최종 프로젝트 결과 보고서

**< 2D그래픽스를 활용한 뷰어 >**

|  |  |
| --- | --- |
| 팀원 | 정재현 (팀장) |
| 정유나 (PM) |
| 조재영 |
| 한은지 |

**One By One**

본 프로그램은 의사의 시점으로 맞춰진 치과 전용 2D 뷰어 프로그램입니다.

치아 진료가 필요한 환자로부터 치의에 대한 정보와 시술에 필요한 정보를 시각화하여 보여주어 진료의 방향을 설명할 수 있고, 의사의 담당 환자의 치아 사진을 수정, 저장함으로 프로젝트를 통해 치료의 도움이 될 수 있도록 제작되었습니다.

목 차

1. 프로젝트 방향 --------------------------------------------------------------- 4
2. 개발 배경 및 목적(개요) ------------------------------------------------- 4
   * 1. 배경
     2. 내용
     3. 이유
     4. 목적
     5. 핵심 경쟁력
3. 개발 환경 및 개발 언어 ------------------------------------------------- 5
4. 개발 기술 ------------------------------------------------------------- 6
5. 시스템 구성 및 아키텍처 ------------------------------------------------ 7
6. 파트 분배 ------------------------------------------------------------------- 8
   * 1. 개발 일정 , 프로젝트 구축 일정 ------------------------------------------ 8
7. 주요 기능 및 기술 설명 ----------------------------------------------------- 9
   1. UML------------------------------------------------------------------- 9
   2. Flow Chart--------------------------------------------------------------10
   3. Server & Client Network ------------------------------------------------ 11
      * + 1. Node.js
          2. 서버 / 클라이언트 연동
          3. MongoDB Atlas View Monitering
   4. 2D Viewer-------------------------------------------------------------- 19
      * + 1. Log In / Log Out
          2. Layout
          3. Draw
          4. Image Processing
          5. Key Features
          6. ToolBar
8. 기대 효과 및 활용 분야 ---------------------------------------------------- 51
   1. 기대효과
   2. 발전 방향

**Ⅰ. 프로젝트 방향**

1. **개발 배경 및 목적 (개요)**
2. 개발 배경

* 디지털 이전 시대의 문제점 개선.
* 치과 의사들이 환자를 진단하고 수기로 메모했을 때의 문제점을 개선하기 위해 제작.
* 연필, 볼펜 같은 문구류들은 깨끗한 수정이 어렵기에 PC 상에 수정할 수 있는 기능 탑재.
* 종이 문서 같은 훼손이 쉬운 문서를 PC의 파일 및 데이터 형태로 송/수신.

1. 개발 내용

* 이미지 확대, 축소, 이동 : 이미지의 치아 및 구강의 자세한 위치나,

형태를 알아보기 위한 뷰어 기능

* 임플란트 식립 : 임플란트 이미지를 로드 후 회전 및 대칭이동하여 구체적인 설계에 도움
* 원본 / 처리 이미지를 저장하는 기능
* OpenCV를 이용하여 이미지 프로세싱을 하여 원하는 이미지 결과를 생성
* Draw(브러쉬, 도형, 메모, 커서) 기능으로 진료 시, 환자가 쉽게 이해하도록 도움
* 각도, 길이 측정 시 수학공식을 이용하여 실제 길이 및 각도를 측정 기능
* 의사별 진단, 설계 필요정보를 기억하여 빠르게 호출하는 기능
* 서버로부터 이미지를 다운로드 및 업로드하는 통신 및 멀티태스킹 기능

1. 개발 이유

* 서버를 운용한 멀티태스킹으로 PC 내에서의 정보뿐만이 아닌 외부 서비스와

공유하기 위함.

* 뷰어에 사용되는 Layout, Draw, Processing, 을 통해 필요한 상황에 맞게 환자들의 구강 정보를 기록 및 저장하기 위함.

1. 개발 목적

* 환자나 의사에게 보다 쉽고 편리하게 보여줄 수 있는 2D 영상 뷰어를

제공하기 위해 제작.

* 편의상 키보드 보다 마우스 사용의 비중을 기울여 조작이 쉬운 프로그램을 제작.
* 사용자(치과 의사)의 편의를 위한 영상 툴 편집 기능을 제작.
* 의사 입장에서 환자들의 정보와 해당 환자의 이미지 정보를 알아보기 위해 제작.

1. 핵심 경쟁력

* 환자들에게 진단 정보를 시각적으로 보이도록 Qt GUI Framework를 사용
* 진단 정보를 문서화함으로써 보관 및 보안 용이
* 치과의사는 빠르고 정확하면서 편리하게 진료할 수 있도록 개선
* 치과 진료는 디지털 투명교정, 디지털 임플란트, 등 디지털화 시대에 맞춰 개발

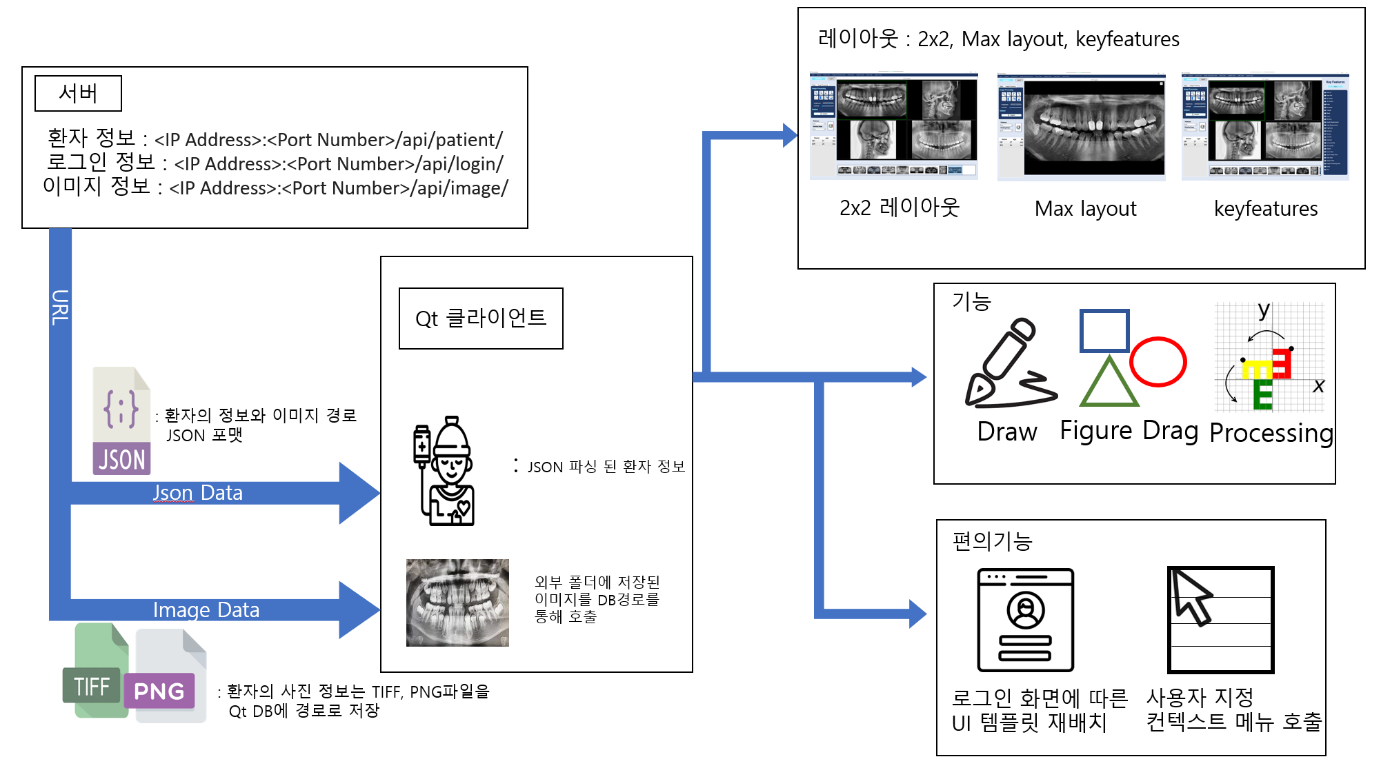
1. **개발 환경 및 개발 언어**

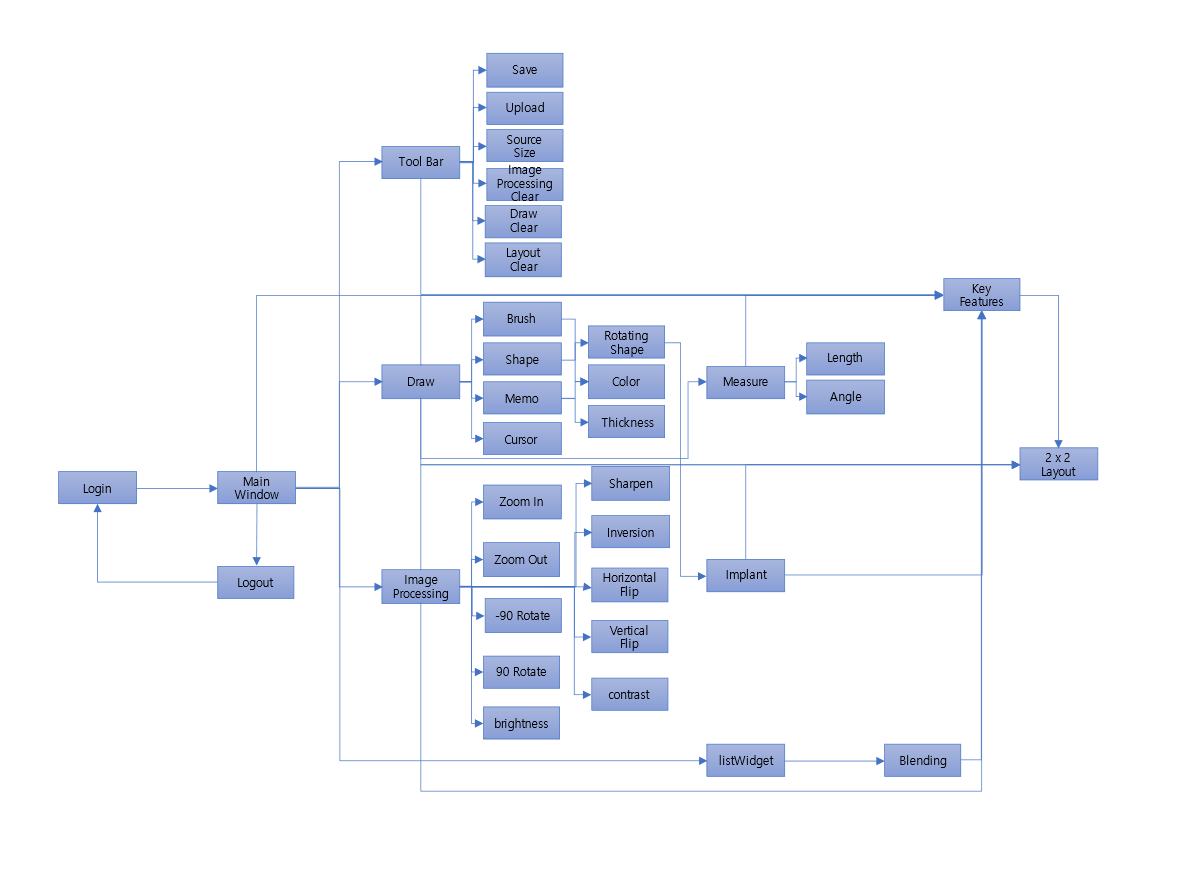
|  |  |
| --- | --- |
|  | * Qt Creator는 GUI 애플리케이션 개발을 단순화하는 크로스 플랫폼 및 C++, JavaScript 및 QML 통합 개발 환경입니다. * 그래픽 사용자 인터페이스를 사용하는 경우에는 Qt를 위젯 툴킷으로 분류합니다. * Qt Creator에서는 다음과 같은 서비스를 지원합니다. * 복합적인 코드 편집기 (code editor) * 통합적 UI 디자인 * 프로젝트 및 빌드 관리 * 다양한 시스템 지원 |
|  | * Visual Code는 마이크로소프트가 마이크로소프트 윈도, macOS, 리눅스용으로 개발한 소스 코드 편집기입니다. * Node.js 자바스크립트와 MongoDB를 연동하기 위한 소스 코드 편집기 용도로 사용되었습니다. |
|  | * C++은 기존의 C언어에 여러 가지 기능을 추가하여 만든 프로그래밍 언어입니다. * C++은 C언어에서 절차 지향적 언어의 특징을 가져왔을 뿐만 아니라, 클래스를 사용하는 객체 지향적 언어인 동시에 템플릿으로 대변되는 일반화 프로그래밍 방식의 언어이기도 합니다. |
|  | * JavaScript는 객체 기반의 스크립트 프로그래밍 언어입니다. * JavaScript는 웹 브라우저 내에서 주로 사용되며, 다른 * 응용 프로그램의 내장 객체에도 접근할 수 있는 기능을   가지고 있습니다.   * Node.js와 같은 런타임 환경과 같이 서버 프로그래밍에도   사용하고 있습니다. |

1. **개발 기술**

|  |  |
| --- | --- |
|  | * node.js는 확장성 있는 네트워크 애플리케이션   개발에 사용되는 소프트웨어 플랫폼입니다.   * 작성 언어로 자바스크립트를 활용하며 논 블로킹 I/O와   단일 스레드 이벤트 루프를 통한 높은 처리 성능을  가지고 있습니다. |
|  | * PostMan은 개발자들이 API를 디자인하고 빌드하고 테스트하고 반복하기 위한 API 플랫폼입니다. * 환자 정보, 이미지 정보 송수신 테스트 플랫폼이며, 환자 정보 컨트롤 시 사용되는 API 플랫폼입니다. |
|  | * MongoDB는 크로스 플랫폼 도큐먼트 지향 데이터베이스 시스템이다. * NoSQL 데이터베이스로 분류되는 몽고 DB는 JSON과 같은 동적 스키마 형 도큐먼트들을 선호함에 따라 전통적인 테이블 기반 관계형 데이터베이스 구조의 사용을 하지 않습니다. |
|  | * SQLite는 MySQL 과 같은 데이터베이스 관리 시스템이지만, 서버가 아니라 응용 프로그램에 넣어 사용하는 비교적 가벼운 데이터베이스입니다. * 대규모 작업에는 적합하지 않지만, 중소 규모라면 속도에   문제가 없습니다. |
|  | * OpenCV은 실시간 컴퓨터 비전을 목적으로 한 프로그래밍 라이브러리입니다. 실시간 이미지 프로세싱에 중점을 둔 라이브러리로 윈도, 리눅스 등에서 사용 가능한 크로스 플랫폼이며 오픈소스 BSD 허가서 하에서 무료로 사용할 수 있습니다. |

1. **시스템 구성도 및 아키텍처**

* 시스템 구성도
* 아키텍처



**Ⅱ. 파트 분배**

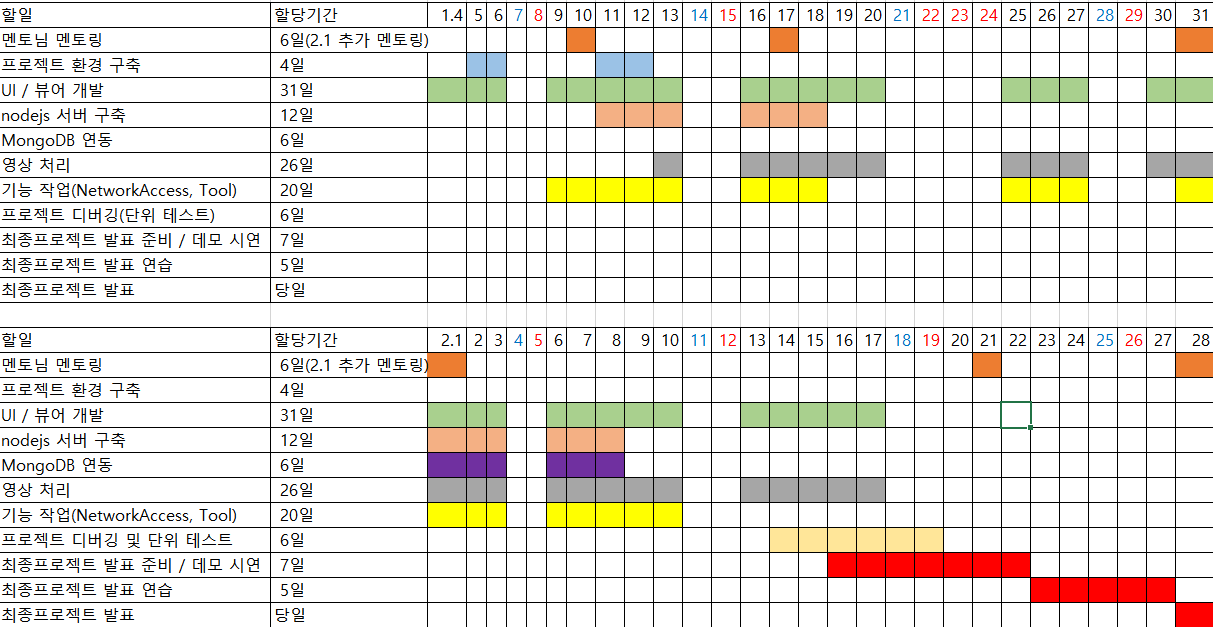
**1) 개발 일정, 개발역할, 프로젝트 구축 일정**

* 개발 기간 : 2023.01.04 (수) ~ 2023.02.28(화), 2개월
* 개발 역할

- 정재현 : 서버 구성, 서버 / 클라이언트 통신, 이미지 내부 길이 / 각도 측정  
알파 블렌딩   
- 정유나 : 프로그램 GUI 설계, 로그인/로그아웃, Layout 정렬 및 최대화, 이미지 확대, 패닝, 축소, -90 º/ +90 º 회전, 임플란트 식립 기능, 원본 크기, View 초기화, Layout 초기화, Key Features 편의 기능 발표  
- 한은지 : 브러시, 셰이핑(도형), 안티 에일리어싱, 펜 굵기 조정, 색상 조정, 도형 회전, 메모 기능, 커서, 드로우 초기화

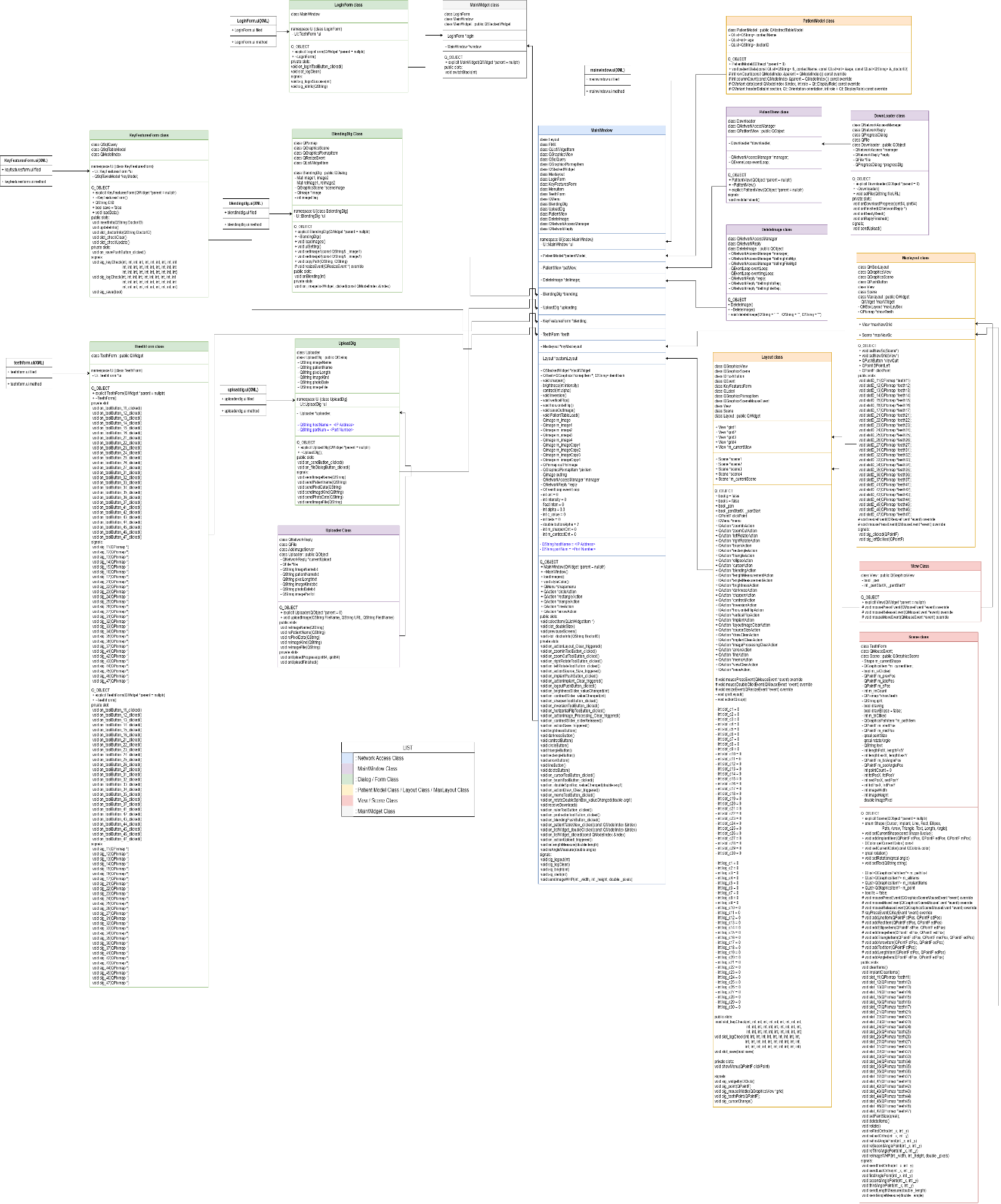
- 조재영 : 밝기 / 어둡기, 명암, 대비(색 반전), 선명화, 수직, 수평 반전, 프로세싱 초기화, 처리 이미지 저장

* 프로젝트 구축 일정



**Ⅳ. 주요 기능 및 기술 설명**

1. **UML**

첨부파일 :

1. **Flow Chart**

|  |  |  |  |
| --- | --- | --- | --- |
| 환자 정보 탐색 순서도 | 이미지 업로드 순서도 | | 이미지 정보/파일 다운로드 순서 |
| 이미지 정보 / 파일 삭제 순서 | | 이미지 픽셀 당 길이 값, 너비, 높이 탐색 순서도 | |

1. **Server & Client Network**
2. Node.js
3. 구성 (Directory / Files)

|  |  |
| --- | --- |
|  | * **서버 디렉터리 및 파일 설명** * **Controllers**   : 환자, 이미지, 로그인 정보를 입력, 수정, 저장, 삭제가 가능한 디렉터리  → ImageController.js : Image 정보 컨트롤 파일 → LoginController.js : Login 정보 컨트롤 파 → PatientController.js : Patient 정보 컨트롤 파일   * **Middleware**  : 서버상에 업로드되는 이미지를 uploads 폴더로 동작하게 하는 디렉토리   → upload.js : 클라이언트로부터 받는 이미지를 처리하는 파일   * **Models**  : 환자, 이미지, 로그인 정보 스키마 디렉터리   → Image.js : Image 스키마 설정 파일  → login.js : login 스키마 설정 파일  → Patient.js : Patient 스키마 설정 파일   * **Node\_modules**  : Node.js 에서 설정된 모듈 디렉터리 * **Routes**  : 환자, 이미지, 로그인 정보를 처리하는 Url을 정의하는 발송 디렉터리   → image.js : Image controller Url 설정 파일  → login.js : Image controller Url 설정 파일  → patient.js : Image controller Url 설정 파일   * **Uplads** : 클라이언트에서 서버로 업로드하는 이미지들을 저장하는 디렉터리 * **package.json, Package-lock.json**  : Node.js에서 실행한 패키지 모음을 Json으로 포맷한 파일 * **server.js**  : 해당 파일을 실행하면 서버를 활성화하는 파일 * **variables.env**  : MongoDB에서 제공된 URL 정보를 변수로 저장한 환경 변수 파일 |

1. Node.js와 MongoDB 연동 / Server 활성화

|  |  |
| --- | --- |
|  | MongoDB 홈페이지에서  Create 버튼으로 Database를  생성합니다. |
|  | 생성된 데이터 베이스의  NetworkAccess 항목 선택 후, 클라이언트에서 연동할  IP Address를 재설정합니다. |
|  | 클라이언트에서 Database를 연동할 URL을 받기 위해  connect 버튼에서 application 항목의 application code를  복사합니다. |
|  | Variable.env 파일 내부에 복사한 URL을 PATIENTDB\_URL이라는 변수명으로 환경 변수를 설정합니다. |
|  | node 명령어로 서버를 실행하면 설정한 IP 주소와 Port 번호에 맞게 서버를 활성화할 수 있습니다. |

1. 서버 / 클라이언트 연동 계략도

|  |
| --- |
|  |
| MVC (MODEL, CONTROLLER, VIEW) 패턴을 기반으로 세 기능을 나누었습니다.  MVC란 CONTROLLER를 조작하면 MODEL의 데이터를 가져오고  MODEL에서 가져온 데이터를 VIEW를 통해 사용자에게 환자 정보와  이미지 정보를 보여주는 개발방법론을 적용 하였습니다.  서버를 구성하는 node.js와 MongoDB를 병합한 서버 네트워크를 구성함으로  Qt에서 제공하는 네트워크 클래스로 서버의 URL을 접근 후 환자와 이미지  정보를 송수신 할 수 있습니다.  서버와 클라이언트를 연동하는데 MongoDB Atlas로 활성화 하기 때문에  클라이언트와 서버상에 방화벽을 해제할 필요는 없습니다. |

1. 서버 / 클라이언트 연동
   * 1. 환자 정보 다운로드

|  |  |  |
| --- | --- | --- |
|  | 프로그램을 동작하면 마주하는 로그인 화면에 의사의 아이디를 맞게 입력하면  환자 정보 URL로 접근하여 경로가 맞는지 확인합니다. | |
| URL에 저장된 환자 정보를 모두 탐색할 때까지 의사 아이디와 일치하는 환자 정보만 찾도록 반복합니다. | | | |
|  | | 경로가 맞으면 입력한 의사 아이디에 해당하는 환자의 정보만 추출하여 테이블에 출력합니다.  해당 환자 정보를 모두 탐색하면 URL 연결을 마치고 탐색을 종료합니다. | |

* + 1. 이미지 업로드

|  |  |
| --- | --- |
|  | |
|  | 이미지 업로드 전 환자 이미지 리스트 |
|  | 1. 툴바(Tool Bar)의 업로드 버튼을 클릭 시 다이얼로그 호출을 합니다.  2. 업로드 다이얼로그를 호출 여부 메시지 박스가 나오면 yes 버튼을 눌러 업로드  다이얼로그를 호출합니다.  3. 업로드 다이얼로그 내부의 6개의 데이터를 입력합니다.  마지막 입력 데이터인 Image File은  업로드하고자 하는 이미지 경로를 찾는 file dialog 버튼으로 이미지를 찾고  send 버튼을 눌러 서버에 이미지를 업로드합니다. |
|  | 이미지 업로드가 완료됨을 listWidget으로  확인할 수 있습니다. |

* + 1. 이미지 다운로드

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | 초기 listWidget 화면으로  어떤 이미지도 다운로드하지 않은 상태입니다. |
|  | | 환자 테이블의 이름을 클릭 시 이미지 정보 Url을 탐색합니다. |
|  | |  |
|  | |  |
|  | | 이미지 정보에 해당 환자 이름이 있으면 이미지를 다운로드하고 listWidget에 이미지를 출력합니다.  환자 테이블의 이름에 맞는  이미지 정보와 이미지를 다운로드합니다. |
|  | 이미지 정보의 끝까지 탐색한 후 완료하면 탐색을 종료합니다. | |

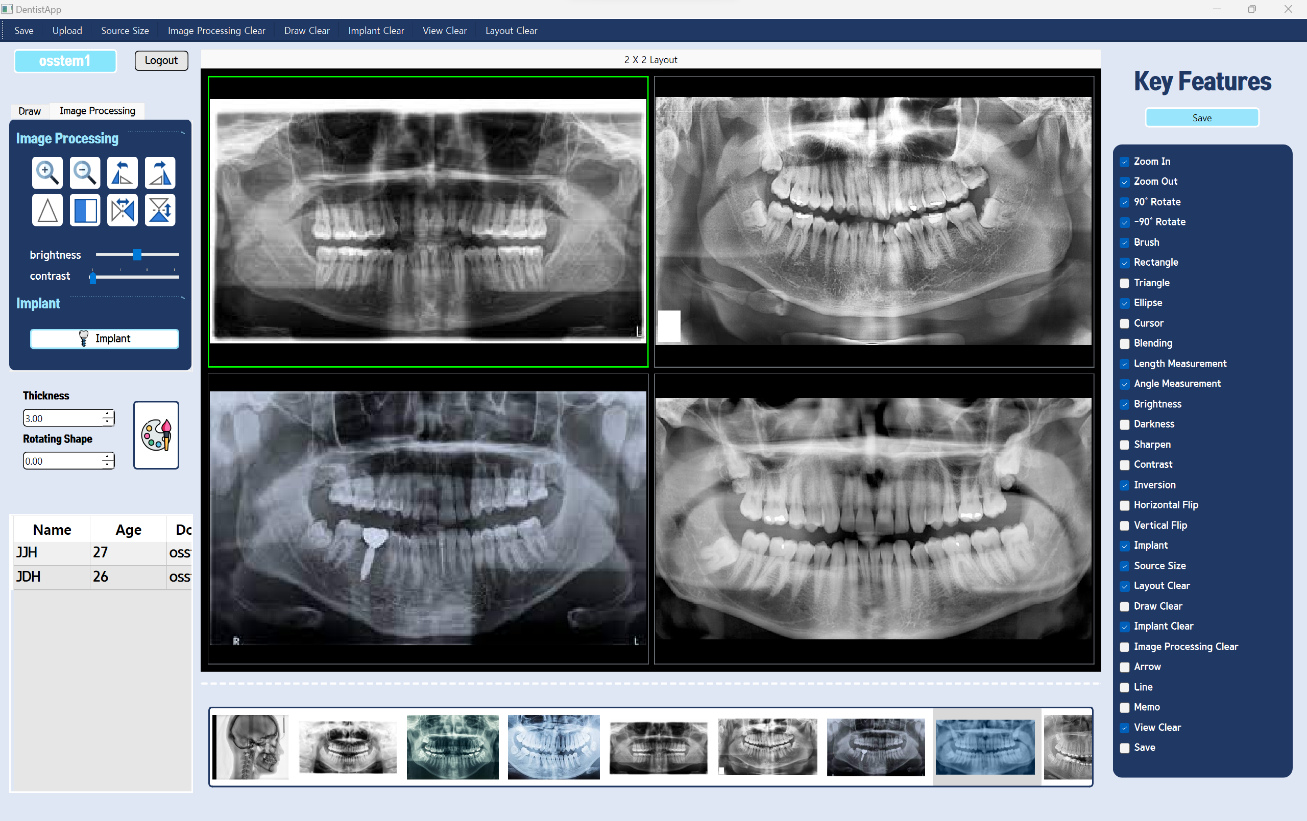
* + 1. 이미지 삭제

|  |  |
| --- | --- |
|  | 삭제할 이미지 선택 후 마우스 더블 우 클릭을 하여 삭제를 시도합니다. |
|  | 삭제 여부 메시지 박스를 출력하여 Yes 버튼을 클릭 시 이미지 정보와 이미지 파일을 삭제하는 URL을 찾아갑니다. |
|  |  |
| 두 URL을 찾으면 이미지 정보와 이미지 파일을 삭제합니다. | |
|  | 삭제 시 listWidget에서 선택한 이미지는 삭제됩니다. |

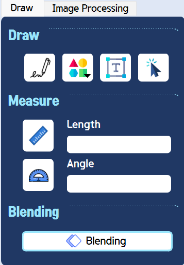
|  |  |
| --- | --- |
| 1. MongoDB Atlas View Monitering   MongoDB Atlas 홈페이지의 View Monitering 페이지는 서버의  Opcounter, Network, Connections 정보를 나타냅니다. | |
|  | - **OpCounter :** 서버에서 주고 받은 정보의 연산 속도를 시간에 따라 출력하는 그래프 입니다 |
|  | - **Network**  : 서버에서 주고받는 정보의 데이터 흐름을 경과 시간에 따라 출력하는 그래프입니다. |
|  | - **Connections**  **:** 서버의 활성화부터 종료까지 클라이언트의 접속 상태 여부를 나타내는 그래프입니다. |
|  | |

1. **2D Viewer**

GUI 설계



Key Features



Draw Page

Thickness /

Rotating Shape / Color

ToolBar 기능 : Save / Upload / Source Size / Image Processing Clear / Draw Clear / Implant Clear / View Clear / Layout Clear

Doctor ID

Logout

Image Processing

Page

선택된 View 테두리 효과

Patient Information DB

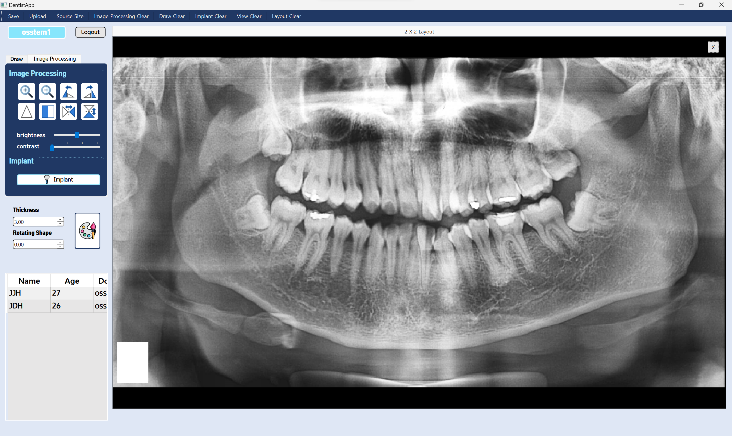
Image Load ( listWidget )

View1

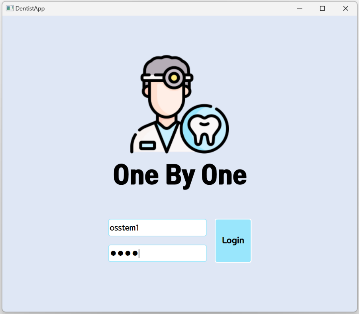
View2

View3

View4



Max New View



Login / Logout

[ 기능 설명 ]

1. **ToolBar** : 저장 및 초기화 항목들을 최상단에 배치하여 편리하게 사용할 수 있습니다.

– Save / Upload / Source Size / Image Processing Clear / Draw Clear /

Implant Clear / View Clear / Layout Clear

1. **Doctor ID** : 로그인 한 의사 정보(ID)를 표시해줍니다.
2. **Login / Logout** : 여러 사용자가 프로그램을 사용할 수 있도록 Login / Logout 기능을

설계하였습니다.

1. **Draw Page** : 화면에 그림을 그려 환자에게 쉽게 설명할 수 있도록 Brush 및 도형, 메모 기능과 이미지에 길이와 각도 측정을 하여 실제 치아의 길이를 알 수 있습니다.

또한 블랜딩 기능을 추가하여 두 이미지를 혼합하여 투명하게 비치는 효과를 볼 수 있습니다. – Bursh, Circle, Rectangle, Triangle, Line, Arrow, Length, Angle, Blending

1. **Image Processing Page** : 이미지를 세밀하게 볼 수 있도록 확대/축소/회전/수평, 수직 반전 기능과 이미지가 흐리거나 어두워서 경계가 뚜렷하지 않을 때 밝기/선명화/대비/명암 기능을 사용할 수 있습니다.

또한 임플란트를 식립하여 치아 및 구강의 자세한 위치나 형태를 알아볼 수 있습니다.

– Zoom in, Zoom out, Rotate, Flip, Sharpen, Inversion, Brightness, Contrast, Implant

1. **Thickness / Rotating Shape / Color** : Brush와 도형, 메모 및 길이, 각도 측정의 굵기와

색상을 변경할 수 있고 임플란트와 포함하여

모든 도형을 회전할 수 있습니다.

1. **Patient Information DB** : 담당 의사에 대한 환자 목록을 보여줍니다.
2. **Image Load** : 환자를 선택하면 해당 환자에 대한 이미지가 나타납니다.
3. **Key Features** : One By One 프로그램에서 사용할 수 있는 모든 기능 30가지를 한 번에

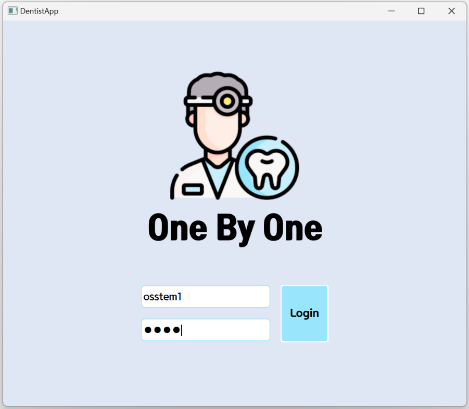
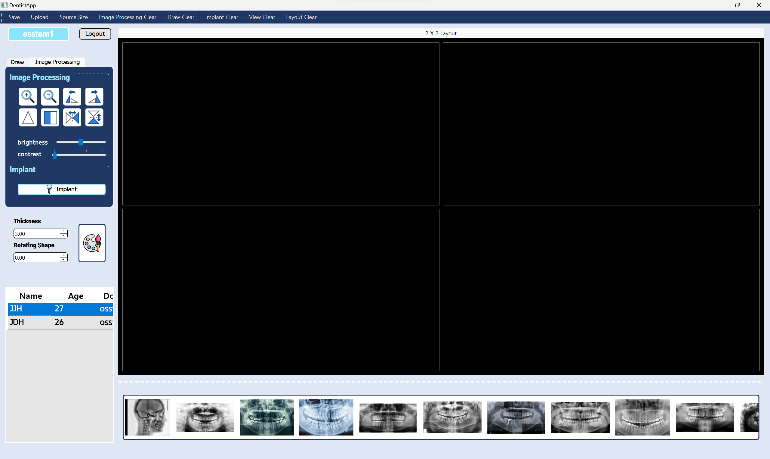
모아 의사가 체크 박스로 선택한 주요 기능들을 마우스만으로 편리하게

사용할 수 있도록 설계하였습니다.

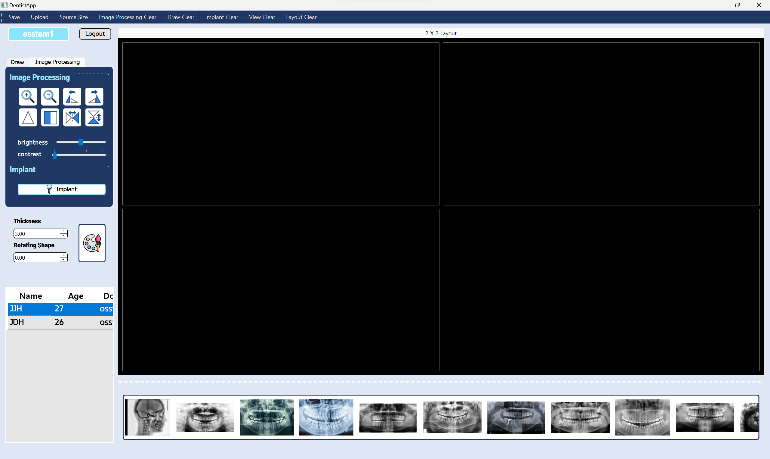
1. **Layout** : 4개의 화면으로 분리하여 여러 이미지를 한 눈에 볼 수 있으며,

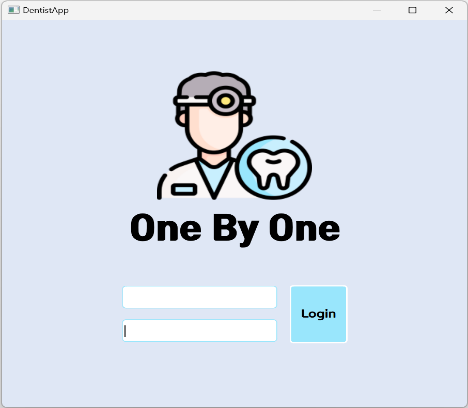
이미지를 크게 보고 싶은 경우 화면을 더블 클릭하면 1개의 큰 화면으로 볼 수 있습니다. – View1, View2, View3, View4, Max New View

A. Login / Logout Page [로그인/로그아웃]

1. Login [로그인]



1. Logout Page[로그아웃]



[ 설계 목적 ]

: 하나의 프로그램을 여러 명의 의사가 사용할 수 있기 때문에 의사 별로 담당하고 있는 환자들의 목록을 구분하기 위해 로그인 기능을 만들었고, 로그아웃 기능을 추가하여 접속된 의사 외 다른 의사로 쉽게 변경할 수 있도록 설계하였습니다.

[ 사용 방법 ]

1. Login 버튼을 눌렀을 때 입력한 ID와 비밀번호가 일치하면 프로그램이 접속됩니다.

- 만약 ID와 비밀번호가 일치하지 않는 경우 “다시 시도해주세요” 팝업창이 나타나며

프로그램에 접속할 수 없습니다.

1. Logout 버튼을 누르면 다시 로그인 페이지로 이동합니다.

B. Layout [레이아웃]

[ 설계 목적 ]

: 여러 각도에서 촬영된 치아 이미지(ex. 세팔로, 파노라마 등)를 불필요한 화면 이동 없이 4개의 화면으로 한 번에 볼 수 있습니다. 또한 1개 이미지만 크게 보고 싶은 경우 화면을 더블 클릭하면 큰 창으로 변경되어 확대된 이미지를 볼 수 있습니다.

[ 사용 방법 ]

|  |  |  |
| --- | --- | --- |
| View4  View2  View1  View3 | - **Layout 위치**  → grid1 : 0행 0열 위치에 있는 View1 화면입니다.  → grid2 : 0행 1열 위치에 있는 View2 화면입니다.  → grid3 : 1행 0열 위치에 있는 View3 화면입니다.  → grid4 : 1행 1열 위치에 있는 View4 화면입니다. | |
|  | - **Layout 동작원리**  : View를 선택하지 않고 ListWidget에 있는 이미지를  선택 하면 View1 View2 View3 View4 순서대로 선택한 이미지가 출력됩니다. | |
|  | : 원하는 위치의 View(현재 View1)를 선택하게 되면  선택된 View(현재 View1)의 테두리 색상이 변경되고 해당 View(현재 View1)에서만 이미지가 출력됩니다.  테두리 영역이 잡혀진 View1에서 listWidget에 있는 다른 이미지를 선택할 경우 View1에서만 이미지가 변경됩니다. | |
| **- Layout 특징 ①**  **1) Layout 세로 비율이**  **가로보다 클 경우**  : 파노라마 이미지의 경우 이미지의 가로 길이가 세로 보다 크기 때문에 view의 가로의 길이가 세로보다 작을 경우 이미지 위, 아래에 여백이 생기게 됩니다.  이때 위젯의 크기를 가로보다 세로가 커지게 조절할 경우 이미지의 원본 크기를 유지하면서 View의 크기에 맞춰 재정렬해줍니다. | | **2)  Layout 가로 비율이 세로보다 클 경우**  : 위젯의 크기를 세로보다 가로가 커지게 조절할 경우  View의 가로의 길이가 더 커지게 되므로 이미지  좌, 우에 여백이 생기게 됩니다.  이때 위젯의 크기를 세로보다 가로가 커지게 조절할 경우 파노라마 원본 이미지 가로, 세로 크기 비율을 유지하면서 View의 크기에 맞춰 재정렬됩니다. |
| X 버튼  maxNewGrid | **- Layout 특징 ②**  : View를 더블 클릭하게 되면 큰 화면(maxNewGrid)으로 위젯이 변경되고 선택한 View에 있는 이미지가 큰 화면에서 출력됩니다.  (큰 화면(maxNewGrid)으로 변경된 경우 이미지를 선택할 수 있는 listWidget이 사라지게 되므로 이미지를 선택할 수 없습니다.)  큰 화면 위젯 역시 원본 이미지의 가로 세로 비율을 유지하면서 maxNewGrid의 크기에 맞춰 최대 이미지 크기로 출력되고 빈 공간은 여백이 생기게 됩니다.  우측 상단에 있는 x버튼을 클릭하게 되면 다시 Layout 위젯으로 변경되면서 View의 크기에 맞춰 이미지가 재정렬됩니다. | |

C. Draw

* 드로우 툴

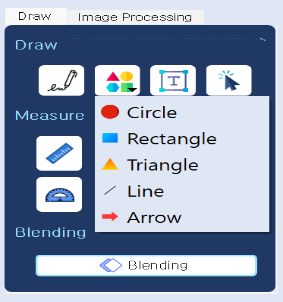
|  |  |  |
| --- | --- | --- |
| **아이콘** | **이름** | **설명** |
|  | 브러쉬 | 브러쉬를 이용한 그리기 |
|  | 도형 | 타원, 사각형, 삼각형, 직선, 화살표와 같은 도형 그리기 |
|  | 메모 | 문자열 출력 |
|  | 커서 | 커서 모드 설정 |

➀ Brush 기능 사용하기

Draw 툴의  아이콘을 클릭하면 Brush 모드가 활성화되며 사진 위에 펜을 이용하여 그리는 기능이 활성화됩니다.

|  |  |
| --- | --- |
| **메뉴** | **설명** |
| **Brush를 통한 그리기** | 원하는 위치에 왼쪽 마우스를 클릭하고 이동하면서 원하는 영역에 그리기 기능을 수행합니다.    **[ Brush로 그림 그리기 전(좌) / 후(우) ]** |

➁ Shape 기능 사용하기



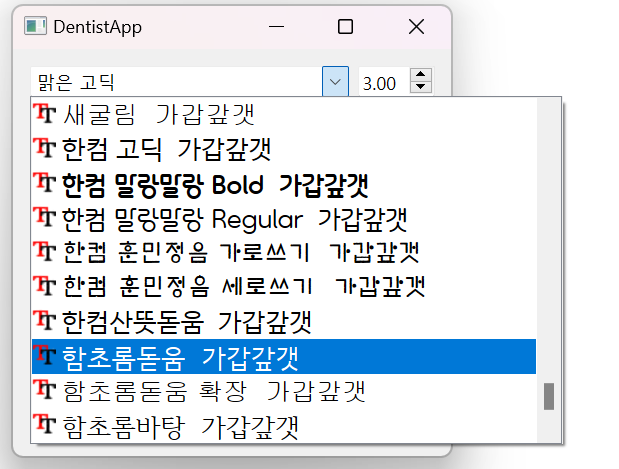
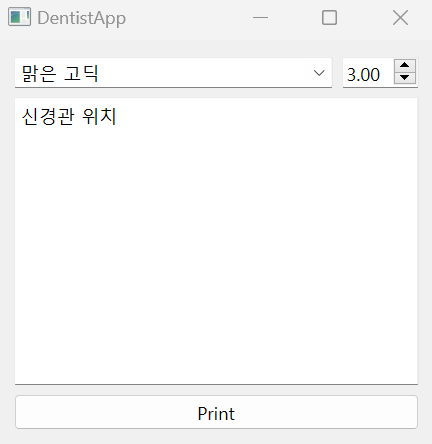
Draw 툴의  아이콘을 클릭하면 포함된 도형들이 나타납니다. 사용하고자 하는 도형을 선택하면 마우스 드래그나 클릭을 통해 해당 도형을 그릴 수 있습니다.

|  |  |
| --- | --- |
| **메뉴** | **설명** |
| **Shape를 통한 타원 그리기** | 마우스를 드래그하여 영역을 잡아 해당 영역만큼의 원이 그려집니다.      **[ Shape로 타원 그리기 전(좌) / 후(우) ]** |
| **Shape를 통한 사각형 그리기** | 시작점과 끝점의 좌표를 클릭하여 해당 영역만큼 사각형 도형이 그려집니다.      **[ Shape로 사각형 그리기 전(좌) / 후(우) ]** |
| **Shape를 통한 삼각형 그리기** | 세 점의 꼭짓점을 좌표로 클릭하여 해당 영역만큼 삼각형 도형이 그려집니다.    **[ Shape로 삼각형 그리기 전(좌) / 후(우) ]** |
| **Shape를 통한 직선 그리기** | 시작점과 끝점의 좌표를 클릭하여 점을 이어 직선 도형이 그려집니다.    **[ Shape로 직선 그리기 전(좌) / 후(우) ]** |
| **Shape를 통한 화살표 그리기** | 시작점과 끝점의 좌표를 클릭하여 해당 영역만큼 화살표 도형이 그려집니다.    **[ Shape로 화살표 그리기 전(좌) / 후(우) ]** |

➂ Memo 기능 사용하기

Draw 툴의 아이콘을 클릭하면 다음과 같은 문자를 입력받을 위젯이 나타납니다.

폰트와 사이즈 설정이 가능하며 문자열을 입력받습니다. Print 버튼을 클릭하고 보여주고자 하는 곳에 좌표를 클릭하면 문자열이 나타납니다.

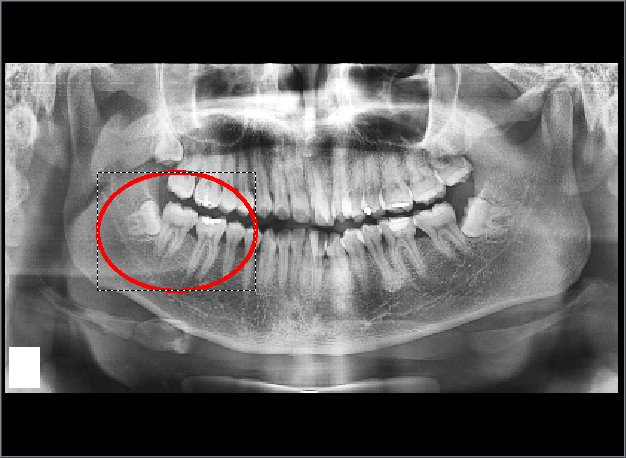


< 폰트와 사이즈 변경 >

|  |  |
| --- | --- |
| **메뉴** | **설명** |
| **Memo를 통해 문자열 출력** | Textedit에 출력하고자 하는 문자열을 입력하고 Print 버튼을 클릭하고 위치를 찍으면 문자열이 출력됩니다.    **[ Memo로 문자열 출력 전(좌) / 후(우) ]** |

➃ Cursor 기능사용하기

Draw 툴의 아이콘을 클릭하면 Cursor 모드가 활성화되고 브러쉬, 도형, 문자열 아이템들이 선택됩니다. 선택된 아이템들은 영역이 설정되고 마우스 드래그를 통해 움직일 수 있습니다.



선택된 아이템 영역 설정

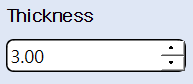
|  |  |
| --- | --- |
| **메뉴** | **설명** |
| **Cursor를 통해 아이템 선택 및 이동** | Cusor 버튼 또는 오른쪽 마우스를 클릭하면 Cursor 모드로 설정됩니다. Cursor 모드에서 아이템을 선택하면 아이템이 Select되고 이동 및 삭제를 할 수 있습니다.      **[ Cursor로 아이템 선택 및 이동 전(좌) / 후(우) ]** |

**※ 오른쪽 마우스를 클릭하면 자동으로 Cursor 모드로 설정됩니다.**

* 기타 설정  
    
  설정 항목에서 브러쉬 및 도형의 굵기 조절, 각도 회전, 색상 변경 할 수 있습니다.

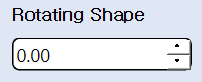
|  |  |
| --- | --- |
| **기능** | **설명** |
|  | 설정된 포인트 값만큼 브러쉬 및 도형 굵기 조절 |
|  | 설정된 각도 값만큼 브러쉬, 문자열, 도형 회전 |
|  | 브러쉬, 문자열, 도형 색상 변경 |

➀ 굵기 조절하기

 에서 두께 값을 설정하여 브러쉬 및 도형의 굵기를 조절합니다. (초기값 : 3.0)

|  |  |
| --- | --- |
| **메뉴** | **설명** |
| **설정된 값만큼 굵기 조절** | Thickness 스핀 박스에서 화살표 또는 키보드로 원하는 값을 설정하여 굵기를 설정할 수 있습니다.    **10.0 point**  **3.0 point**  **[ 굵기 조절 3.0point(좌) / 10.0point(우) ]** |

➁ 도형 회전하기

 에서 설정된 각도값만큼 도형 및 문자를 회전시킵니다.

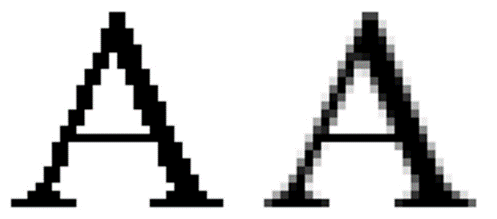
|  |  |
| --- | --- |
| **메뉴** | **설명** |
| **설정된 각도값만큼 아이템 회전** | Rotating Shape 스핀 박스에서 화살표 또는 키보드로 원하는 값을 설정하여 회전 각도를 설정할 수 있습니다. 화살표를 이용하면 5도씩 증가하며 값이 증가할수록 시계방향으로 회전됩니다.      **[ 180도 회전 전(좌) / 후(우) ]** |

➂ 색상 변경하기

아이콘을 클릭하면 다음과 같은 Dialog가 나타납니다. 사용자가 원하는 색상을 저장하고 추가할 아이템의 색상을 변경할 수 있습니다.

|  |  |
| --- | --- |
| **메뉴** | **설명** |
| **색상 변경** | 색상 다이얼로그에서 원하는 색상을 선택하고 난 후 브러쉬, 도형, 메모 기능의 아이템들이 해당 색상으로 변경되어 출력됩니다.        **[ 색상 선택 (좌) / 색상 적용 (우) ]** |

**※ 안티에일리어싱**



**[ 안티-에일리어싱(anti-aliasing) 전(좌) / 후(우) ]**

제한된 해상도 환경에서 곡선에 계단 모양의 외곽이 생기는데 이것을 ‘에일리어싱(aliasing)’이라고 합니다. 이 현상을 완화하고자 경계를 모호하게 하는 기법을 **‘안티-에일리어싱(anti-aliasing)’**이라 합니다.

|  |  |
| --- | --- |
| **메뉴** | **설명** |
| **안티에일리어싱 적용** | 브러쉬 및 도형에 ‘안티-에일리어싱(anti-aliasing)’이 자동으로 적용되어 계단 현상이 거의 없는 것을 확인할 수 있습니다.      **확대**  **[ 안티에일리어싱 적용 전(좌) / 후(우) ]** |

|  |  |
| --- | --- |
| * 길이 및 각도 측정 | |
|  | 서버와 연동된 환자 정보 중 1명의 환자 이름을 클릭하여 listWidget에 이미지를 출력합니다 |
|  | 선택된 이미지를 View에 출력하고 좌측 탭 위젯의 Draw 파트에 Length와 Angle 위젯을 띄워 측정할 준비를 합니다. |
|  | listWidget의 이미지를 클릭 시  측정 간 필요한 정보 중  Pixel Length의 정보를 받아오기 위해 이미지 정보 URL에 접속합니다.  해당 이미지 파일과 동일한 정보를 가진 이미지 정보 내의 픽셀 데이터를 추출하고, 클릭한 이미지의 width, height 정보를 추출합니다.  이미지 정보를 모두 탐색하면 탐색을 종료합니다. |
|  |  |
| 각도 측정 시 blue 색상의 Poly Line 아이템을 사용하여 세 점의 좌표를 클릭 시 두 변의 사잇각을 출력합니다.  1번째 변과 2번째 변을 B, C로 정하고 그려지지 않은 변을 A로 정합니다. 그러면 구해야 할 각도는 A 변과 대칭되는 각도의 공식 다음과 같습니다.  길이 측정 시 green 색상의 Line Item으로 두 좌표 사이의 거리를 표시합니다.  A =  라는 길이 측정 공식을 사용한 뒤 A x Pixel Length로 계산하면 실제 길이를 측정할 수 있습니다. | |
|  | |

1. Image Processing
2. **Image Processing [이미지 효과]**

이미지에 효과를 주어 환자에게 진단 정보를 쉽게 설명할 수 있도록 도움을 줍니다.

예를 들어, 이미지를 확대(Zoom In)하여 특정 부위를 자세하게 볼 수 있고, 확대된 이미지를 축소(Zoom out)할 수 있습니다. 또한 확대된 이미지를 마우스 오른쪽 버튼을 누른 채 드래그 하면 이미지를 이동(Panning)할 수 있으며, 크기가 변경된 이미지를 원본 크기(Source Size)로 되돌리거나 회전(Rotate) 및 반전(Flip)을 이용하여 이미지 효과를 적용할 수 있습니다.

1. **ImageProcessing 기능 설명**

**선명화**

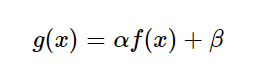
* 선명화 작업에 사용된 공식으로는 [원본 이미지 + ( 원본이미지 – 블러처리된 이미지 ) \* 가중치] 를 이용하였습니다.
  + OpenCV의 Matrix함수를 이용하여 5 \* 5필터의 가우시안 블러링를 구현하고, 공식을 이용하여 선명화를 구현하였습니다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 원본 | 블러 처리 | 원본 – 블러 영상 | 샤프닝 |
| 이미지 픽셀 값 |  |  |  |  |
| 공식 | [원본 이미지 + ( 원본 이미지 – 블러 영상 ) \* 가중치] | | | |

**대비** : QImage 함수내에 invertPixels을 이용하여 RGB 픽셀 값을 (255 – i (현재 RGB픽셀의 값)) 식으로 계산된 값으로 변환합니다.

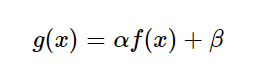
**명암**

* 이미지의 픽셀 값( 0 ~ 255 )을 읽어 OpenCV 함수인 Mat 함수를 이용하여 매트릭스 연산으로 명암을 구현합니다.
* 알파값을 곱하여 픽셀의 어두운 부분은 더 어둡게 , 밝은 부분은 더 밝게 픽셀 값을 변경됩니다.
* 원본 이미지에 베타값을 더하여 픽셀 값을 변경 됩니다.



**밝기** :

* 이미지의 픽셀 값( 0 ~ 255 )을 읽어 OpenCV 함수인 Mat 함수를 이용하여 매트릭스 연산으로 밝기를 구현합니다.



**수직 반전** : View의 X축과 Y축의 값을 변경하여 view의 좌표를 변경하여 수직 반전 합니다.

**수평 반전** : View의 X축과 Y축의 값을 변경하여 view의 좌표를 변경하여 수평 반전 합니다.

[ 기능 적용 결과 ]

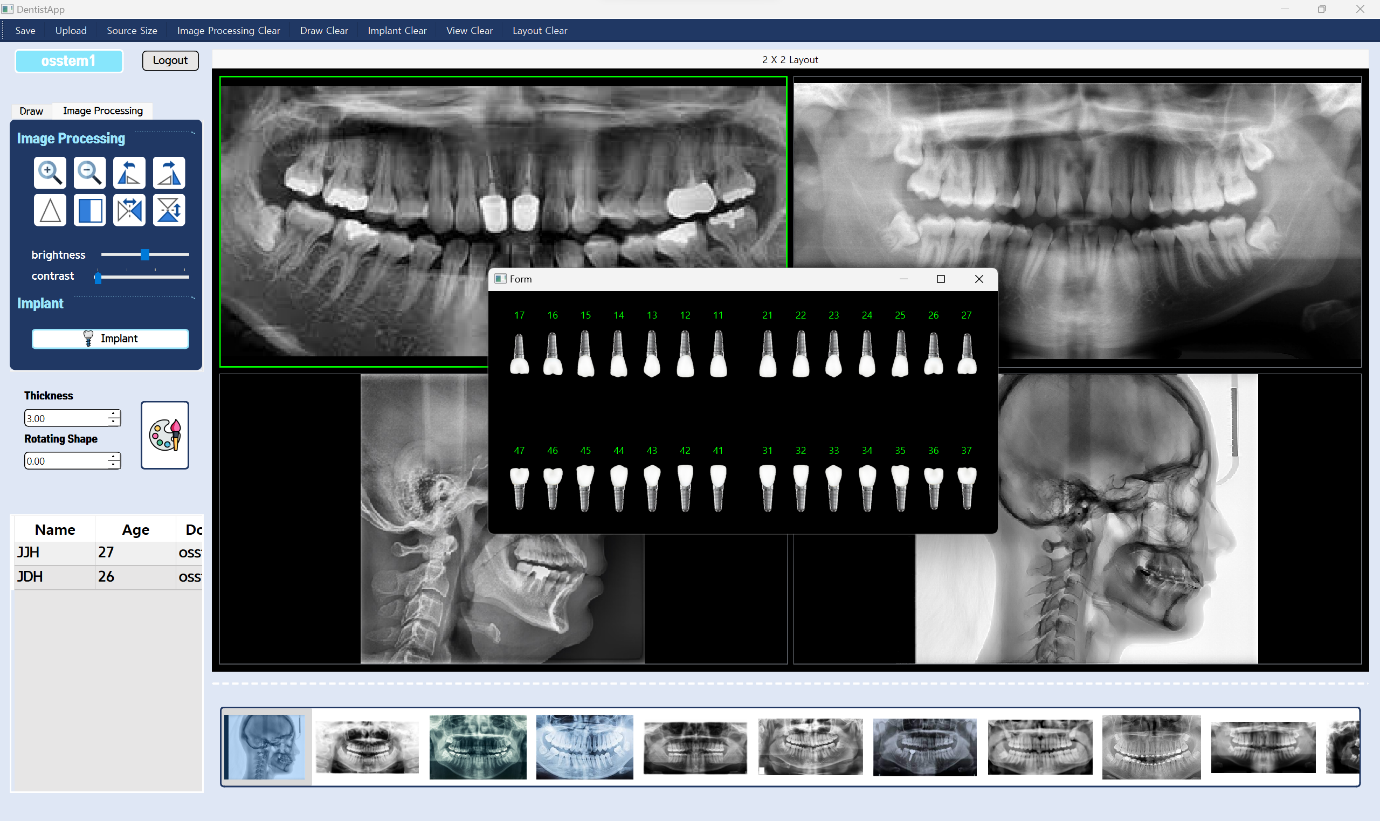
|  |  |  |
| --- | --- | --- |
| 기능 | 효과 전 ( 적용 X ) | 효과 후 ( 적용 O ) |
| Zoom In | - View3에 확대 효과를 적용    - maxNewGrid에 확대 효과를 적용 | |
| Panning | - View3에 확대한 후 이미지를 자유롭게 이동한 모습 (위치 이동)    - maxNewGrid에 확대한 후 이미지를 자유롭게 이동한 모습 (위치 이동) | |
| Zoom Out | - View3에 축소 효과를 적용    - maxNewGrid에 축소 효과를 적용 | |
| -90° Rotate | - View3에 -90° 회전 효과를 적용    - maxNewGrid에 -90° 회전 효과를 적용 | |
| +90° Rotate | - View3에 -90° 회전 효과를 적용    - maxNewGrid에 -90° 회전 효과를 적용 | |
| Brightness | - View3에 밝기 값 적용    - maxNewGrid 에 밝기 값 적용    - View3에 어둡기 값 적용    - maxNewGrid 에 어둡기 값 적용 | |
| Contrast | - View3에 명암 값 슬라이더 적용      - maxNewGrid 에 명암 값 슬라이더 적용 | |
| Sharpen | - View3에 선명화 단계별 적용      - maxNewGrid 에 선명화 단계별 적용 | |
| Inversion | - View3에 대비 적용    - maxNewGrid 에 대비 적용 | |
| Vertical  Flip | - View3에 수직 반전 적용    - maxNewGrid 에 수직 반전 적용 | |
| Horizontal Flip | - View3에 수평 반전 적용    - maxNewGrid 에 수직 반전 적용 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. 알파 블랜딩 구현 | | | | |
|  | | | 서버에 연동된 환자 이름을 클릭하여 listWidget의 이미지를 출력합니다. | |
|  | | |  | |
| 좌측 탭 위젯의 Blending 버튼을 클릭하여 질문 메시지 박스에 Yes 버튼을 누릅니다.  만일 다이얼로그 호출을 취소할 시 No 버튼을 클릭하여 다이얼로그 호출을 중단합니다. | | | | |
|  | |  | | |
| 두 개의 이미지가 필요하기 때문에 전 이미지와 후 이미지를 클릭하여 좌측 상단에는 전 이미지 파일 이름을 출력하고, 우측 상단에는 후 이미지 파일 이름을 출력합니다.  만일 두 이미지의 파일 이름이 존재하지 않을 경우에는 슬라이더를 변경해도 이미지가 나오지 않게 설정되어 있습니다. | | | | |
| - A. alpha = 0인 경우 | - B. alpha = 5인 경우 | | | - C. alpha = 10 인 경우 |
| 슬라이더를 왼쪽으로 가깝게 조정할수록 이전 이미지의 영상에 가중치를 둡니다.  오른쪽으로 슬라이더를 이동할 때마다 전 이미지의 투명도가 증가하고, 후 이미지의 투명도 감소하여 슬라이더가 가운데에 있을 경우 50 : 50 블랜드 된 결과 이미지를 출력합니다.  슬라이더를 오른쪽으로 완전히 옮기면 전 이미지와 후 이미지의 투명도 비율은 100 : 0 이 되어 후 이미지만 출력합니다.  블렌딩 된 이미지는 Image 폴더에 저장되어 다른 환자를 선택하거나, 프로그램을 종료하면 블렌딩 결과이미지는 사라집니다. | | | | |

1. Implant [임플란트 식립]

[ 설계 목적 ]

: 치아 상실, 심한 충치, 틀니 부작용 등으로 인해 임플란트 식립이 필요한 경우 파노라마 이미지 치아 위치에 임플란트를 직접 식립하여 환자에게 상세 설명 및 이해를 도울 수 있습니다.

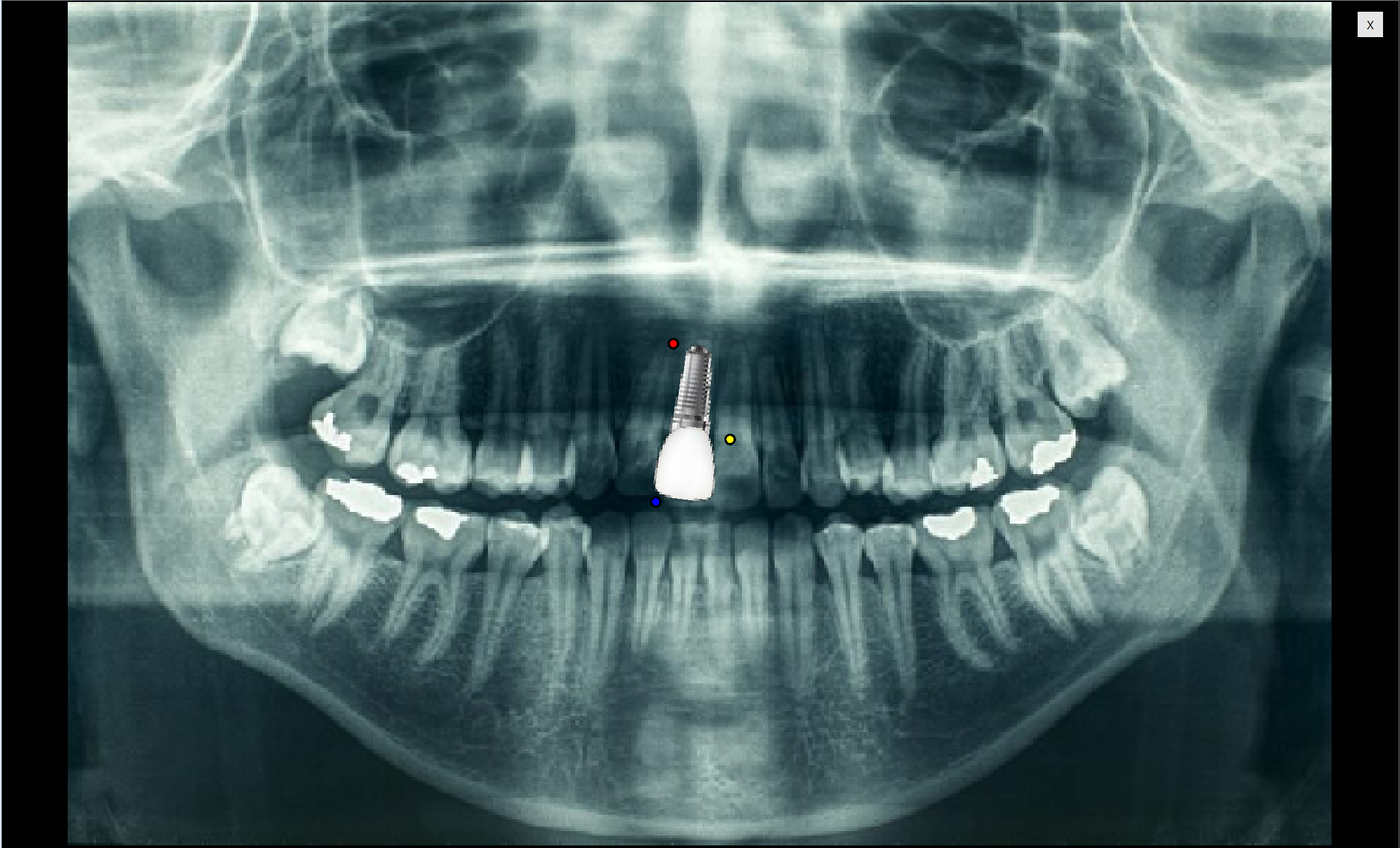


임플란트

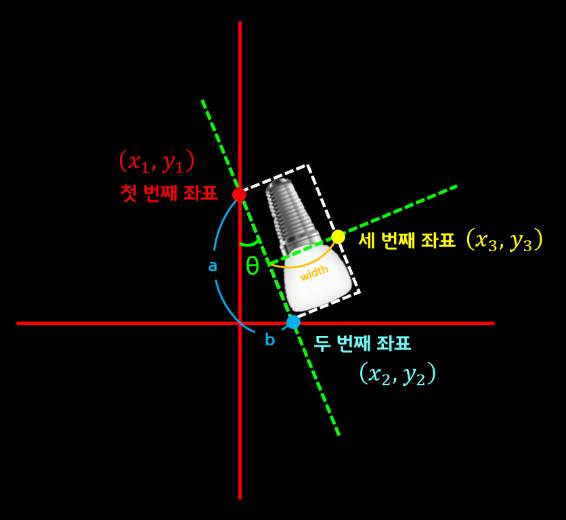
임플란트 식립 기능 버튼을 클릭하면 28개의 치아 이미지를 보여주는 위젯이 출력됩니다.

앞니부터 어금니까지 28개 치아를 선택하여 화면에 임플란트를 식립할 수 있습니다.

* 임플란트 식립 동작원리



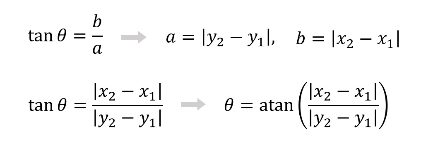
28개의 치아 중 1개의 치아를 선택한 후 화면에 좌표 점 3개를 찍으면 선택한 치아 모양의 임플란트가 식립됩니다.

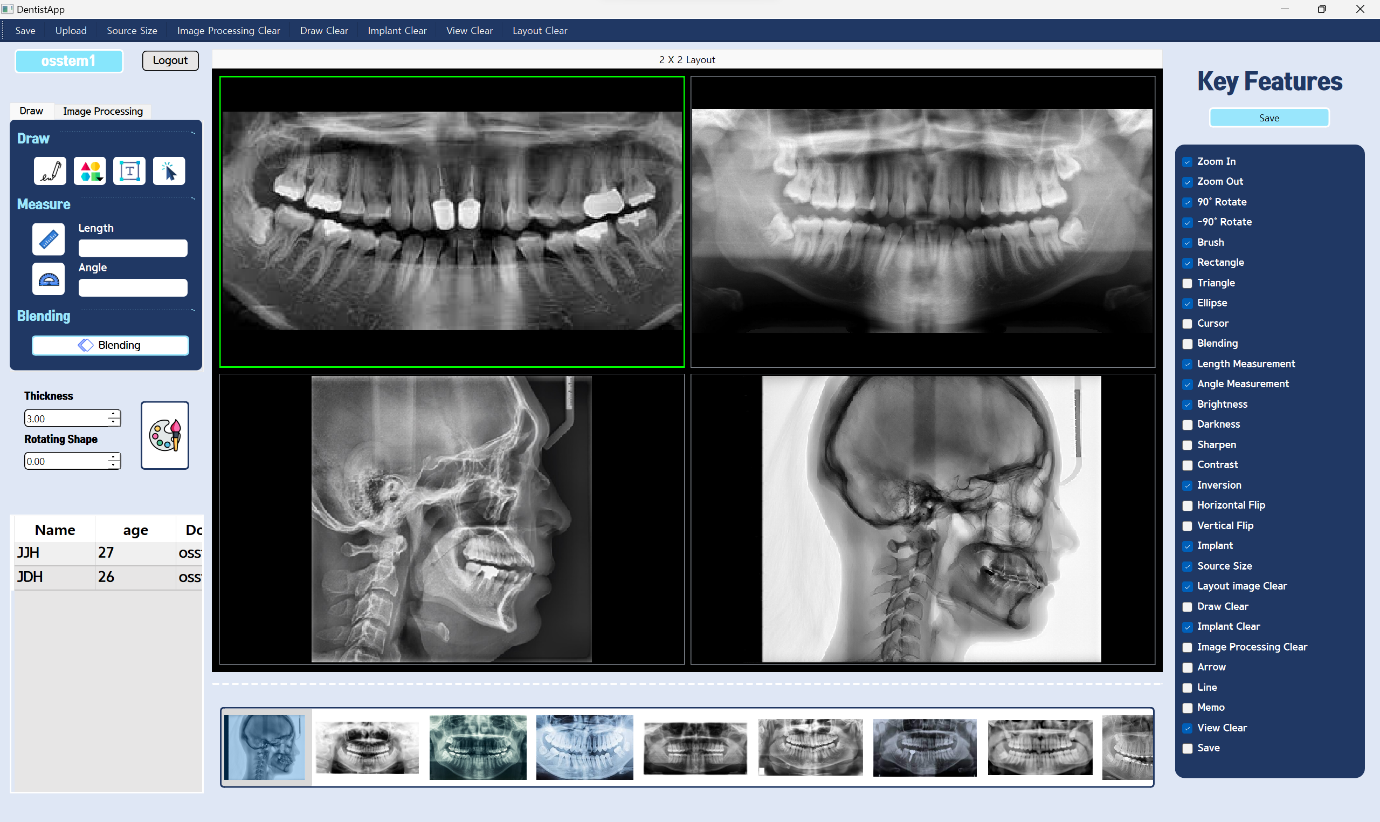


[ 구현 방법 ]

1. 임플란트를 식립할 첫 번째 좌표를 찍습니다. ( 첫 번째 좌표 = 이미지의 TopLeft )
2. 삽입하고 싶은 임플란트 크기(height)만큼 두 번째 좌표를 찍어줍니다. ( 두 번째 좌표 = 이미지의 BottomLeft )
3. 삽입하고 싶은 임플란트 가로폭(width)만큼 세 번째 좌표를 찍어줍니다.

첫 번째 좌표(빨간점)와 두 번째 좌표(파란점)를 이은 선분()의 크기만큼 임플란트의 높이를 설정해주고, 의 중간 좌표()와 세 번째 좌표(노란점)가 이어진 선분(width)만큼 임플란트의 넓이를 설정해줍니다.

1. 첫 번째 좌표와 두 번째 좌표를 찍었을 때 생기는 의 기울기를 통해 를 이용하여 θ 를 구하여 각도만큼 임플란트를 회전시켜 줍니다.
2. Key Features [주요 기능]



저희 One By One 프로그램에서 사용할 수 있는 모든 기능들을

Key Features 체크 박스로 선택할 수 있게 만들어 의사가 주로 사용하는 기능을

편리하게 사용할 수 있도록 설계하였습니다.

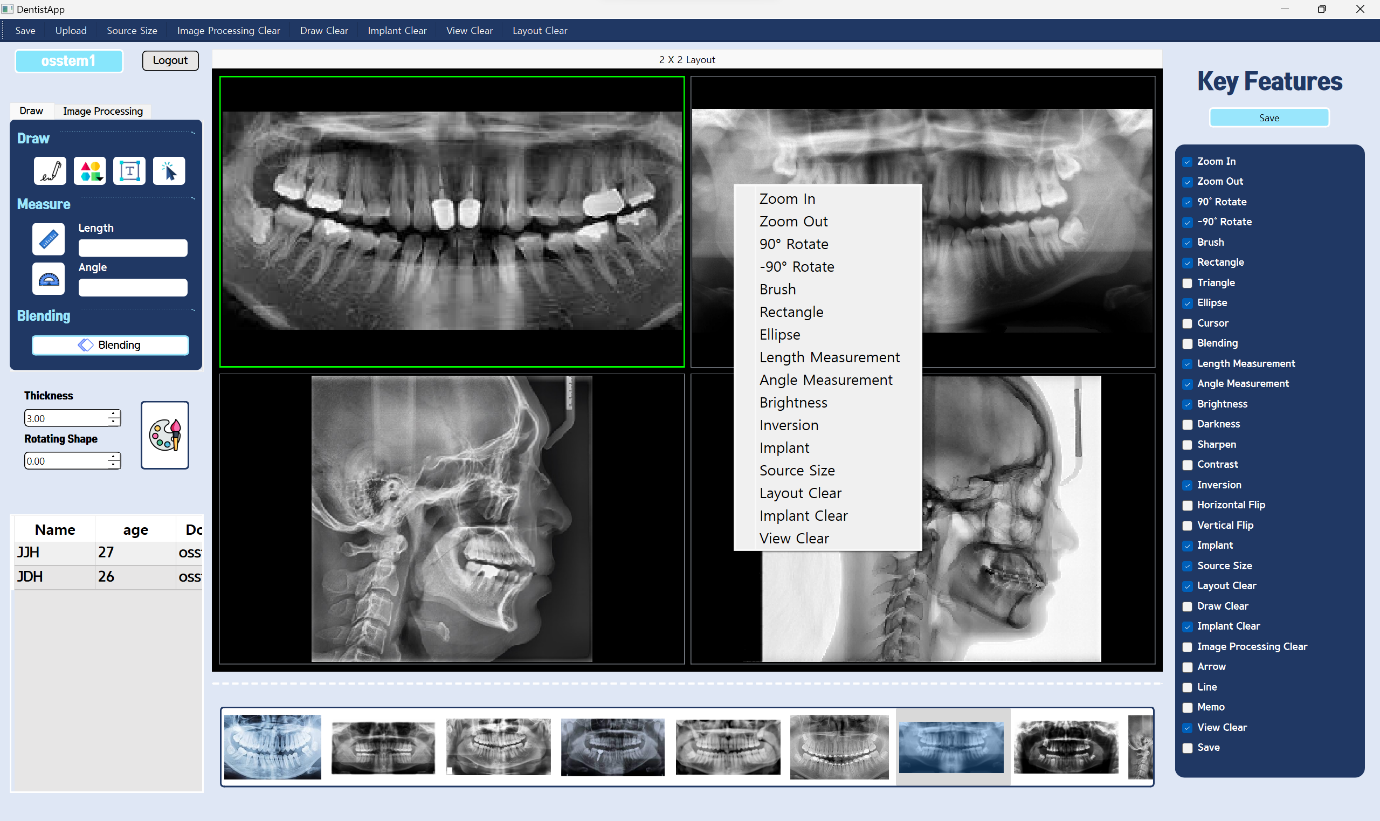
[ 설계 목적 ]

: 어떤 화면이라도 마우스 가운데 버튼을 1번만 클릭하게 되면 자신이 선택한 기능들만 메뉴에 나타나기 때문에 Draw Page, Image Processing Page에서 일일이 기능을 찾을 필요 없이 바로 사용할 수 있도록 편의성을 고려하였습니다.

[ 설계 방법 ]

: 처음 의사 ID로 로그인하게 되면 데이터베이스에 의사 ID 정보가 삽입되고 기능 값들이 0으로 초기화됩니다. 그 후 사용하고 자 하는 기능을 선택 후 Save 버튼을 누르면 DB에 선택된 기능의 값이 1로 변경되고 메뉴에 출력되게 됩니다. 로그아웃 및 프로그램을 종료하게 되면 체크 박스의 기록이 초기화되고 로그인할 때 입력한 의사 ID와 DB의 의사 ID가 일치하는 행을 찾아 기능 값을 읽어 체크 박스를 상태를 업데이트하여 메뉴에 그대로 나타납니다.

* Key Features 동작원리



Doctor ID

Key Features

먼저 로그인을 하게 되면 해당 의사의 DoctorID가 표시됩니다.

의사마다 선호하는 기능이 다르기 때문에 DoctorID 별로 기능들을 다르게 선택할 수 있습니다.

[ 사용 방법 ]

1. 예를 들어 ‘osstem1’ DoctorID로 로그인 합니다.
2. 30가지 기능이 있는 Key Features 체크 박스에서 Zoom In/Out, -90°/+90° Rotate,

Brush, Rectangle, Ellipse, Length Measurement, Angle Measurement, Brightness, Inversion, Implant, Sourve Size, Layout/Implant/View Clear 기능을 선택하고 Save 버튼을 눌러줍니다.

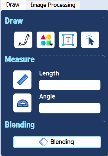
1. 기능을 적용하고 싶은 View를 선택해줍니다. (View1을 선택하겠습니다.)
2. View를 선택한 후 마우스 가운데 버튼을 클릭하게 되면 Key Features에서 선택한 기능들만

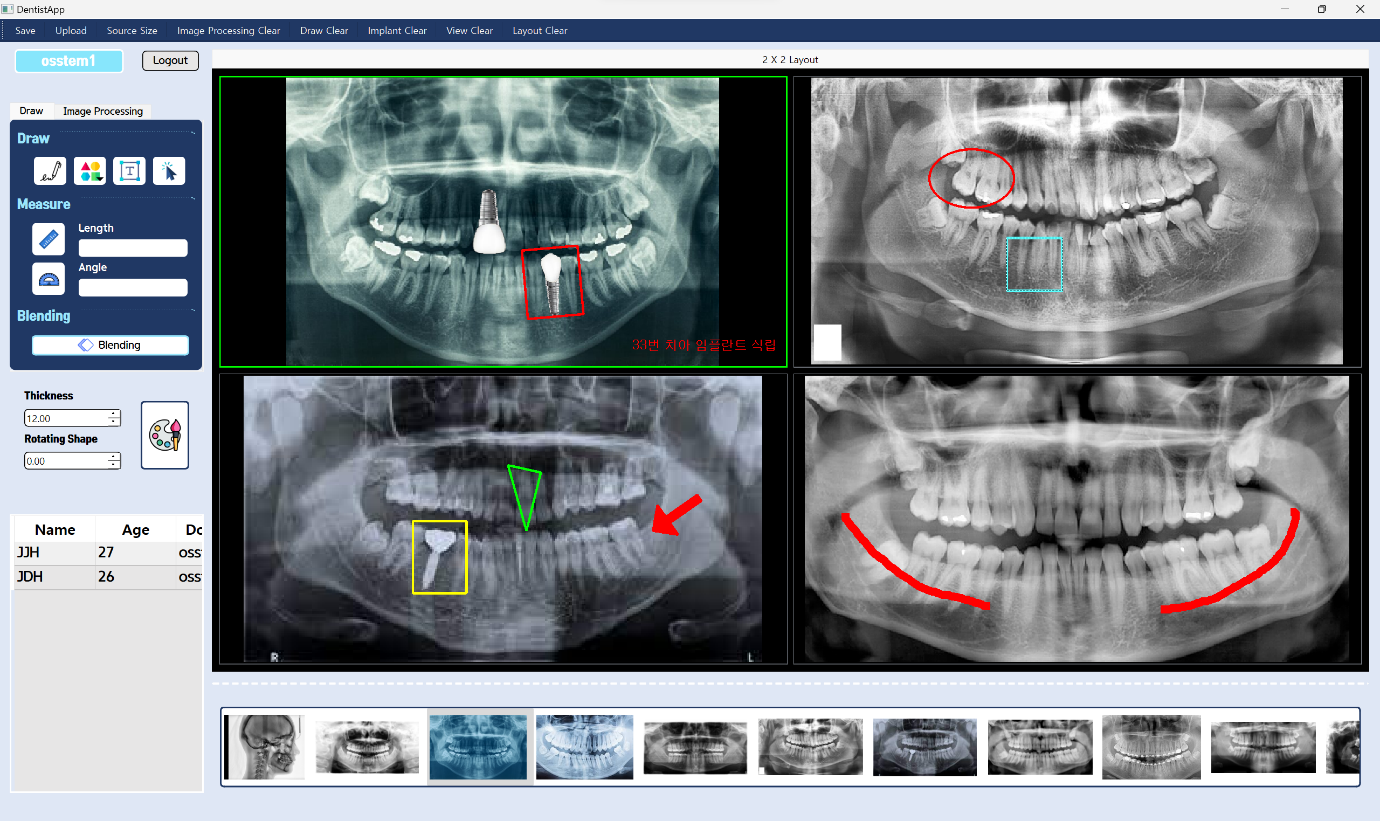
메뉴에 출력되게 됩니다.

1. 나타난 메뉴에서 적용하고 싶은 기능을 선택하게 되면 활성화 된 View에서 기능이 적용됩니다.

|  |  |  |
| --- | --- | --- |
|  | **- Key Features 특징**  **1) 처음 로그인 할 경우**  : Key Feautures에 체크된 기록이 없기 때문에 마우스 가운데 버튼을 클릭해도 메뉴가 출력되지 않습니다. | |
|  | **2) Key Features에서 저장한 기록이 있는 경우**  : Save 버튼을 누를 때 체크박스에 선택되어 있는 항목만 DB에 값이 변경됩니다.  기존에 1번이라도 저장한 기록이 있다면 가장 최근에 저장된 체크 항목이 메뉴에 출력됩니다.  이때 로그아웃 및 프로그램 종료 후 다시 로그인을 하여도 DB에 기록이 남아있기 때문에 마우스 가운데 버튼을 누르면 가장 최근에 저장된 메뉴가 출력됩니다. | |
| Doctor ID = osstem1     1. ‘ osstem1 ’로 로그인 한 경우 | | Doctor ID = osstem2     1. ‘ osstem2 ’로 로그인 한 경우 |
| DoctorID가 ‘osstem1’ 인 경우와 ‘osstem2’일 때 의사마다 Key Features 에서 선택한 기능들이 다르기 때문에 의사가 각각 선택한 기능들만 메뉴에 출력됩니다. | | |
|  | : 큰 화면(maxNewGrid)일 때도 의사가 선택한 모든 기능들을 동일하게 적용할 수 있습니다. | |

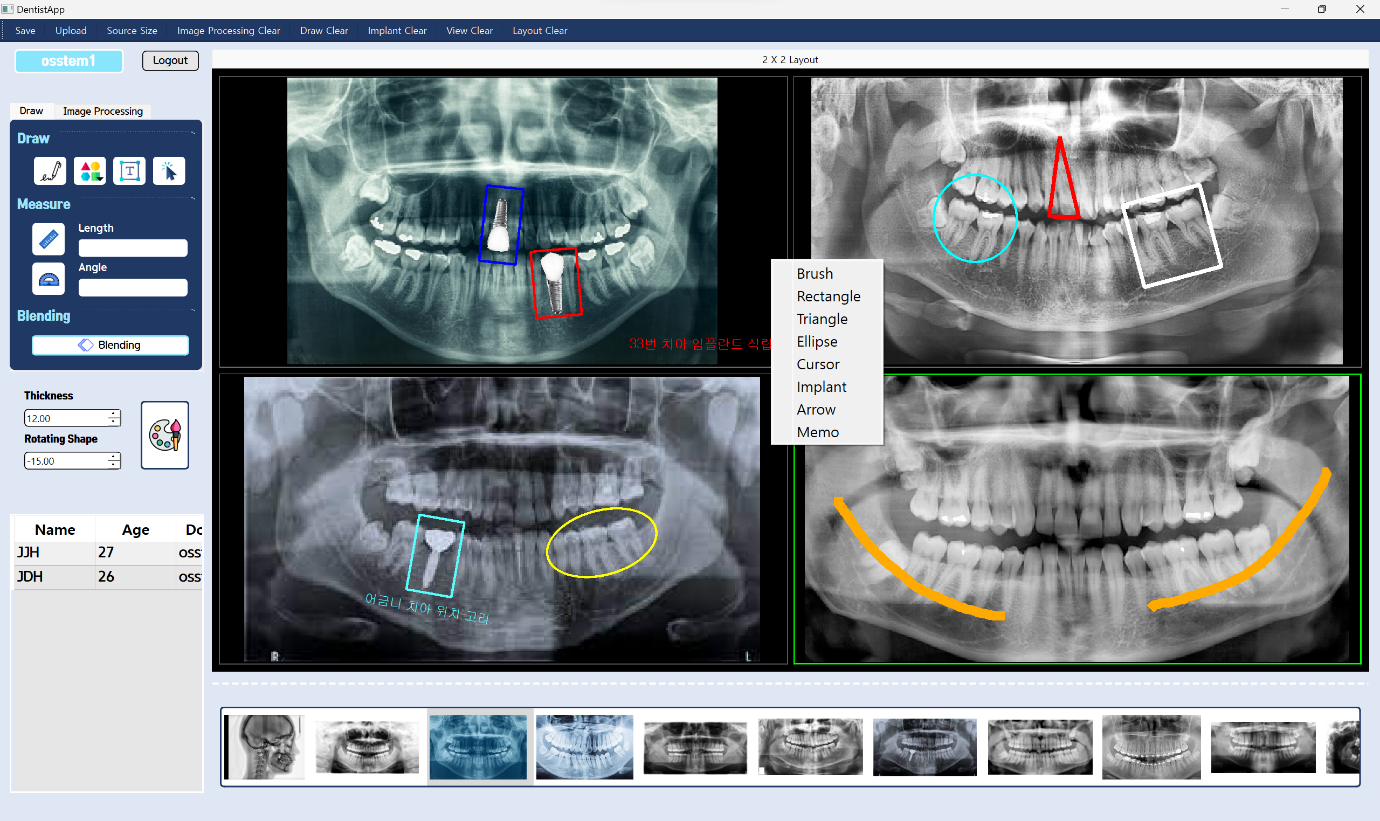
**1. 버튼을 사용하여 이미지에 효과를 적용한 경우**





****

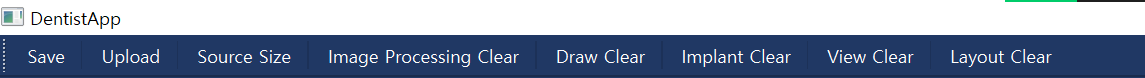
**2. Key Features 기능을 사용하여 이미지에 효과를 적용한 경우**



버튼을 사용하여 효과를 적용한 모습과 Key Features 기능을 사용하여 화면에 효과를 적용한 모습이

동일하다는 결과를 확인할 수 있습니다.

F. ToolBar



**[기능 설명]**

**Save** : 프로그램 상단 툴바의 Save를 통해 원하는 view를 .png 파일로 저장하는 기능을 사용할 수 있습니다.

**Upload :** 화면 상단의 Upload 버튼을 누르면 서버에 이미지를 업로드할 수 있는 다이얼로그를  
호출합니다. (3.1.1.B. 참고)

**Image Process Clear** : 화면 상단에 있는 Image Processing Clear 버튼을 눌리면 다른 기능들은 남아 있는 상태에서 이미지 처리된 화면만 원본 상태로 돌립니다.  
  
**Draw Clear** : 화면 상단에 있는 Draw Clear 버튼을 누르면 이미지는 그대로 남아있는 상태에서 브러쉬, 도형, 메모 아이템들만 초기화됩니다.

|  |  |
| --- | --- |
| 선택 된 view에  이미지가 있을경우  ( All Save ) | -view1에 있는 이미지 전체를 저장 합니다. |
| 선택 된 view에  이미지가 있을경우  ( process Save ) | -view1에 Process 처리 된 이미지만 저장 합니다. |
| maxNewGrid  ( All Save ) | -원하는 항목을 선택할 수 있습니다. |
| maxNewGrid  ( Process Save ) | -Process 처리 된 이미지만 저장 합니다. |
| 선택 된 view에  이미지가 없을 경우 | -view4이미지가 없는 상태 저장을 하게 되면 경고 메시지가 발생합니다. |

|  |  |
| --- | --- |
| Source Size | - View3의 확대된 이미지에서 원본 크기(Source Size) 기능을 적용한 모습    - maxNewGrid 의 확대된 이미지에서 원본 크기(Source Size) 기능을 적용한 모습 |
| View Clear | - View3에 View Clear 효과를 적용    - maxNewGrid 에 View Clear 효과를 적용 |
| Layout Clear | - View3에 View Clear 효과를 적용    - maxNewGrid 에 View Clear 효과를 적용 |
| Implant Clear | - View1에 5개의 임플란트를 식립 후 Implant Clear 효과를 적용    - maxNewGrid 에 5개의 임플란트를 식립 후 Implant Clear 효과를 적용 |
| Image Process Clear | - View1에 임플란트를 식립,Draw,Processing기능 추가 후 Processing Clear 효과를 적용    - maxNewGrid 에 5개의 임플란트를 식립 후 Implant Clear 효과를 적용 |

**Ⅲ. 기대 효과 및 활용 분야**

1. 기대효과
   1. 편리성 향상 : 쉽고 직관적인 UI 구성으로 고객들이 쉽게 사용할 수 있고 사용자 맞춤형 2D 뷰어를 제공하여 편리성이 향상
   2. 효율적인 업무 : 사용자마다 뼈의 골격, 치아 모양을 수기로 작성하던 이전과는 달리 환자 데이터를 불러와 Painter기능을 사용함으로써 시간을 단축하여 효율적인 업무가 가능
   3. 신속하고 정확한 고객 관리 : 고객 정보를 수집하고 고객 데이터가 날짜 별로 저장되어 있기 때문에 신속하게 확인이 가능하고 고객의 상태를 모니터링
2. 발전 방향
   1. 이미지 업로드와 다운로드, 삭제기능에서 수정기능 추가할 수 있습니다
   2. View1~5에 있는 이미지 정보를 저장할 수 있습니다
   3. 이미지 프로세싱을 추가하여 더 깔끔한 이미지를 출력할 수 있습니다.
   4. 마우스에 포커스를 맞춰 구현하였지만 키패드를 이용하여 구현할 수도 있습니다.
   5. Painter 기능의 사이즈를 마우스를 이용하여 조절 함으로서 편리성을 높일수 있습니다.
   6. 필요에 따랄 사용자 지정 레이아웃을 구현하여 설정할 수 있습니다.