

클라이언트 기능 명세서

좋은징조 - 김도엽

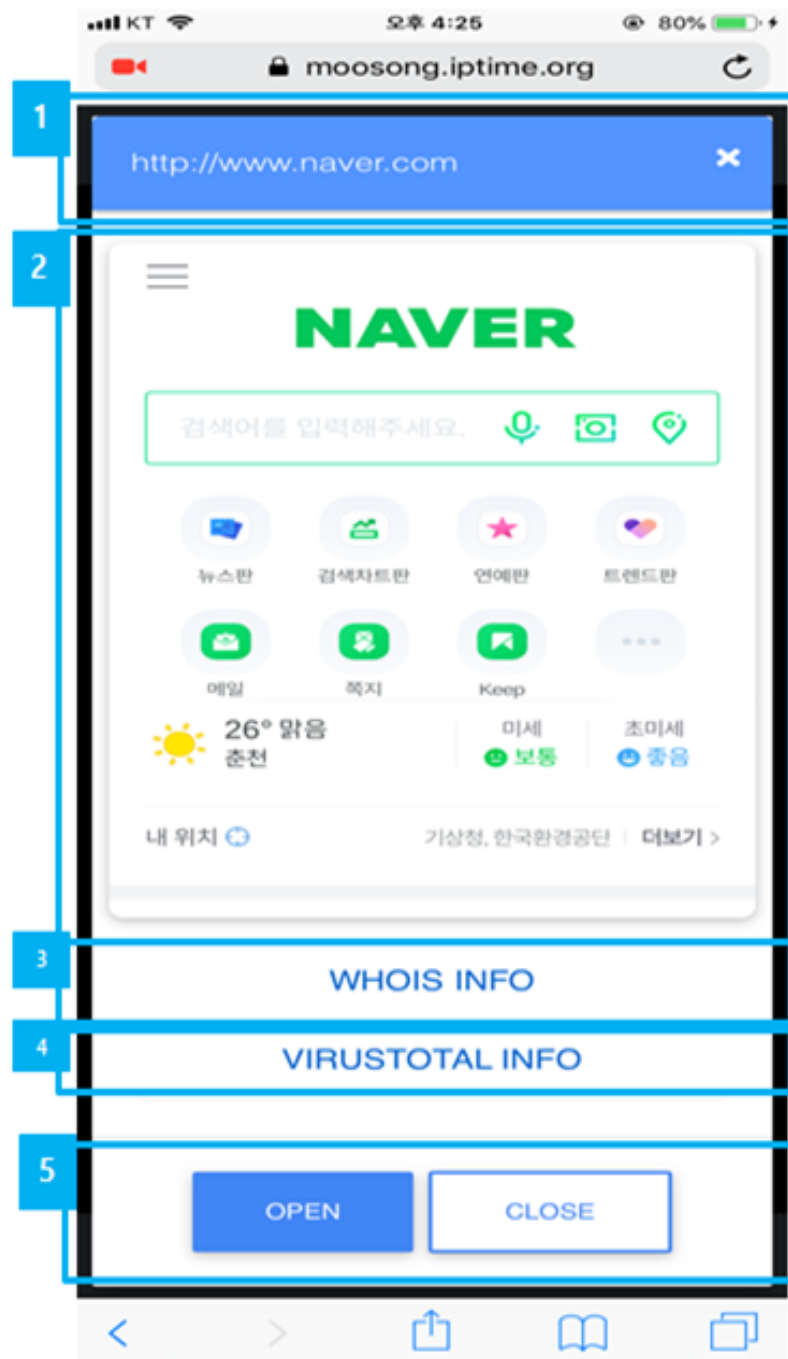
1. 화면 디자인 개요

1) 메인화면



번호	이름	내용
1	QR-Shield	앱의 제목
2	video	카메라 화면 촬영
3	footer	프로젝트 개발팀 이름

2) 모달(팝업)창 화면



번호	이름	내용
1	modalTitle	모달의 제목
2	sitelImage	요청 사이트의 스크린샷 이미지
3	whoisInfo	whois사이트를 이용해 얻은 정보 나열
4	virusInfo	virustotal 사이트를 이용해 얻은 정보 나열
5	footer	open 클릭시 요청 사이트로 이동 close 클릭시 종료

(1) 모달(팝업)창 화면 - WHOIS INFO



번호	이름	내용
1	wDomainNames	도메인 정보
2	wRegistrar	등록기관 정보
3	wOrg	회사명 정보
4	wCountry	국가정보
5	wCity	도시 정보
6	wAddress	주소 정보
7	wEmails	이메일 정보

(2) 모달(팝업)창 화면 - VIRUSTOTAL INFO



번호	이름	내용
1	vScanDate	검사날짜 정보
2	vPositivesAndTotal	발견, 검사한 엔진의 수
3	vScanResults	검사결과 정보

2. 소스정보

이 프로그램을 동작하기 위해서는 다음과 같은 자바 스크립트와 스타일 시트가 필요하다.

1) Javascript File

```
<script type="text/javascript" src="static/js/main.js"></script>
```

- QR-Shield의 메인 자바스크립트 파일

```
<script type="text/javascript" src="static/js/grid.js"></script>
<script type="text/javascript" src="static/js/version.js"></script>
<script type="text/javascript" src="static/js/detector.js"></script>
<script type="text/javascript" src="static/js/formatinf.js"></script>
<script type="text/javascript" src="static/js/errorlevel.js"></script>
<script type="text/javascript" src="static/js/bitmat.js"></script>
<script type="text/javascript" src="static/js/datablock.js"></script>
<script type="text/javascript" src="static/js/bmparser.js"></script>
<script type="text/javascript" src="static/js/datamask.js"></script>
<script type="text/javascript" src="static/js/rsdecoder.js"></script>
<script type="text/javascript" src="static/js/gf256poly.js"></script>
<script type="text/javascript" src="static/js/gf256.js"></script>
<script type="text/javascript" src="static/js/decoder.js"></script>
<script type="text/javascript" src="static/js/qrcode.js"></script>
<script type="text/javascript" src="static/js/findpat.js"></script>
<script type="text/javascript" src="static/js/alignpat.js"></script>
<script type="text/javascript" src="static/js/databr.js"></script>
```

- webqr 모듈을 사용하기 위한 자바스크립트 오픈 소스 (출처 : <https://github.com/LazarSoft/jsqrcode>)

```
<script src="static/bootstrap-4.3.1-dist/js/bootstrap.js"></script>
```

- 부트스트랩 스타일을 적용하기 위한 기본 파일

```
<script src="static/bootstrap-4.3.1-dist/js/bootstrap.bundle.js"></script>
```

- 부트스트랩 스타일 추가 속성을 사용하기 위한 파일

```
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
```

- 부트스트랩 제이쿼리 스타일을 적용시키기 위한 파일

2) CSS 파일

```
<link rel="stylesheet" type="text/css" href="static/css/mainStyle.css" />
```

- 메인 페이지 스타일

```
<link rel="stylesheet" type="text/css" href="static/css/modal.css" />
```

- 모달(팝업)창의 스타일

```
<link href="static/bootstrap-4.3.1-dist/css/bootstrap.min.css" />
```

- 페이지 전체에 대한 부트스트랩 적용 스타일

```
<link rel="stylesheet" href="static/bootstrap-4.3.1-dist/css/mdb.css" />
```

- 모달(팝업)창의 부트스트랩 적용 스타일

3. 메인 기능 함수 정보

main.js의 각 함수에 대한 내용은 다음과 같다.

- init : 페이지 로드시에 카메라 사용 여부를 묻고 불러온다.

```
async function init() {
  try {
    const stream = await navigator.mediaDevices.getUserMedia(constraints);
    //사용 가능한 카메라를 불러온다,
    handleSuccess(stream);
    //불러오는데 성공했다면 이 함수를 호출한다.
  } catch (e) {
    alert('cannot open camera : ' + e);
  }
  qrcode.callback = read;
}
```

- handleSuccess : 카메라 환경인 비디오를 스트림방식으로 재생시킨다.

```
function handleSuccess(stream) {
  window.stream = stream;
  video.srcObject = stream;
}
```

- read: 디코딩한 결과를 저장후 바이러스의 여부에 따른 모달(팝업)창의 테마를 변경시킨다.

```
function read(result) {
  current_decoded = result;
  var infoData = sendURL(result); // return JSON data
  modifyModalTitle(result);
  parse(infoData);
  if(infoData.virusInfo.positives > 0){
```

```

        changeModalTheme('danger');
    }
    else{
        changeModalTheme('info');
    }

    autoModal(current_decoded);
}

```

- modifyModalTitle : 제목의 길이가 25를 넘으면 그 이후 글자는 “...”으로 표시한다.

```

function modifyModalTitle(title){
    if(title.length > 25){
        title[25] = 0;
        title+="...";
    }
    $('#modalTitle').text(title);
}

```

- autoModal : 매개변수 값이 존재하면 모달(팝업)창을 보여준다.

```

// autoModal
function autoModal(result_url) {
    if (result_url != 0) {
        $(document).ready(function() {
            // Show modal on page load
            $("#qrResult").modal('show');
        });
        document.getElementById("web_print").innerHTML = result_url;
    }
    else {
        return;
    }
}

```

- sendURL : 동기방식으로서버의 주소 끝에 매개변수로 받은 url을 더해 GET요청을 하고 응답텍스트를 파싱한 값을 반환시킨다.

```
//GET JSON DATA about URL
function sendURL(url){
    var xhttp = new XMLHttpRequest();
    xhttp.open("GET", "https://moosong.iptime.org:8000/processing/"+url, false);
    xhttp.send();
    var raw = xhttp.responseText
    return JSON.parse(raw);
}
```

- changeModalClass : 매개변수로 받은 elem의 class 속성을 추가하고 삭제 시킨다.

```
//MODAL FUNCTION
//display modal-info or modal-danger
function changeThemeClass(classList, className, elem){
    for(var i = 0; i < classList.length ; i++){
        if(elem.classList.contains(classList[i])){
            elem.classList.remove(classList[i])
        }
    }
    elem.classList.add(className);
}
```

- changeModalTheme : 각 모달 테마에 따라 스타일을 다르게 적용시킨다.

```
function changeModalTheme(themeName){
    var elemBtn = $("#btnopen")[0];
    var elemOutlineBtn = $("#btnclose")[0];
    var elemModalColor = $("#modal-color")[0];

    titleclassList = ['modal-info', 'modal-danger'];
    btnclassList = ['btn-primary', 'btn-danger'];
    btnOutlineclassList = ['btn-outline-primary', 'btn-outline-danger'];
    switch(themeName){
        case 'info':
            changeThemeClass(titleclassList, 'modal-info', elemModalColor);
            elemBtn.style.display="";
            changeInfoNameColor("#0062CC");
            changeThemeClass(btnOutlineclassList, 'btn-outline-primary', elemOutlineBtn);
            break;
        case 'danger':
            changeThemeClass(titleclassList, 'modal-danger', elemModalColor);
            changeInfoNameColor("red");
            elemBtn.style.display="none";
            changeThemeClass(btnOutlineclassList, 'btn-outline-danger', elemOutlineBtn);
            break;
    }
}
```


- changeInfoNameColor : 각 모달 테마에 따라 스타일을 다르게 적용시킨다.

```
function changeInfoNameColor(color){
    var elemInfoName = $(".infoName");
    for(var i =0 ; i < elemInfoName.length ;i++){
        elemInfoName[i].style.color=color;
    }
}
//MODAL FUNCTION END
```

- parse : 주소와 whoisInfo, virusInfo를 파싱한다.

```
//START PARSE FUNCTION
function parse(infoData){
    parsescrInfo(infoData.scrInfo);
    parsewhois(infoData.whoisInfo);
    parseVirusTotal(infoData.virusInfo);
}
```

- parseInfo : 악성사이트일 경우 경고사진을 표시시킨다.

```
//parse scrInfo
function parsescrInfo(scrInfo){
    var elSiteImage = $("#siteImage")[0];
    if(scrInfo.imgPath === "malsite"){
        elSiteImage.setAttribute('src', 'static/images/malsite.png');
    }
    else{
        elSiteImage.setAttribute('src', 'static/images/'+scrInfo.imgPath);
    }
}
//end scrInfo
```

- clearText : whoisInfo와 virusInfo에 들어갈 내용을 초기화 시킨다.

```
function clearText(){
    //whois info
    var elDomainNames = $('#wDomainNames');
    var elRegistrar = $('#wRegistrar');
    var elOrg = $('#wOrg');
    var elCountry = $('#wCountry');
    var elCity = $('#wCity');
    var elAddress = $('#wAddress');
    var elEmails = $('#wEmails');
    //virustotal info
    var elScanDate = $('#vScanDate');
```

```

var elPositivesAndTotal = $("#vPositivesAndTotal");
var elScanResults = $("#vScanResults");
//clear whois info
acceptText(elDomainNames, "");
acceptText(elRegistrar, "");
acceptText(elOrg, "");
acceptText(elCountry, "");
acceptText(elCity, "");
acceptText(elAddress, "");
acceptText(elEmails, "");
//clear virustotal info
acceptText(elScanDate, "");
acceptText(elPositivesAndTotal, "");
clearScanResult();
}

```

- parseWhois : whoisInfo 정보를 추가시킨다.

```

//parse whoisInfo
function parsewhois(whoisInfo){
    clearText();
    var elDomainNames = $('#wDomainNames');
    var elRegistrar = $('#wRegistrar');
    var elOrg = $('#wOrg');
    var elCountry = $('#wCountry');
    var elCity = $('#wCity');
    var elAddress = $('#wAddress');
    var elEmails = $('#wEmails');
    acceptText(elDomainNames, whoisInfo.domainNames);
    acceptText(elRegistrar, whoisInfo.registrar);
    acceptText(elOrg, whoisInfo.org);
    acceptText(elCountry, whoisInfo.country);
    acceptText(elCity, whoisInfo.city);
    acceptText(elAddress, whoisInfo.address);
    acceptText(elEmails, whoisInfo.emails);
}

```

- whoisObjectsToString : 매개변수의 길이만큼 이름과 이메일을 합치고 개행

```

function whoisObjectsToString(objects){
    try{
        var contents = Array();
        for(var i = 0; i < objects.length; i++){
            contents = contents.concat(objects[i].name || objects[i].email);
        }
        return contents.join('<br>');
    }
    catch(e){

```

```

        console.log(e);
        return "";
    }
}

```

- acceptText : 매개변수 content의 타입이 'object'일 경우 와 'string'일 경우를 나누어 content의 text를 입력

```

function acceptText(jQueryElement,Content){
    if(typeof Content === 'object'){
        jQueryElement.html(whoisObjectToString(Content));
    }
    else if(typeof Content === 'string'){//string
        jQueryElement.text(Content);
    }
}
//end whoisInfo

```

- parseVirusTotal : virusInfo 정보를 추가시킨다.

```

//parse virusInfo
function parseVirusTotal(virusInfo){
    var elScanDate = $("#vScanDate");
    var elPositivesAndTotal = $("#vPositivesAndTotal");
    var positivesAndTotal = virusInfo.positives+ ' / '+virusInfo.total;
    var elScanResults = $("#vScanResults");
    acceptText(elScanDate,virusInfo.scanDate);
    acceptText(elPositivesAndTotal,positivesAndTotal);
    addScanResult(elScanResults,virusInfo.scanResult);
}

```

- addScanResult : 검사결과 항목의 속성을 변경시킨다.

```

function addScanResult(elScanResults,arrScanResults){
    //clear ScanResult before add scan result
    clearScanResult();
    if(arrScanResults.length === 0) // if not detected
        return; // pass
    else{
        $('#vScanResults').empty();// remove scanResults
        for(var i = 0; i < arrScanResults.length ;i++){
            //create div element and parse Machine Name
            elMachineName = document.createElement('div');
            elMachineName.setAttribute('class','col-4');
            elMachineName.textContent = arrScanResults[i].machineName;

```

```

        //create div element and parse Scan Result
        elResult = document.createElement('div');
        elResult.setAttribute('class', 'col-8');
        elResult.textContent = arrScanResults[i].result;

        //add scanresults
        elScanResults.append(elMachineName);
        elScanResults.append(elResult);
        elScanResults.append("<hr>");
    }
}
}

```

- clearScanResult : 검사결과 id값을 가져와서 class의 속성을 변경시키고 '발견되지 않음'을 출력

```

function clearScanResult(){
    var elScanResults = $('#vScanResults');
    var elNoResult = document.createElement('div');

    elNoResult.setAttribute('class', 'col-12');
    elNoResult.textContent = "발견되지 않음";
    elScanResults.empty();
    elScanResults.append(elNoResult);
}
//end virusInfo
//END PARSE FUNCTION

```

- captureToCanvas : 캡처한 QR코드 이미지를 가져오고 이를 복호화 시킨다.

```

function captureToCanvas() {
    try {
        context.drawImage(video, 0, 0);
        qrcode.decode();
        setTimeout(captureToCanvas, 3000);
    } catch (e) {
        console.log(e)
        setTimeout(captureToCanvas, 3000);
    }
}
//event handle

```

- setTimeout : 매 3초마다 화면을 캡처

```
//event handle
setTimeout(captureToCanvas, 3000);

$("#btnopen").on("click",function(){
    window.open(current_decoded, "_blank");
});
$("#btnclose").click(function() {
    $("#qrResult").modal('hide');
});
```