは4章に述べる.

#### 3.2.3 開発すべき項目とスコープ

本プロジェクトはイテレーション開発を行う、各イテレーションの内容を次に示す、

• S0: 既存システムの安定化, 品質の向上

最初のイテレーションでは,ユーザフィードバックを元に既存システムの細かい部分の 改善を行い,先行研究システムの理解を深める.筆者は,コアデータに対する色付けを 調整してほしいという顧客の要望に答えるため,レンダラへの変更を行った.

• S1: ブックマーク機能の開発

次のイテレーションでは,ブックマーク機能の開発を行う.筆者は,最初のイテレーションで完了しなかった機能について開発を行った.機能の内容としては,コアデータの横にスケールを表示するというものである.

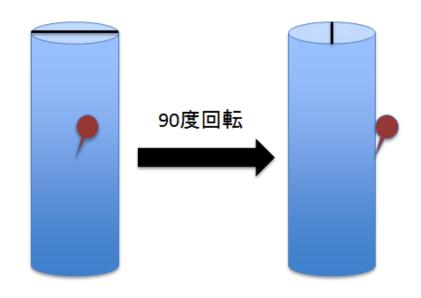


図 3.10: メモの表示位置

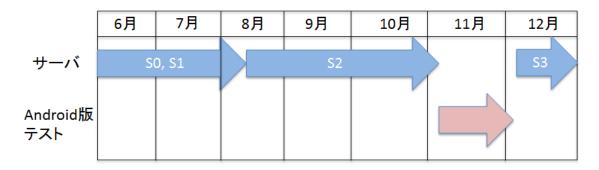


図 3.11: 担当箇所のスケジュール

#### • S2:アノテーション機能の開発

次のイテレーションでは,アノテーション機能の開発を行う.筆者は,アノテーション機能を利用するために必要となるユーザ管理機能を開発した.タブレット端末向けのクライアントアプリケーションのテストも担当した.

#### • S3: 開発した機能の安定化

最後に,本プロジェクトにて開発したシステムを実際に利用してもらい,そのレビューをもとにシステムの動作の安定化と品質の向上を行う.

各イテレーションの詳細を図 3.11 に示す . S0 と S1 を 8 月上旬まで行い , S2 を 11 月の上旬まで , そこから Android 端末向けアプリケーションのテストを 12 月上旬まで行った . 現在は納品の準備を行っている最中である .

本プロジェクトは、開発メンバに筆者を含む筑波大学の学生5名、プロジェクトの担当教員である和田耕一教授と山際伸一准教授、共同研究者でありJAMSTEC 高知コア研究所に所

属する久光敏夫様の合計 8 名により行う. 開発を行う 5 名の担当する機能は次のように分担した.

- Web ブラウザアプリケーションの開発一人 Web ブラウザ版アプリケーションにブックマーク機能とアノテーション機能に対応した ユーザインタフェースを追加する.
- Android 端末向けアプリケーションの開発一人 Android 版アプリケーションにブックマーク機能とアノテーション機能に対応したユー ザインタフェースを追加する.
- ユーザの端末からのリクエストを受け取るサーバの機能の開発二人 ブックマーク機能,アノテーション機能,ユーザ管理機能といった機能をサーバに追加 する.
- 開発したシステムのテストー人 開発したシステムの動作を検証する。

筆者はサーバの機能の開発を行った.その中でもユーザ管理機能の開発を主に行った.また,Androido 端末向けアプリケーションのテストも筆者が行った.

## 第4章 ユーザ管理機能の開発

本章では筆者が開発を担当したユーザ管理機能の概要と設計,実装について述べる.

#### 4.1 機能概要

アノテーション機能ではコアデータの任意の位置にメモを保存することができる.しかし、アノテーション機能だけでは保存されたメモは誰でも操作することができ、他者が残したメモを変更することもできてしまう.そのため、自分が残したメモを他者に勝手に変更されたり、削除されるといった問題が発生する可能性がある.そこで、ユーザ管理機能ではアノテーション機能を利用し保存されたメモが、誰によって作成されたものかを判別することができるようにする.これにより、メモを保存した人しかそのメモを操作できないようにし、他者に自分が残したメモを変更させないようにする.そのため、アノテーションには本文のほかに作成者を判別するための情報が必要となり、ログインしているユーザしかアノテーションを作成することができない.よって、ユーザはアノテーション機能を利用するためにユーザ登録をする必要がある.保存されたアノテーションを閲覧するだけであればユーザ登録をする必要はなく、だれでも閲覧可能である.これにより、保存されたアノテーションを悪意があるユーザから保護することができる.同時に、無意味なメモを大量に生成する、といった荒らし行為からシステムを保護することができる.また、悪意があるユーザがユーザ登録しシステムに対してメモを保存した場合には、システムの運用者がそのユーザを削除することによりメモは表示されなくなる.ユーザ登録には

- メールアドレス (他のユーザには非公開,運用者は見ることができる)
   メールアドレスは重複することがなく唯一のものであるため,ログイン時にユーザ IDとして利用する.
- ユーザネーム (他のユーザに表示される)ユーザネームはアノテーションを保存した際にほかのユーザに表示される名前である。
- パスワード

メールアドレスは Web サイトや論文などに掲載していることが多く,それだけを認証に用いてしまうと誰でもなりすましをすることができるため,メールアドレスと本人だけが知っているパスワードを利用して認証を行う.

の三つが必要である.また,ユーザ登録にはシステム運用者の許可が必要である.これは,悪意のあるユーザがアカウントを作成することを防ぐためである.ユーザ登録からアノテーション機能利用までの流れとしては,

- 1. 三つの情報をフォームに記入し仮登録する
- 2. システム運用者がユーザ登録を許可する
- 3. ユーザは登録時のメールアドレスとパスワードを利用しログインする
- 4. アノテーション機能を利用しコアデータにメモを保存する

となっている.システム運用者はユーザ登録を許可するだけでなく,拒否することもできる. 拒否した場合にはユーザはそのメールアドレスとパスワードではログインすることができない.

#### 4.2 全体構成と開発計画

ユーザ管理機能は,次の機能を持っている.

- ユーザ登録機能
  - ユーザ登録機能はユーザがユーザ登録の申請を行う機能
- ログイン機能 (ユーザ認証機能)

ログイン機能はユーザがログインする際にメールアドレスとパスワードの組が一致しているか確認を行う機能であり、一致していた場合にはセッションを開始する

● ログアウト機能

ログアウト機能はユーザがログアウトする際にセッションを破棄する機能

ユーザ管理画面 (システム運用者向け)

ユーザ管理画面はユーザ登録の許可や拒否,登録済みのユーザを削除しアノテーション機能を利用させなくする,といったことを行うための機能

ユーザがログインしているかどうかはアノテーション API(図 3.6) から参照され , ゲートウェイの内部に実装されている .

ユーザ管理機能は図 4.1 のように構成される.図 4.1 の流れを次に示す.

- 1. ユーザ登録機能ではユーザから受け取った情報を仮登録 DB に保存する
- 2. システムの運用者がユーザ登録を許可機能を利用し許可すると, 仮登録 DB からユーザ 管理 DB に移動される
- 3. ユーザがログインする際にはメールアドレスとパスワードの組によりユーザ認証を行う
- 4. ユーザ認証に成功するとサーバとユーザの端末間にセッションを開始する

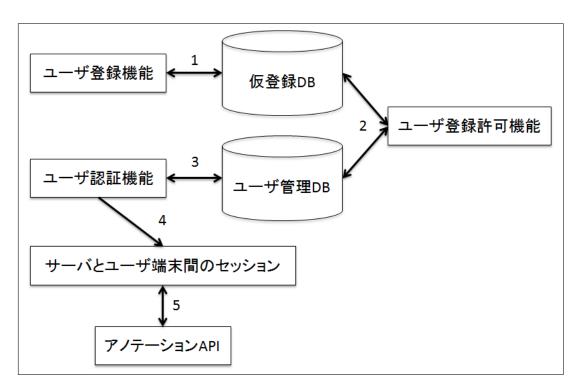


図 4.1: ユーザ管理機能の構成

5. アノテーション API では , サーバとユーザの端末間にセッションが開始されているかどうか , 開始されている場合にはそのユーザ ID は何か , を確認する

ユーザ管理機能ではユーザのメールアドレス,ユーザネーム,パスワードを保存するため, ユーザ管理機能のために新しい二つのテーブルを追加した.一つはユーザ登録を申請し,運 用者からの許可を待っているユーザ情報を保存するためのテーブルである、もう一つはユー ザ登録が完了したユーザ情報を保存するためのテーブルであり,ログイン時のユーザ認証時 に参照される、ユーザ認証機能はログイン時に Web ブラウザと Android 端末アプリケーショ ンの両方から同様にアクセスされる機能である.ユーザ認証に成功するとサーバとユーザの 端末間にセッションを開始する.セッションは一度アクセスしたユーザが次回にアクセスし た際に同一のユーザかどうかを判別するための方法である.セッションを開始する際に,セッ ション ID が発行される.セッション ID はユーザ情報とはなんの関係もなく,毎回ランダム に生成される文字列である、発行されたセッション ID はアクセスしたユーザの端末とサーバ に保存される.ユーザの端末ではクッキーを利用してセッション ID を保存する.次回からの アクセス時にはセッション ID をサーバに送信する . サーバでは , サーバが保存しているセッ ション ID と送信されたセッション ID が一致しているかどうかを確認することにより , 同一 ユーザであるということを判別することができる.アノテーション API からはセッションが 開始されているかどうかにより、アノテーション機能の動作を変更する.開始されている場 合にはメモの保存と同一ユーザが作成したメモを操作することができるようにする.開始さ

れていない場合にはメモの取得のみを可能にする.

ユーザ管理機能はアノテーション機能を公開する際に必要であり、アノテーション機能の動作を確認した後に統合することができるため、アノテーション機能と並列に開発を行う、アノテーション機能とユーザ管理機能をそれぞれ個別に開発し、各動作の確認後にマージし、最後に統合した後の機能の動作確認を行う、具体的に8月から9月までにシステムの設計を行った、9月の初めから機能の開発を行い、9月の中旬にユーザ管理機能単体の動作を確認する、そこから10月までに機能のマージを行い、簡単な動作確認を行う、その後、正確なテストを行い、アノテーション機能とユーザ管理機能が統合できていることを検証する、

#### 4.3 設計

ユーザ管理機能では,前節で述べたユーザ登録機能,ログイン機能,ログアウト機能,ユーザ管理画面の4つの機能を開発する.ユーザ登録機能はユーザが入力した情報を DB に登録することを行う.ログイン機能は DB に登録された情報とユーザがログイン時に入力した情報が一致していることを確認する.ログアウト機能はセッションを終了させるための機能である.ユーザ管理画面はシステムの運用者がユーザの削除やユーザ登録を許可する際に利用する.

また,今回新しく

- ◆ペンディングユーザアカウントテーブルシステム運用者に許可される前のユーザ情報を保存するテーブル
- ユーザアカウントテーブルシステム運用者に許可されたユーザ情報を保存するテーブル

以上の二つのテーブルを追加した.

#### 4.4 実装

次にユーザ管理機能を構成する機能の実装について述べる.

#### 4.4.1 ユーザ登録機能

最初にユーザ登録機能について述べる.ユーザ登録機能では図4.2のようなページを用意しており,Web ブラウザからメールアドレス,ユーザネーム,パスワードの三つをフォームに記入し送信ボタンを押す.送信された3つの情報は,すでに同じメールアドレスが登録されていないかどうかを確認した後,ペンディングユーザアカウントテーブルに保存される.プログラム中ではポストされた文字列を元にSQL文を作成しDBに保存する.この時,SQLインジェクションをさせないために文字列にはサニタイズを行う.

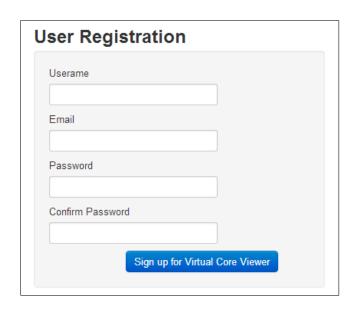


図 4.2: ユーザ登録画面

SQL インジェクションとはアプリケーションのセキュリティ上の不備を意図的に利用し,アプリケーションが想定しないSQL文を実行させ,データベースを不正に操作する攻撃方法である.この攻撃は主にSQL文の断片として解釈できる文字列を入力する事により発生する.SQL インジェクションの例を示す.

SELECT \* FROM table WHERE user='[入力值]';

という SQL 文があったとする.[入力値] はユーザが入力した値に置き換わる.このときユーザが

'; DELETE FROM table WHERE 'a'='a

と入力すると生成される SQL 文は

SELECT \* FROM table WHERE user=' ';DELETE FROM table WHERE 'a'='a';

となり、table テーブルの全てのデータを削除することができる.このように SQL インジェクションには特殊な記号を利用することが多く、これらの文字を無効化する処理を行う必要があり、それをサニタイジングと呼ぶ.サニタイジングは SQL インジェクションに限らず、XSS など html 中に意図しないコードを埋め込まれることを防ぐ場合に利用される.本システムではユーザのメールアドレスやパスワードといった情報しか保存していないが、ユーザのカード情報など、とても重要なデータを保存しているシステムは多数あり、そのような場合にはもっと用心が必要である [10].また、ここでは簡単な例のみの述べたが、SQL インジェクションといっても様々な攻撃手法があり、それらすべてに対応しなければならない [11].

次に,パスワードの保存について述べる.最近ではパスワードの使い回す人が多いという現状があり[12],パスワードが一つ流出したことにより,ほかのサービスでのなりすましなどが可能になることがある.そこで,DBにパスワードを保存する際には,パスワードは登録した本人にしかわからないように暗号化を行う.これにより,仮にDBの中身が漏洩したとし

% php hash.php test a94a8fe5ccb19ba61c4c0873d391e987982fbbd3 % php hash.php test a94a8fe5ccb19ba61c4c0873d391e987982fbbd3

図 4.3: ハッシュ関数の実行結果

% php hash.php test a94a8fe5ccb19ba61c4c0873d391e987982fbbd3 % php hash.php test a94a8fe5ccb19ba61c4c0873d391e987982fbbd3

図 4.4: 乱数とハッシュ関数の実行結果

ても元のパスワードはわからないようにする.また,今回は実装していないが,パスワード を誰かに知られてしまった時にもなりすましを防ぐ方法として、ワンタイムパスワードと呼 ばれる,認証情報が毎回変わる方式もある[13].今回実装する暗号化の方法はパスワードに 対してランダム文字列を付加し、ハッシュ関数を適用するものである、ランダム文字列を付 加する理由として,ハッシュ関数のみしか利用しなかった場合,パスワードを複数人が同じ に設定したときに、そのパスワードがお互いに分かってしまうという問題が発生する、ハッ シュ関数は,入力されたデータを代表する値を生成することができる.ハッシュ関数は非可 |逆変換であり,生成されたデータから元のデータを計算するということは難しい.また,同 じデータが生成されるデータを作成するということも難しい. そのため, ハッシュ関数によ り変換されたデータを誰かに見られても、元のデータを知られることはない、しかし、ハッ シュ関数に同じデータを入力した場合には必ず同じデータが生成されてしまう、図 4.3 に例を 示す,hash.phpというプログラムは引数を元にしたハッシュ値を出力するプログラムであり, 今回ハッシュ関数の例を示すために作成した.図の出力を見ると 'test' というデータを引数に 与えると,2回とも全く同じデータが出力されていることがわかる.このように,同じパス ワードを設定したユーザがいた場合には保存されるデータが同じものになってしまう.この 問題を解決するため、ハッシュ関数を適用する前のパスワードに乱数を付加する.これによ り,元のデータが同じであったとしても生成されるデータは異なるものにできる.図 4.4 に例 を示す, rand\_ hash.php は引数に乱数を付加したデータを元にしたハッシュ値を出力するプロ グラムであり、例を示すために作成した.図4.4の実行結果を見ると引数に 'test' というおな

じデータを与えているが,図 4.3 とは異なり,毎回異なるデータが出力される.このように,パスワードに乱数を付加しハッシュ化されたデータに,使用した乱数を付加しデータベースに保存する.ここで使用した乱数は,ログイン時などのパスワードの認証を行うときに利用するため,パスワードと一緒に保存する必要がある.

#### 4.4.2 ログイン機能

次にログイン機能について述べる.ユーザがログインする際にはメールアドレスとパスワー ドをフォームに入力し送信ボタンを押す、それらの情報を受け取ったサーバは登録されてい るメールアドレスとパスワードの組が一致していることを確認する.ユーザ登録機能と同様 に SQL インジェクションへの対策としてサニタイジングを行う. また, 保存されているパス ワードは暗号化されており,そのまま比較することはできない.さらに,ハッシュ化されて いるため元のパスワードを計算することもできない.そのため,入力されたパスワード文字 列に対して登録時にパスワードを暗号化した時と同じ操作を行う . ハッシュ関数は入力する データが同じであれば,出力されるデータも同じであるため,パスワードが一致していれば, 今回生成したデータと DB に保存されているデータが一致するはずである.暗号化の際に使 用した乱数はパスワードに付加して保存しており、これをユーザが入力したパスワードに付 加し,ハッシュ関数を適用する.これにより生成されたデータと,保存されたデータを比較 し、一致した時に認証は成功したとみなし、ユーザの端末とサーバの間にセッションを開始 する.アノテーション機能はセッションが開始されているかどうかを確認することにより,機 能の動作を切り替えている.セッション ID はクッキーに保存されるため,クッキーの設定を オフにしていると機能できない.多くの Web ブラウザでは標準の機能として動作するため特 にクッキーの設定を行う必要はない. Android 端末向けアプリケーションでは, クッキーを保 持する機能を追加しなければならない.クッキーを扱うための CookieStore クラスが用意され ており,このクラスを利用しセッション ID の管理を行う必要がある.具体的には,セッショ ンを開始する際に CookieStore クラスの変数にサーバから発行されたセッション ID を保持す る.サーバへアクセスする際には保持しているセッション ID を一緒に送信するようにしなけ ればならない.

#### 4.4.3 ログアウト機能

次にログアウト機能について述べる.ログアウト機能では,ログイン機能により開始されたセッションを終了させる.これはサーバに保存されているセッション ID を削除することにより可能である.Android 端末向けアプリケーションでも保存したクッキーを削除する.

#### 4.4.4 ユーザ管理画面

次に,ユーザ管理画面について述べる.ユーザ管理画面用に二つの画面を開発した.一つは仮登録状態のユーザを管理する画面(図 4.5)である.この画面では仮登録を行ったユーザが

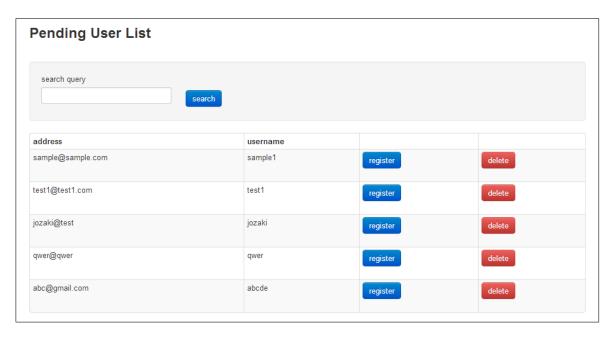


図 4.5: 仮登録ユーザ管理画面

表示されており,ユーザ情報の右にある青いボタンと赤いボタンにより,そのユーザにユー ザ登録を許可,または拒否する事ができる.この画面の実装は,ペンディングユーザアカウ ントテーブルからユーザのメールアドレスとユーザネームのリストを取得し,表示している. 運用者がユーザ登録を許可する,または拒否する,と画面には表示されなくなる.許可した 場合には、ユーザアカウントテーブルにユーザ情報を保存し、ペンディングユーザアカウン トテーブルからはユーザ情報を削除する.拒否した場合にはテーブルからユーザ情報の削除 のみを行う.上のテキストバーに文字列を入力すると,メールアドレスかユーザネームに入 力した文字列が含まれているユーザのみを表示することができる.もう一つの画面はユーザ 登録済みのユーザを管理する画面(図4.6)である.この画面ではユーザ登録済みのユーザが表 示されており,ユーザ情報の右にある青いボタンをおすことにより,ユーザ登録を取り消すこ とができる、仮登録状態のユーザを管理する画面と同様に、データベースからユーザの情報 を取得し表示している.また,この画面でもメールアドレスかユーザネームによるユーザの 検索機能を利用することができる.この画面でユーザ登録を取り消した場合,そのユーザが コアデータに残したメモは表示されなくなる、メモは表示されなくなるだけで、メモの内容 は DB には保存されており,必要になった際にはシステムの運用者は取り出すことができる. 最後に, DB のテーブルの詳細を表 4.1 に示す. 二つのテーブルのフィールドとタイプは同

field	email	name	password
type	varchar(64)	varchar(16)	varchar(256)

表 4.1: ユーザ情報を保存するテーブル

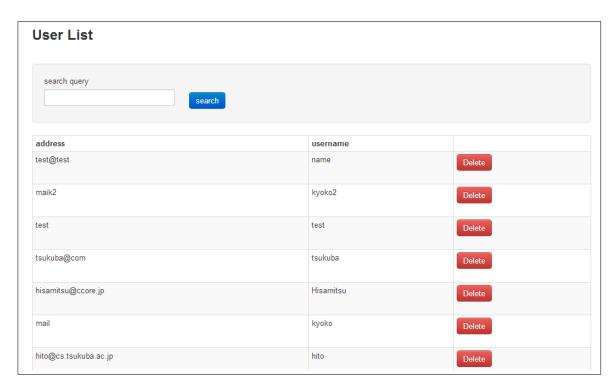


図 4.6: 登録済みユーザ管理画面

じである、メールアドレスが主キーとなり、ユーザネームの重複は可能である、

#### 4.5 まとめ

本章ではユーザ管理機能の構成と実装について述べた.ユーザ管理機能を実装することにより,アノテーション機能により保存されたメモを誰もが操作できるという状態から,メモを保存した人しか操作できないようにすることができるようになる.これは他者に勝手に保存したメモを改竄されないようにするために必要な機能である.これにより,コアデータに保存されたメモを保護することができるとともに,悪意のあるユーザからの,無意味なメモを大量に生成する,といった攻撃からシステムを守ることもできる.ユーザ登録を行うためにはメールアドレスとユーザネームとパスワードが必要であり,これらのユーザ情報はサーバの DB に保存される.特にパスワードに関しては暗号化を行い,もし DB の中身が漏洩したとしても元のパスワードを推測することができないようにした.ログイン時には DB に保存されたユーザ情報と,入力されたユーザ情報が一致していた場合に,セッションを開始する.アノテーション API からはセッションが開始されているかどうかを確認し,その機能の動作を変更させる.セッションが開始していない場合には保存されたメモの閲覧のみを可能にし,開始している場合にはメモの閲覧に加えて,そのユーザが保存したメモの編集と削除,メモを新しく作成する,ことが可能になる.

# 第5章 Android 端末向けクライアントアプリケーションのテスト

本章では, Android 端末向けクライアントアプリケーションのテストの手法とテストの結果と評価を述べる.

#### 5.1 テスト概要

一般に,開発されたソフトウェアは運用される前にテストを行い,仕様書通りに動作するかということを検証する.テストには次の二つの種類がある.

#### • ホワイトボックステスト

ホワイトボックステストは、システムの内部の構造を把握した上で、一つ一つの機能が意図したとおりに動作しているかどうかを検証する方法である。ホワイトボックステストでは、プログラム中の命令や分岐といった内部構造を網羅するようにテストを行う。この、命令や分岐についてどの程度網羅できているかをカバレッジ(網羅率)と呼び、ホワイトボックステストではカバレッジが評価の基準となる。カバレッジにはいくつかの基準がある。それらの基準について図 5.1 の処理を行うプログラムを例に述べる。図 5.1 には分岐が三つあり、分岐 1 が true であった場合に命令 1 が実行される。命令 2 は毎回必ず実行される。分岐 2 が true である場合には命令 3 が実行され、false である場合には分岐 3 の判定が行われる。分岐 3 が true である場合には命令 4 が実行され、false である場合にはそのままプログラムを終了する。図 5.1 に対して分岐網羅を行うには、

#### 命令網羅 (C0)

命令網羅では,すべての命令を一度は実行するようにテストを行う.図 5.1 に対して命令網羅を行うには,図 5.2 の赤線で示されたように実行する二通りのテストを行う必要がある.この二つをテストすることにより命令1 から命令4 までの全ての命令を一度は実行することができる.

#### 分歧網羅 (C1)

分岐網羅では,全ての分岐に対して true と false の両方が一度は実行するようにテストを行う.図 5.3 の赤線で示されたように実行する三通りのテストを行う必要がある.この三つをテストすることにより分岐1から分岐3の各分岐に対して true と false の両方を一度は実行させることができる.

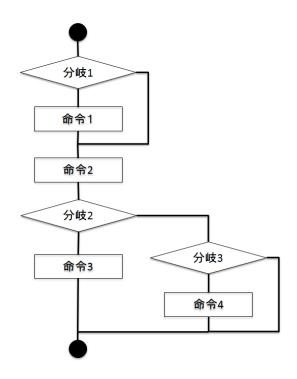


図 5.1: サンプルプログラム

#### 条件網羅 (C2)

条件網羅では,プログラムが動作しうるすべての経路を網羅するテストを行う.図 5.1 に対して命令網羅を行うには,八通りのテストを行う必要がある.

ホワイトボックステストにおいて,全ての経路をテストする条件網羅が理想ではあるが,一般にテストパターンは命令網羅,分岐網羅,条件網羅の順に増える.特に条件網羅では分岐の数が指数としてテストパターン数に影響するため,プログラムの規模がとても小さい場合を除いて,納期内に条件網羅を実現することは,不可能である.そのため,一般には分岐網羅か命令網羅を達成するようにテストを行う.プログラムを解析することにより,自動的に網羅率が高いテストケースを作成するような研究[15]もあり,これを使えば人間がテストケースを作成するよりも正確なテストケースを楽に作成することができる.

#### ブラックボックステスト

ブラックボックステストは,システムの内部構造については気にせず,外部から見て仕様書通りの動作を行っているかどうかを検証する方法である.ホワイトボックステストはプログラムの内部に注目し,意図した動作をしているかどうかを検証する方法であり,必ずしも仕様書通りに動作していることを検証しているわけではない.例えば,

|if(x>0)|

 $\overline{\ \ \ \ \ \ \ \ \ \ }$ という分岐があった時,条件網羅を達成するためには x=-1 と x=1 を試せば良い.

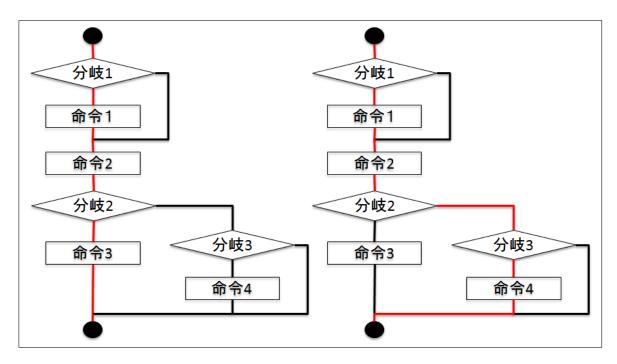


図 5.2: 命令網羅のパターン

しかし、仕様書に x=0 の時には x=1 と同じ動作をする、と書いてあった場合にはプログラムは仕様書通りに動作していないことになる、ブラックボックステストでは、仕様書を元に動作の検証を行う、ブラックボックステストには、次のような方法がある、

#### 境界值分析

境界値分析とは,テストの際の入力値として入力の境界値を入力するものである.境界値分析を行う際には,一緒に同値分割を行う.同値分割では,同じ振る舞いをする入力値を集めた同値クラスというものを作成し,同値クラスの中の一つの値をテストに利用する.この時,有効な同値クラスを有効同値クラス,無効な同値クラスを無効同値クラスと呼ぶ.簡単な例として,あるユーザが 20 歳以上かどうかを判定するようなプログラムを考える.20 歳以上であれば 'OK' が出力され,20 歳未満であれば 'NG' が出力されるとする.このとき,ユーザが入力した値が  $0 \sim 19$  であれば 'NG' が表示されるため  $0 \sim 19$  は同値クラスである.同様に 20以上の数値も同値クラスである.また,人間の年齢は 0 歳からスタートするため,負の値は無効な値であり,無効同値クラスである.境界値分析では同値クラスのうち,境界にある値を代表値としてテストを行う.プログラムのエラーは未満と以下を間違えるなど,分岐の境界で発生することが多く,同値分割よりも多くのバグを発見することができる.先程の例では,'NG'を出力する同値クラスの代表値として 0 と 0 りの値を用いる."OK'を出力する同値クラスの代表値として 0 の値を用いる.無効同値クラスでは代表値として-1 の値を用いる.これにより,開

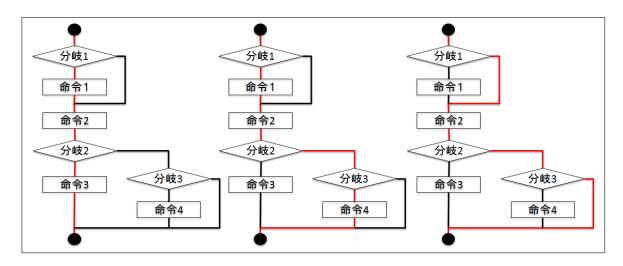


図 5.3: 分岐網羅のパターン

発者のタイプミスや仕様の勘違いによる不具合を検出しやすくなる.

#### デシジョンテーブル

デシジョンテーブルは入力が複数のパラメータから構成される場合に,入力と出力を表にしたものである.先程の例のパラメータに,もう一つ身分証明証を持っているかどうかという入力値を付け加え,プログラムは 20 歳以上かつ身分証明証を持っている場合にのみ 'OK' を出力すると仮定する.すると,デシジョンテーブルは表 5.1 のようになる.

入力	20 歳以上である	0	0	×	×
	身分証明証を持っている	0	×	0	×
出力		OK	NG	NG	NG

表 5.1: ユーザ情報を保存するテーブル

今回筆者が行うテストは,本プロジェクトが開発した機能が仕様書通りに動作していることを検証するためブラックボックステストを行う.今回実装した Android 端末向けアプリケーションはボタンをやスライドバーを操作する操作が多く,いくつかのテキスト入力がある.テキストに関しては境界値分析を行う.

テストの方法は,最初に小さい一つの機能単位での動作の確認を行い,次に,利用シナリオを想定し,その操作を最初から順番に実行することにより行う.例えば,

- 1. アプリを起動
- 2. ログイン
- 3. 閲覧するコア画像の選択

- 4. 視点の移動
- 5. CT 値に対する色付けの変更
- 6. 切断角度の指定とカット
- 7. メモの保存,編集,削除
- 8. ブックマーク機能により共有

といった一連の流れを行う.機能単位での動作確認により,各機能が仕様書通りに動作していることを検証する.利用シナリオに沿った一連の操作を行うことにより,全ての機能をマージした後に機能が連携して動作している,または各機能が排他的に動作している,といったことを検証する.

テストの手順としては,最初に仕様書と画面定義書からテスト項目を作成する.機能単位での動作確認はテスト項目を元にテストを自動化する.一連の流れについては,いくつかのパターンを筆者が手作業で行う.これは,仕様書に沿った動作であることを検証するとともに,ボタンを押しにくい,どこを操作できるかわかりにくい,というようなユーザビリティについての問題を見つけるためである.仕様書に沿った動作をしてなかった機能については不具合として,不具合の修正を行う.

#### 5.2 テスト環境

テストは次の環境にて行った.

• 端末名: Nexus7, Regza Tablet

• OS のバージョン: 2.3 以上

• テストツール: Espresso version1.0

システムのユーザが実際に利用する可能性がある端末のうちの幾つかに開発したアプリケーションをインストールし、それらの端末上にてアプリケーションが仕様書通りに動作していることを確認する。Android 端末には様々な種類があり、それぞれ処理速度や画面のサイズといった性能がバラバラである。Android アプリケーションには一部の端末において発生する不具合といったものも存在する。現在、Android 端末は多くの種類が販売されており、すべての端末にて動作を確認することは困難であるため今回のテストでは、システムの運用者が主に使っている端末と筆者らが主に使っている端末をサンプルにアプリケーションの動作を確認する。

Espresso は google が公開している Android 端末向けアプリケーションの UI テストを自動化 するためのフレームワークであり,本プロジェクトではこのフレームワークを利用した.一般 的な Android アプリケーションの開発では UI のレイアウトを XML に定義し,ボタンやテキ

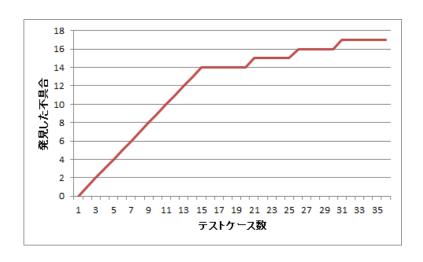


図 5.4: バグ曲線

ストエリアなどのパーツに ID を振り、それらの ID を指定しボタンの動作を定義するといったことを行う. Espresso では、各パーツの ID に対してクリックやテキストの入力、と言った操作を指定することができる. さらに、各パーツにどのような表示がされているかを検証することができる. Espresso を利用し自動化した結果、一つ一つ手作業でテストを行うよりも一回のテストにかかる時間が大きく短縮され、大きな時間の短縮につながった.

#### 5.3 結果と評価

機能単位のテストに関しては各画面毎に全てのボタンのクリックやテキストの入力を個別に検証する.テキストの入力に関しては正常系として境界値の入力,異常系として入力する文字列の長さを最大以上に入力する,といったテストを行う.利用シナリオに沿ったテストでは,全ての機能を一度は利用するようにした.本プロジェクトが開発した Android アプリケーションはイベント駆動型であり,テキストを入力する,ボタンを押すといった操作によりプログラムが動作する.このようなアプリケーションの場合,一つ一つの機能は動作していても実行時エラーによりアプリケーションがクラッシュすることがある.これらの実行時エラーを検証するためにクロウラーベースの GUI テストツール [16] や GUI ripping [17] といった技術を利用して,様々な角度から GUI テストに関する研究が行われている.それでもすべての動作を検証することは困難であり,今回はバグ曲線と呼ばれるテストを評価するための一つの指標を終了条件とした.バグ曲線は横軸にテストケース数,縦軸に見つかった不具合の数のグラフであり,テストケースを重ねるごとに不具合の数は収束する.一般には,発見された不具合の数がある一定の幅以下になった時に終了とする.

今回筆者が行ったテストにより検出した不具合は 19 件であった.図 5.4 の縦軸は検出した不具合の数,横軸は実施したテストケースの数である不具合の数が収束しており,納期日が近いこともあり,結合テストを終了した.

筆者が行ったテストにより, Android 端末向けアプリケーションの担当者が気づかなかった, 仕様書とシステムの差異を修正することができた.これは,ユーザに正常かつ高品質なサービスを提供することにつながったと考えられる.

#### 第6章 おわりに

JAMSTEC が所有する地球深部探査船「ちきゅう」が掘削したコア試料は X 線 CT スキャナを利用してコアデータに変換される.しかし, X 線 CT スキャナは医療用であり,コアデータを閲覧するビューアが異様に特化したものしか存在しない,という問題があった.そこで,先行研究ではコアデータを閲覧するために適したインタフェースを開発した.本プロジェクトでは,先行研究をベースにブックマーク機能とアノテーション機能の二つの機能を開発する「海底コア CT スキャンデ・タ可視化・情報共有を可能とするクラウドサ・ビスの構築」を行った.ブックマーク機能ではコアデータの閲覧状態を保存することができ,さらにその閲覧状態を他者に共有することができる.アノテーション機能ではコアデータの任意の位置にメモを保存することができるようになった.このメモはだれでも閲覧することが可能であり,システム上で他者に自分の意見を発信することができるようになった.これらの機能により,研究者はコアデータに関する議論をシステム上の機能を利用して行うことができるようになった.本報告書にはプロジェクト中の筆者が担当したことと,その貢献について述べた.一つはユーザ管理機能の開発であり,これはアノテーション機能により保存されたメモを他者に変更させないようにすることができる.さらに,システムを悪意があるユーザから保護することができるようにし,ユーザが安心してシステムを利用できるように努めた.筆者はもうーつ,Android 端末向けアプリケーションのテストを行った.Android 端末向けアプリケーショ

更させないようにすることができる.さらに,システムを悪意があるユーザから保護することができるようにし,ユーザが安心してシステムを利用できるように努めた.筆者はもうつつ,Android 端末向けアプリケーションのテストを行った.Android 端末向けアプリケーションのテストでは,アプリケーションが仕様書通りの動作をしているか検証した.アプリケーションのテストの一部は自動化ツールを利用することにより,テストにかかる工数を減らし,できるだけ多くのテストパターンを検証できるよう努めた.結果として,開発を担当した人が気づかなかった仕様書との差異を発見し,アプリケーションの品質を向上させる事ができた.

今後は,アノテーションに文字情報だけでなく,ファイルを添付することができればより良くなると考えられる.コアデータに関連するファイルを添付することにより,それを見たユーザはより理解を深めることができると考えられる.また,本システムの機能はコア試料に特化したものではないため,作成した 3D モデルを公開することができれば,それに対して複数人のユーザが議論を行うことができる.例えば,新製品のモデルなどを作成し,それに対してコメントをつけながら議論を行うといったことが考えられる.また,システムをより使いやすく便利なものにする必要がある.鳥海ら [14] は SNS を活性化させるためには既存のユーザの利用率の増加が重要であると述べており,今後は,本システムの利便性を向上させ,ユーザに多く利用されるシステムにしなければならない.

#### 謝辞

高知コア研究所の久光敏夫様の精力的な協力により,プロジェクトを円滑に進めることができたこと感謝しております,本当にありがとうございました.IPLABの田中二郎教授,嵯峨智准教授,志築文太郎准教授,高橋伸准教授,三末和男准教授,Simona Vasilache 助教には,私が IPLAB に配属されて以来,数多くの指導をしていただき感謝しております.本プロジェクトの担当教員である和田耕一教授,山際伸一准教授には,日々ご指導ご助言をいただき感謝しております.また,山際研究室の秘書である,森田美紀さん,以前秘書であった,櫻井美知代さんには本プロジェクトの事務的なサポートをしていただきました.これまで支えて下さった,先生,友人,家族全員にこの場を借りて感謝申し上げます.

#### 参考文献

- [1] 坂本侑一郎. 海底コア CT スキャンイメージ可視化のためのクラウドサービスの開発. 筑波大学大学院博士課程 システム情報工学研究科特定課題研究報告書, 2013.
- [2] 地球深部探査船「ちきゅう」. http://www.jamstec.go.jp/chikyu/jp/CHIKYU/
- [3] Chester et al. Structure and composition of the plate-boundary slip zone for the 2011 Tohoku-Oki earthquake. Science 6 December, pp.1208-1211, 2013.
- [4] Ujiie et al. Low coseismic shear stress on the Tohoku-Oki megathrust determined from laboratory experiments. Science 6 December, pp.1211-1214, 2013.
- [5] Chester et al. Low coseismic friction on the Tohoku-Oki fault determined from temperature measurements. Science 6 December, pp.1214-1217, 2013.
- [6] Virtual Core Viewer. http://gpgpu.cs.tsukuba.ac.jp/ ccore/guest/index.php(Web 版). https://play.google.com/store/apps/details?id=net.atlantis.coreviewer(Android 版).
- [7] Virtual Core Library. http://www.kochi-core.jp/VCL/.
- [8] International Ocean Discovery Program. http://www.iodp.org/.
- [9] F. Lamberti and A. Sanna. A streaming-based solution for remote visualization of 3D graphics on mobile devices. IEEE Transactions on Visualization and Computer Graphics, 13(2), pp.247-260, 2011.
- [10] W. G. J. Halfond and A. Orso. AMNESIA: analysis and monitoring for NEutralizing SQL-injection attacks. Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering, pp.174-183, 2005.
- [11] W. G. J. Halfond and A. Orso. ABSTRACT A Classification of SQL Injection Attacks and Countermeasures Proceedings of the IEEE international Symposium on Secure software engineering, 2006.
- [12] 平野 亮, 森井 昌克. パスワード運用管理に関する考察および提案とその開発. 電子情報 通信学会技術研究報告. LOIS, ライフインテリジェンスとオフィス情報システム 111(286), 129-134, 2011.

- [13] 辻 貴介, 藤本 卓, 清水 明宏. ワンタイムパスワード認証方式の高速化に関する検討. 電子情報通信学会技術研究報告. OIS, オフィスインフォメーションシステム 103(44), 21-26, 2003.
- [14] 鳥海不二夫, 石田健, 石井健一郎. 小規模 sns のモデル化と活性化シミュレーション. 電子情報通信学会論文誌, Vol. J91-B, pp. 397-406, 4 2008.
- [15] S. Khor and P. Grogono. Using a Genetic Algorithm and Formal Concept Analysis to Generate Branch Coverage Test Data Automatically. Proceedings of the 19th IEEE international conference on Automated software engineering, pp.346-349, 2004.
- [16] D. Amalfitano, A. R. Fasolino and P. Tramontana. A GUI Crawling-based technique for Android Mobile Application Testing. Software Testing, Verification and Validation Workshops (ICSTW), 2011 IEEE Fourth International Conference on, pp.252-261, 2011.
- [17] D. Amalfitano, A. R. Fasolino, P. Tramontana, S. D. Carmine and A. M. Memon. Using GUI ripping for automated testing of Android applications. ASE 2012 Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering, pp.258-261, 2012.

## 付録A 単体テスト項目

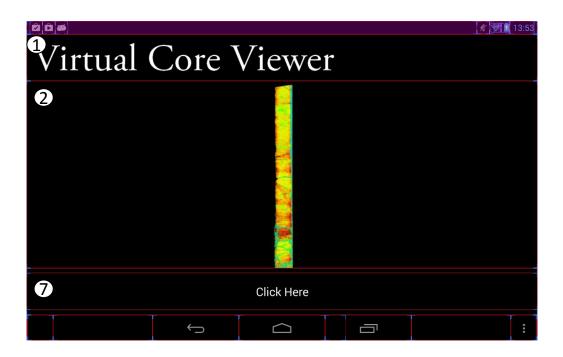
Android繼末向けアプリケーション テスト項目 Activity

Activity	操作対象の番号 操作内容	操作内容	期待する動作	備考
タイトル画面	7	端末がネットワークにつながった状態で click here ボタンを押す		
	7	端末がネットワークにつながっていない状態 で	check network status を表示	
初期コア選択画面 16	16	見るコアを選択しエンターを押す	コアブラウズ 画面へ遷移	スピナーで管理されているため、 存在しないコアは選択できないことを確 認
コアブラウズ画面	2	画面をタップする	特に変化なし	
	2	画面を縦方向にスワイプする(rotateモード)	コアが縦方向に回転した画像に変化する	
	2	画面を横方向にスワイプする(rotateモード)	コアが横方向に回転した画像に変化する	
	2	画面を縦方向にスワイプする(moveモード)	コアが縦方向に移動した画像に変化する	
	2	画面を横方向にスワイプする(moveモード)	特に変化なし	
	2	モード切替ボタンを押す	rotateモードとmoveモードを切り替える	
	9	閲覧情報リセットボタンを押す	閲覧情報がすべて初期状態になる	color や cut もすべて初期状態に戻る
カラ一設定画面	12	APPLYボタンを押す	画像を現在の設定でロードする	CT値を変えて試す
	3	リセットボタンを押す	カラ一設定が初期状態に設定される	
	8	シークバーを操作する	⑦、⑨のテキストの表示が変化する	
	7	テキストの数値を変更する	⑧のシークバーの位置が変化する	のより小さな値
	6	テキストの数値を変更する	⑧のシークバーの位置が変化する	⑦より小さな値
カット画面	1	画面をタップする	特に変化なし	
	1	画面を縦方向にスワイプする(rotateモード)	コアが縦方向に回転した画像に変化する	モードはコアブラウズ画面と共通
	1	画面を横方向にスワイプする(rotateモード)	コアが横方向に回転した画像に変化する	
	1	画面を縦方向にスワイプする(moveモード)	コアが縦方向に移動した画像に変化する	
	1	画面を横方向にスワイプする(moveモード)	特に変化なし	
	4	バーを操作	③のコアが回転した画像になる	
	2	ボタンを押す	4の値に応じた角度でコア画像をカットする	
	9	ボタンを押す	カットしてない状態に戻る	
アノテーション画画	1	アノテーションマークをクリックする	②に表示されるアノテーションがクリックした(⑦のリストの番号も変化する	⑦のリストの番号も変化する
			位置についているものに変化する	
	1	その他の場所をクリックする	特に変化なし	
	5	ボタンをクリックする	メモ編集画面を表示する	ログインしているユーザがつけた アノテーションのみ表示
	9	ボタンをクリックする	メモ削除画面を表示する	ログインしているユーザがつけた アノテーションのみ表示
	7	リストのアイテムをクリックする	②に表示されるアノテーションが 選択! た メモに 変化する	①の赤くなる部分も変わる

コア選択画面	1	画面をタップする	特に変化なし	
	1	画面を縦方向にスワイプする(rotateモード)  :	コアが縦方向に回転した画像に変化する	モードはコアブラウズ画面と共通
	1	画面を横方向にスワイプする(rotateモード)  :	コアが横方向に回転した画像に変化する	
	1	画面を縦方向にスワイプする(moveモード)	コアが縦方向に移動した画像に変化する	
	1	画面を横方向にスワイプする(moveモード)	特に変化なし	
	4, 6, 8, 10,	閲覧したいコアを選択することができる		
	12, 14, 16			
	17	画像を選択されたものに変更する	画像が変化する	
ログイン画面	2	0文字	一全工	文字数
	2	文字		文字数
	2	2000文字(最大)		文字数
	2	2001文字(最大+1)	一全工	文字数
	2	各種記号 ″!@#\$%^&*()_+'″□¥./,<>`~  :"?″		禁則文字チェック
	2	<meta <="" http-equiv="Refresh" td=""/> <td></td> <td></td>		
		CONTENT="5; URL=http://google.com/"/>		スクリプト
	4	0文字	一全工	文字数
	4	1 文字		文字数
	4	2000文字(最大)		文字数
	4	2001文字(最大+1)	一	文字数
	4	各種記号 ″!@#\$%^8*() +'″□¥./.<>`~{  :″?″		禁則文字チェック
	4	CONTENT="5; URL=http://google.com/"		
		<u> </u>		スクリプト

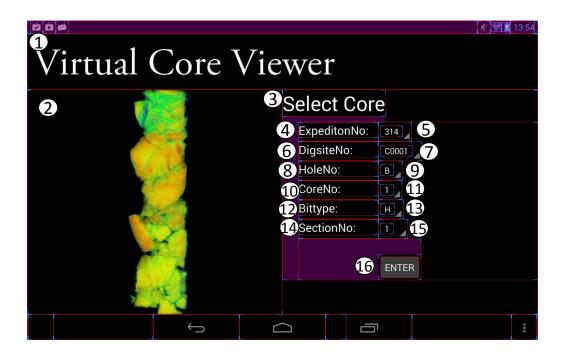
## 付 録 B 画面定義書

### 1. タイトル 画面



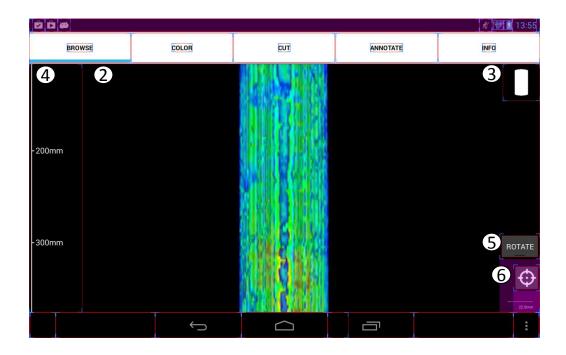
_				
No	画面名	レイアウトファイル	部品名	ID
1	startScreen	start_screen.xml	ImageView	start_title_image
2	startScreen	start_screen.xml	ViewFlipper	start_flipper
3	startScreen	start_screen_first	ImageView	start_firstlayout
4	startScreen	start_screen_second	ImageView	start_secondlayout
5	startScreen	start_screen_third	ImageView	start_thirdlayout
6	startScreen	start_screen_fourth	ImageView	start_fourthlayout
7	startScreen	start_screen.xml	TextView	start_click_text

### 2. 初期コア選択 画面



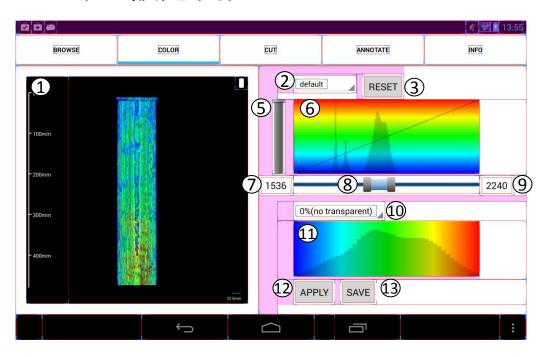
No.	画面名	レイアウトファイル	部品名	ID
1	coreSelectScreen	core_select_screen.xml	ImageView	select_title_image
2	coreSelectScreen	core_select_screen.xml	ImageView	select_core_image
3	coreSelectScreen	core_select_screen.xml	TextView	select_message_text
4	coreSelectScreen	core_select_screen.xml	TextView	select_seano_text
5	coreSelectScreen	core_select_screen.xml	Spinner	select_seano_spinner
6	coreSelectScreen	core_select_screen.xml	TextView	select_digsite_text
7	coreSelectScreen	core_select_screen.xml	Spinner	select_digsite_spinner
8	coreSelectScreen	core_select_screen.xml	TextView	select_hole_text
9	coreSelectScreen	core_select_screen.xml	Spinner	select_hole_spinner
10	coreSelectScreen	core_select_screen.xml	TextView	select_coreno_text
11	coreSelectScreen	core_select_screen.xml	Spinner	select_coreno_spinner
12	coreSelectScreen	core_select_screen.xml	TextView	select_bittype_text
13	coreSelectScreen	core_select_screen.xml	Spinner	select_bittype_spinner
14	coreSelectScreen	core_select_screen.xml	TextView	select_sectionno_text
15	coreSelectScreen	core_select_screen.xml	Spinner	select_sectionno_spinner
16	coreSelectScreen	core_select_screen.xml	Button	select_select_button

# 3. コアブラウズ 画面



No.	. 画面名	レイアウトファイル	部品名	ID
1	browseScreen	browse_screen.xml	FrameLayout	browse_framelayout
2	browseScreen	browse_screen.xml	TouchCoreView	browse_touchcoreview
3	browseScreen	browse_screen.xml	CoreGLView	browse_myglview
4	browseScreen	browse_screen.xml	ScaleView	browse_scale
5	browseScreen	browse_screen.xml	ToggleButton	browse_dragaction_toggle
6	browseScreen	browse screen.xml	ImageButton	browse reset button

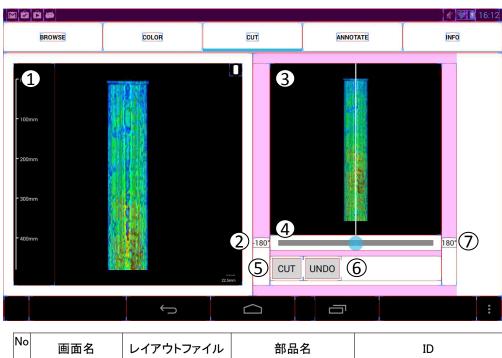
## 4. カラー設定 画面



No.	画面名	レイアウトファイル	部品名	ID
1	colorScreen	color_screen.xml	include(core_view_for_ope ration)	color_touchcore
2	colorScreen	color_screen.xml	Spinner	color_colormodel_spinner
3	colorScreen	color_screen.xml	Button	color_reset_button
4	colorScreen	color_screen.xml	TextView	color_dummy_text
5	colorScreen	color_screen.xml	VerticalSeekBar	color_histgramfrequency_verti calseekbar
6	colorScreen	color_screen.xml	ColorMapView	color_color_mapview
7	colorScreen	color_screen.xml	EditText	color_ctmin_edit
8	colorScreen	color_screen.xml	RangeSelectBar	color_histgramrange_selectbar
9	colorScreen	color_screen.xml	EditText	color_ctmax_edit
10	colorScreen	color_screen.xml	Spinner	color_alpha_spinner
11	colorScreen	color_screen.xml	ToukaMapView	color_alpha_toukamapview
12	colorScreen	color_screen.xml	Button	color_apply_button
13	colorScreen	color_screen.xml	Button	color_save_button

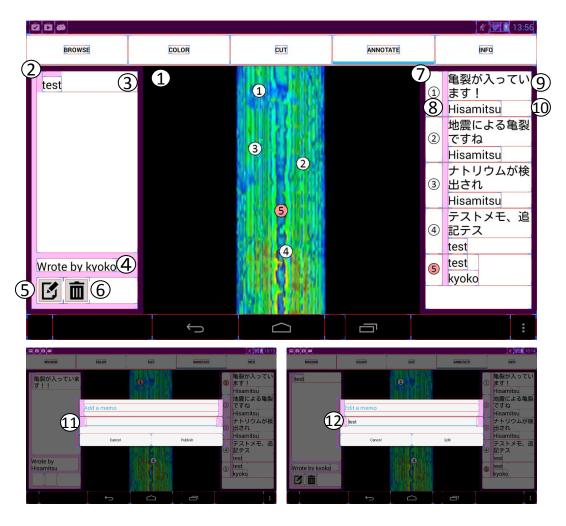
No.	画面名	レイアウトファイル	部品名	ID
1		core_view_for_operatio n.xml		coreview_operate
2		core_view_for_operatio n.xml		myGLView
3	include	core_view_for_operatio n.xml	ScaleView	touch_scale

## 5. コアカット 画面



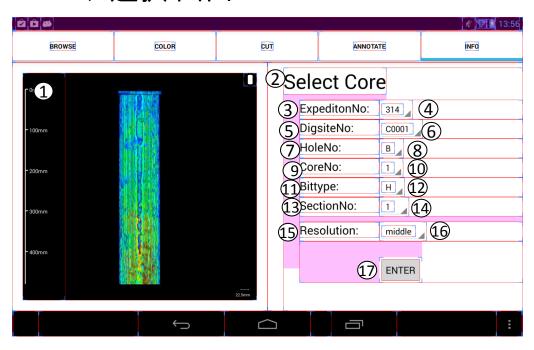
No	画面名	レイアウトファイル	部品名	ID
1	cutScreen	cut_screen.xml	Include(core_view_for_o peration)	cut_touchcore
2	cutScreen	cut_screen.xml	TextView	cut_minangle_text
3	cutScreen	cut_screen.xml	CutCoreView	cut_touchcutcoreview
4	cutScreen	cut_screen.xml	SeekBar	cut_angle_seekbar
5	cutScreen	cut_screen.xml	Button	cut_cut_button
6	cutScreen	cut_screen.xml	Button	cut_undo_button
7	cutScreen	cut_screen.xml	TextView	cut_maxangle_text

### 6. アノテーション 画面



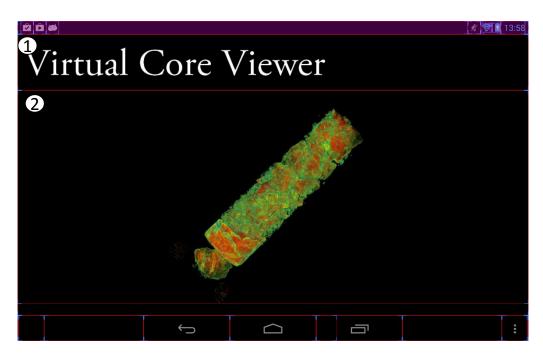
No.	画面名	レイアウトファイル	部品名	ID
1	annotateScreen	annotate_screen.xml	include(core_view_for_ann otation)	annotate_coreview
2	annotateScreen	annotate_screen.xml	ScrollView	annotate_scrollView
3	annotateScreen	annotate_screen.xml	TextView	annotate_text_text
4	annotateScreen	annotate_screen.xml	TextView	annotate_author_text
5	annotateScreen	annotate_screen.xml	ImageButton	annotate_edit_button
6	annotateScreen	annotate_screen.xml	ImageButton	annotate_delete_button
7	annotateScreen	annotate_screen.xml	ListView(custom)	annotate_title_list
8	annotateScreen	annotation_list.xml	ImageView	annotate_list_image
9	annotateScreen	annotation_list.xml	TextView	annotate_list_title_text
10	annotateScreen	annotation_list.xml	TextView	annotate_list_author_text
11	annotateScreen	xml		add_annotate_dialog_text_edit
12	annotateScreen	edit_annotation_dialog. xml	EditText	edit_annotate_dialog_text_edit

## 7. コア選択 画面



No.	画面名	レイアウトファイル	部品名	ID
1	infoScreen	info_screen.xml	include(core_view_for_ope ration)	info_touchcore
2	infoScreen	info_screen.xml	TextView	info_message_text
3	infoScreen	info_screen.xml	TextView	info_seano_text
4	infoScreen	info_screen.xml	Spinner	info_seano_spinner
5	infoScreen	info_screen.xml	TextView	info_digsite_text
6	infoScreen	info_screen.xml	Spinner	info_digsite_spinner
7	infoScreen	info_screen.xml	TextView	info_hole_text
8	infoScreen	info_screen.xml	Spinner	info_hole_spinner
9	infoScreen	info_screen.xml	TextView	info_coreno_text
10	infoScreen	info_screen.xml	Spinner	info_coreno_spinner
11	infoScreen	info_screen.xml	TextView	info_bittype_text
12	infoScreen	info_screen.xml	Spinner	info_bittype_spinner
13	infoScreen	info_screen.xml	TextView	info_sectionno_text
14	infoScreen	info_screen.xml	Spinner	info_sectionno_spinner
15	infoScreen	info_screen.xml	TextView	info_resolution_text
16	infoScreen	info_screen.xml	Spinner	info_resolution_spinner
17	infoScreen	info_screen.xml	Button	info_select_button

## 1. Title 画面



No.	画面名	レイアウトファイル	部品名	ID
1		intent_receive_screen. xml		intent_title_image
2		intent_receive_screen. xml		intent_flipper
3		intent_receive_screen. xml		intent_firstlayout
4		intent_receive_screen. xml		intent_secondlayout
5		intent_receive_screen. xml		intent_thirdlayout
6	intentScreen	intent_receive_screen. xml	ImageView	intent_fourthlayout

## 8. 設定 画面



No.	画面名	レイアウトファイル	部品名	ID
	Screen	delete_colormodel_scr een.xml		delete_colormodel_title_text
	Screen	delete_colormodel_scr een.xml		delete_colormodel_list
3	deleteColormodel Screen	delete_colormodel_scr een.xml	TextView	delete_colormodel_back_text

## 9. ログイン 画面



No.	画面名	レイアウトファイル	部品名	ID
1	preferenceDialog	login_dialog.xml	TextView	login_dialog_mail_text
2	preferenceDialog	login_dialog.xml	EditText	login_dialog_mail_edit
3	preferenceDialog	login_dialog.xml	TextView	login_dialog_password_text
4	preferenceDialog	login_dialog.xml	EditText	login_dialog_password_edit
5	preferenceDialog	login_dialog.xml	TextView	login_dialog_createaccount_text