doi:10.20944/preprints201806.0078.v1

A Simple Four Templates Approach to Programming Android Cell Phones for Use in Clinical Research Projects.

Thomas Francis Hartley BSc(Hons), PhD.
School of Health Sciences
College of Health and Medicine
Newnham Campus
University of Tasmania
Launceston,
Tasmania 7250
Australia.

thomas.hartley@utas.edu.au

ABSTRACT

We describe how clinical researchers can exploit the Android cell phone as an economic platform for the gathering of data from clinical trial participants. The aim was to provide a solution with the shortest possible learning curve for researchers who are comfortable with setting up web pages. The additional requirement is that they extend their skills to the installation of a local webserver on the cell phone and then use four simple PHP templates to construct the clinical research data collection and processing forms. Data so collected is automatically written to local csv files on the cell phone. These csv phones can be retrieved from the device by the researcher simply by plugging the cell phone into their desktop PC and accessing the cell phone memory in just the same way as they would a USB memory stick. The results are presented as a list of recommended Android Apps along with settings that have proved to provide a stable combination likely to be easily used by clinical research participants. We have made a limited 'user trial' of this approach with satisfactory feedback received. We have concluded that this approach will reward researchers with a solution that is user friendly, will provide transcription free data and that is more than cost competitive with the conventional error prone/poor compliance 'paper based participant form – researcher transcription' cycle.

KEYWORDS

- health data science
- clinical trials
- research participant reporting
- personal health data diary
- personal private webserver
- research data integrity

INTRODUCTION

When health research is carried out amongst the wider community the paper questionnaire and more recently the internet questionnaire have been our tools of choice for gathering ongoing information from our participants. Today researchers cannot ignore the uptake of cell/mobile phones onto the very communities we are working with and the challenge is for us to exploit this phenomenon. The challenge to the non-programmer clinical researcher is how to find the time and knowledge to bring this into play? In our case we have been familiar with building websites for teaching and research projects and using the power of PHP scripts to do some or all of the data processing in the background. Because of its simplicity and our historical investment of time and resources into developing this approach we wished to capitalise on this and migrate it to the Android cell phone platform.

At the time of writing it was possible to purchase 3G and 4G cell phones at heavily discounted prices (less than $US45). These cell phones were locked to a designated telecommunications carrier, however, this did not preclude them from being used by us as powerful standalone Android tablet PCs. We have devised a solution to using these phones in any research or development project where there is a requirement to gather feedback from participants over a period of time. Once programmed these phones can be distributed to the participants for the duration of their involvement in the project. Investigators can download the data recorded by the participants at convenient intervals. In order to illustrate this point we have devised a template based questionnaire approach to the problem in a multipage format that is better suited to the small format screens of these phones. We envisage that these simple templates can be edited by other researchers with minimum programming skills to accommodate their particular requirements by following the principles behind our schema and templates.

OBJECTIVES

- **MAIN OBJECTIVE :** To see how feasible it would be to use inexpensive 'surplus' 3G phones as inexpensive data acquisition devices in clinical R & D projects. The concept was that if they are inexpensive and can be easily adapted to the researcher's needs then it would become feasible to distribute them to all the participants in a hypothetical clinical trial and ask them to use these devices to record their 'adverse events data' in 'real time' and 'off site'.

- **SECONDARY OBJECTIVE :** To demystify the idea that Bespoke Apps on the Android platform for R & D are only achievable if you have a professional programmer to write them for you.

doi:10.20944/preprints201806.0078.v1

MATERIALS AND METHODS

Hardware : ZTE T815 3G cell phones running Android Version 4.4.2 and marketed in Australia as Telstra Tempo ($29 from Australia Post Shop, locked to Telstra Network). (Kernal Version 3.4.67, 1.27 GB memory, 4 inch screen).

Platform Used to Develop the Software : Development of the PHP software component was impractical on the phone because its small screen size and limitations of the touch keyboard. Instead a copy of Server2Go (server2go.jaleco.com) was installed on a USB memory stick as specified by the developers. This provided us with a PC based platform running a local webserver environment almost identical to the Palapa webserver used on the phone. Once debugged the PHP script was copied to the phone memory. Debugging of the screen layouts were necessarily carried out on the phone. Figure 1 illustrates where these files were located in the pws/www folder on the phone; note that some of the files were off screen in this screen capture.

Android Specific Software : In order to avoid the need for high level programming skills we discounted approaching the problem via any of the usual Android programming packages or languages. Instead we opted for setting up the system as a self-contained 'web site' on the cell phone itself. This idea stemmed from the recognition of a body of existing solutions on the www that have used a combination of 'web browser + cascading style sheets + Java Script' to create an impression with the user of dynamic interactivity; Stark et al [1]. We did experiment with this for a while and came to three conclusions. The first two were that

- Java Script is not a trivial language to learn or implement
- The HTML5 Document Object Model, (DOM), is similarly a complex

These comments were not made to detract from the approach taken by others but were really a reflection of our experience that both were too complex for our simple needs and certainly would not be transferable to others without a 'learning curve' on their part.

The third and deciding factor in our decision was that we realised that browsers plus Java Script are intentionally designed to make any writing back of permanent data on the 'client' device difficult for longstanding security reasons that have surrounded unsecure browsing of the internet. It was then that we realised that the approach that would work would be to install a web server and website on the phone and use the simple HTTP request on the phone's browser "localhost:8080/". From our previous experience, it then followed that we could execute any type of data manipulation and storage within the web server environment using the PHP language. PHP is a relatively simple language and we have used it to build simple Forms as web pages and then write the returned data back to simple comma delimited files (.csv files) that can be downloaded off the phone via USB cable or Bluetooth into the investigator's PC. Once there the file type is compatible for further analysis using Excel or more advanced packages such as SPSS or Stata etc.

The combination of packages that produced the required performance and simplicity were :

- Palapa Web Server : Version 2.1.1 from alfanla.com/palapa-web-server
- Opera web browser : Version 41.2.2246.111806 from www.opera.com/mobile/operabrowser/android
- TouchPal Keyboard : Version 5.4.1.1 from touchpal-keyboard.en.uptodown.com/android/old
- Adobe Acrobat Reader : Version 17.4.1.178813 from www.adobe.com

The Palapa Web server (Figure 2) was chosen because it is a free of charge Android application and loaded seamlessly in the background whenever the phones were turned on. The user was totally unaware that this was happening. Furthermore, it required no adjustment to its default settings in order to run as we required for the project under development.

The phone came preloaded with an Android generic browser as well as Google Chrome. Neither of these behaved well when required to browse the "localhost:8080/". This turned out to be a well recognised issue and has been regularly reported on the www. We loaded the Opera browser and this proved to behave consistently and to always render our web pages to full screen by simply 'double tapping' the screen. We noted that this was also the best performing browser to use if for any reason, we had wanted to present material in any non-HTML or simple text format. For example 'Help' material was best included as .pdf files and these were handled well by the Opera browser; the browser downloaded the file and this file was then be opened from Opera's Download folder using Adobe Reader. We edited the Home Screen of the phone so that on power up it only showed the Opera web browser icon. Similarly we edited the opening Speed Dialling page of Opera so that it only showed one shortcut and that was to the home page of our App. All the files we describe here were placed in a subdirectory of the Palapa Server's www directory; see Figure 1. Subdirectory naming is user definable but the syntax for the Speed Dial icon in Opera would have to follow this form :

localhost:808/subdirectory1/subdirectory2/subdirectoryn/q1.html

We intentionally removed the SIM card from the phone and would discourage users from activating the phone's WiFi. We did this because we did not want the participant to contend with various updates which get 'pushed' onto the phone by Google and other App developers including Opera and Adobe. We wished to ensure that the platform left us in a state that we knew would perform correctly and we were concerned that if any of these updates had been inadvertently loaded by the participant then there was a risk that the platform would not run as we had intended.

The cell phone we worked with had the option of two keyboards to use whenever user input was required. We chose to set the default keyboard in the phone's 'Settings' to the TouchPal option. We did this for two reasons. Whenever an HTML Form was presented to the user that required numeric input then it presented a keyboard that was extremely clear and also included the decimal point (see left-hand panel of Figure 3). It is not unusual for an Android keyboard to miss out the decimal point and although this can be circumvented programmatically it was a complication we wished to avoid in the interests of keeping our solution simple. The other feature was the 'tool' in the toolbar that looked like pencil. This opened up a pad which included a very practical cursor pad (see right-hand panel of Figure 3). Moving a cursor on a small screen touch phone requires extremely good 'touch skills' and we viewed this 'tool' as being a definite advantage to users of our phones that were challenged when it came to moving a cursor back over some of their text that needed correction. We would encourage the use of a stylus. The so called 'Zombie Finger' syndrome affects different users and different phones in unpredictable ways so to reduce the risks we would recommend that a cheap stylus be provided with the phone to all participants. It is worth noting that that in clinical trial projects where participants are unwell then their motor skills are not going to be as good as usual so the stylus can make 'hitting the target' key much easier for them. We would also recommend that any phone selected for this type of project should have a 5 inch screen.

Local Website Schema and Files :

Normally in a clinical trial there are only five generic questions that need to be included in a participant's feedback form :

- What is your ID number ?
- What are your symptoms ?
- What was the date and time you experienced these symptoms ?
- How would you rate these symptoms ?
- What was your Glucometer glucose reading when this happened ?

The fifth question in this example is specific but the generic question would usually be along the lines of providing some objective information from the participant. Under the desktop PC environment a single HTML Form webpage would have sufficed but because of the small screen size of a cell phone we realised that we could only give the user a reasonable amount of space in which to write their responses if each question was placed onto a sequence of separate webpages. This presented the problem of how to carry forward the data in the responses given in the sequence of webpages to the final data filing and processing steps of the PHP form handling script. The solution was to exploit the ‘sessions’ facility on the web server and use unique session variables with each question so that they could be carried forward to the final PHP form handling script, (see http://www.html-form-guide.com/php-form/php-order-form.html for a fuller explanation of this concept.)

RESULTS

The simple ‘demonstration’ website reported here was composed of six files viz. one HTML file, five PHP files, and one CSV file. These accommodated the on screen formatting of the five questions, each of the five questions and the appending of all the ‘time and date stamped’ responses into a single CSV file.

‘Question 1’ had to be an HTML Form that ‘POSTED’ its response data to the ‘Question 2’ PHP file. That file recorded the posted information from Question 1 into a unique SESSION variable that persisted from question to question. The demonstration had only four session variables but if for example there was a requirement for ten questions then there would be a requirement for nine session variables. Figure 4 shows the schema of the logic and files required for a questionnaire of N questions long. Listings of q1.html, qn.php (this is the nth question in the series of a total of N questions), qN.php and process.php are given below. Other researchers can adapt these to their local requirements by editing the highlighted text accordingly. Initially a style sheet was used but this proved to behave unreliably when we moved the templates to phones with different screen sizes. Instead we persisted with using elements that are technically ‘deprecated’ elements under the current HTML 5 standard. The Opera browser was able to reliably render these deprecated statements in accordance with the pre-HTML 5 specification. The questionnaire was modelled around the clinical trial of a fictional anti-inflammatory product called Wattaren .

- Listing of q1.html : Template for the first question

```
<html>
<head>
<meta name=”viewport” content=”width=320”/>
</head>
<body>
<body bgcolor='green'>
<font size=3>
<font face=’Arial’>
<font color='white'>

<h3>WATTAREN ANTI-INFLAMMATORY CLINICAL TRIAL
</h3>

Participant's Adverse events recording system

<p>
Use this form to record any adverse reactions during this clinical trial of
Wattaren
<p>

You should use this App to record these events as close as possible
to the time at which they occurred
<p>
```

```
There are 5 questions which are presented one question at a time .......


<form action="q2.php" method="post">
Q1 : What is your ID number ?
<p>
<input name="idnumber" type="number">
<p>
<input type="submit" name="Submit" value="Next question ..">
</form>
</div>
</div>
</body>
</html>
```

➢ Listing of qn.php : Template for the nth question

```
<?php
session_start();
$_SESSION['symptoms'] = $_POST['symptoms'];
?>

<html>
<head>
<meta name="viewport" content="width=320"/>
</head>
<body>
<body bgcolor='green'>
<font size=3>
<font face='Arial'>
<font color='white'>

<h3>WATTAREN ANTI-INFLAMMATORY CLINICAL TRIAL
</h3>
Participant's Adverse events recording system
<p>
<form action="qn+1.php" method="post">
Q3 : What was the date and time you experienced these symptoms ?
<p>
<input type='datetime-local' name='day'>
<p>
<input type="submit" value="Next question ..">
</form>
</div>
</div>
</body>
</html>
```

➢ Listing of qN.php : Template for the Nth / Last question

```
<?php
session_start();
$_SESSION['rating'] = $_POST['rating'];
?>

<html>
<head>
<meta name="viewport" content="width=320"/>
</head>
<body>
<body bgcolor='green'>
<font size=3>
<font face='Arial'>
<font color='white'>

<h3>WATTAREN ANTI-INFLAMMATORY CLINICAL TRIAL
</h3>
Participant's Adverse events recording system
<p>
```

```
<form action="process.php" method="post">
Q5 : What was your Glucometer glucose reading when this happened ?
<p>
<input name='glucose' type='number' step=0.1>
<p>

<input type="hidden" id="dmy" name="NOW" value="Content of the extra variable" >

<script>
var d = Date();
document.getElementById("dmy").value = d;
</script>


<p>
<input type="submit" value="File your answers ..">
</form>
</div>
</div>
</body>
</html>
```

- Listing of process.php : Template for processing and filing the data from all N questions plus a time and date stamp

```
<?php
session_start();
?>

<html>
<head>
<meta name="viewport" content="width=320"/>
</head>
<body>
<body bgcolor='green'>
<font size=3>
<font face='Arial'>
<font color='white'>

<h3>WATTAREN ANTI-INFLAMMATORY CLINICAL TRIAL
</h3>
Participant's Adverse events recording system
<p>
Thanks for completing the Form ... here is what has been recorded in your
logfile...
<p>
<b>
<i>
Please note that if you have rated your symptoms as SEVERE or VERY SEVERE then you
must contact the Clinical Trials Helpdesk via the phone number in your Study
Participant's Handbook.
</i>
</b>
<br>

<p>
<?php
$a =  $_SESSION['idnumber'];
$b =  $_SESSION['symptoms'];
$c =  $_SESSION['day'];
$d =  $_SESSION['rating'];
$e =  $_POST['glucose'];
$f =  $_POST['NOW'];

print "This is what has been written to your log file :";
print "<br>";
print "Your ID number : " . $a;
print "<br>";
print "Your symptoms : " . $b;
print "<br>";
print "Date and time this happened : " . $c;
print "<br>";
print "Severity of your symptoms  : " . $d;
print "<br>";
print "Your blood glucose at around that time : " . $e;
print "<p>";
print "---------------debugging-----------------";

print "<p>";

$x = $a . ", " . $b .", " .  $c .", " . $d .", " . $e .", " . $f ."\n";

print $x;
print "<p>";
print "--------------------------------";
$fp = fopen("generallogfile.csv", "a");
fputs($fp, $x);
fclose($fp);

session_unset();
session_destroy();
?>
</div>
</div>
</body>
</html>
```

Other data input fields that can be used in HTML forms were readily available on the internet. We used the website www.phpform.org to obtain the correct syntax for our input types and we would recommend this approach for developers who are not confident in designing HTML forms beyond the templates that have been described here. Their approach would be to design a dummy form on that site with just the input type they require and then make minor edits of the downloaded HTML script to suit their needs and then paste it in as a new ‘question’. The input types that were available are shown in Figure 5.

The Screens : We found that if the Opera rendered any of the webpages smaller than full screen then a simple ‘double tap’ on this combination – Opera Browser on a ZTE T815 cell phone – would always render our pages to full screen. There appears to be a general consensus on the www that this is a problem common to all browsers – it appears that it is programmatically very difficult to ensure that a webpage will always be rendered at full screen across all Android mobile devices.

Figure 6 illustrates the opening HTML page of the questionnaire which provided a minimal introduction to the ‘app’ in the context of the fictional ‘Wattaren Clinical Trial’ and the first question. In the Opera browser as soon as the ID Number field was ‘tapped’ then a bold numeric keypad opened up as an overlay without rendering the first question and input box invisible.

Figure 7 illustrates the screen presented by the first PHP scripted webpage. Again we found the Opera browser popped up the QWERTY keyboard without obscuring the question. That keyboard also had the ‘tool’ that looked like a pencil and the right-hand screen shot in Figure 7 shows how useful this was for moving the cursor back to the position of a misspelt word – 'saw' instead of the intended 'sore'.

Question 3 required the entry of a date on which the symptoms were experienced. Figure 8 shows that the Opera browser has rendered the date entry box with a downward arrow. As soon as this was tapped then the Opera browser overlayed a current date and time scrolling 'chooser'. It defaulted to open at the then current date and time so if the user was making the entry at the time of the symptoms then simply tapping 'set' completed the date and time field. Moving to earlier dates and times was achieved by 'swiping' the zones in calendar until the appropriate date and time was reached. Tapping 'set' button completed the date and time field and closed the calendar overlay.

The responses to question 4 could only be selected from a dropdown list. Figure 9 illustrates the Question 4 PHP generated screen and on the right the options in the dropdown list that we had programmed. The user had to tap on their selection and then tap anywhere outside of the selection overlay for it to close and autocomplete the data entry field for this question.

The final question required a numeric input for the participant's blood glucose concentration. Again the numeric keypad opened up as soon as the data entry cell was tapped (see Figure 10). This time the usefulness of having a numeric keypad with a decimal point on it was realised because blood glucose concentrations are always measured to the first decimal place when using mmol/L units for glucose. On the right-hand side of Figure 10 is the closing screen presented after the 'File your answers' button had been tapped at Question 5. This screen was just about legible when 'double tapped' but a second 'double tap' rendered it very legibly across two screens as shown in the right-hand half of Figure 10.

Results of a limited Peer Review :

The concepts and design were presented to peer staff in the School at a seminar and attendees we invited to go to a website where the cell phone styled questionnaire had been set up, (www.medlabstats.com/seminar/q1.html). They used their cell phones to call the questionnaire up and then asked to score the usability. Participants with iPhones and the web browser Safari gave it a low score which confirmed for us that the choice of a designated browser viz. Opera had been a wise decision. The other users who all had Android cell phones of various makes gave the app a median score of 9 out 10. Ideally a fuller evaluation would have involved issuing multiple prepared cell phones to staff but this was not possible.

doi:10.20944/preprints201806.0078.v1

DISCUSSION

This project demonstrated to us that several criteria 'for a good app' can be met if the researcher is prepared to invest time and effort along some simple but quite disciplined lines.

The first is to come to terms with the limited screen size and word their questions in as succinctly as possible. Participants in clinical research projects are busy people and may also not be feeling particularly well so it is important to present them with a task that is as simple and as intuitive as possible. We believe that the resources that we happened upon during our researching of web based surveys in combination with the TouchPal keyboard will meet the majority of respondent's expectations when it comes to legibility, ease of resizing and simplicity of use.

The second is to carefully edit the templates provided. The higher powered features of HTML and CSS are not needed for these simple pages and so there is no need for the researcher to go further than the basic tags used. PHP is not a high level programming language and is not difficult to read and understand PHP programs. We purposely avoided doing anything with PHP other than to get it to build simple HTML pages and then to file the gathered data to the phone's internal memory. We envisaged that all higher level processing and manipulation of the data file would be done elsewhere on a conventional desktop PC loaded with the user's preferred software packages. The only requirement is that those software packages must be able to import data in .csv file format.

CONCLUSIONS

Using this approach of HTML and PHP templates busy researchers should be able to put together useful Apps in a short period of time. The cell phone is well integrated into our communities so the training time to bring a participant up to speed will be short. From a cost perspective the 'locked' mobile phones are cheap and cost competitive with printed questionnaires when you take into account the costs of printing questionnaires, mail outs, mail backs and time spent by the researcher transcribing the data from the questionnaire into the database etc . Data cleansing is a perennial problem in clinical research and it often hinges around the legibility of the participant's hand writing on the forms. Electronic typos by participants using this cell phone approach can be expected to be less ambiguous and the decision by the researcher to exclude or include a questionable entry can be expected to be easier and quicker.

CONFLICTS OF INTEREST

The author has a personal professional website; www.medlabstats.com

ETHICS

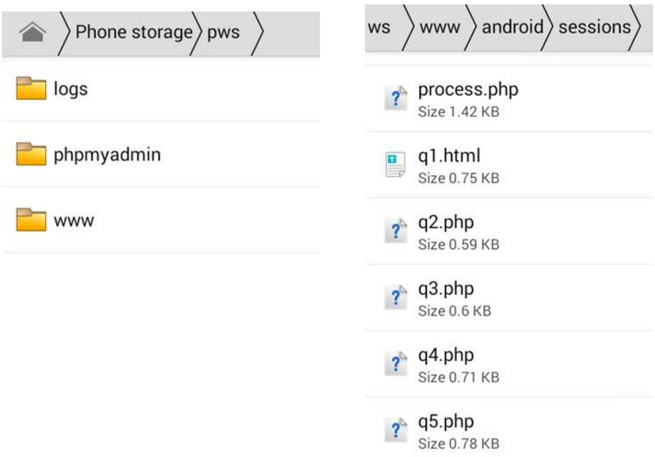
This work did not involve research using human subjects. All health data collected during all evaluations were simulated

REFERENCES

1. Stark J, Nathan P, Papaconstantinou J, Lagerstrom P, Hope P. 'Building Android Apps with HTML, CSS, and JavaScript' Published by O'Reilly Media, ISBN , September 2010.

FIGURES

Figure 1: File structure on the cell phone



Phone storage 〉 pws 〉

logs

phpmyadmin

www

ws 〉 www 〉 android 〉 sessions 〉

process.php
Size 1.42 KB

q1.html
Size 0.75 KB

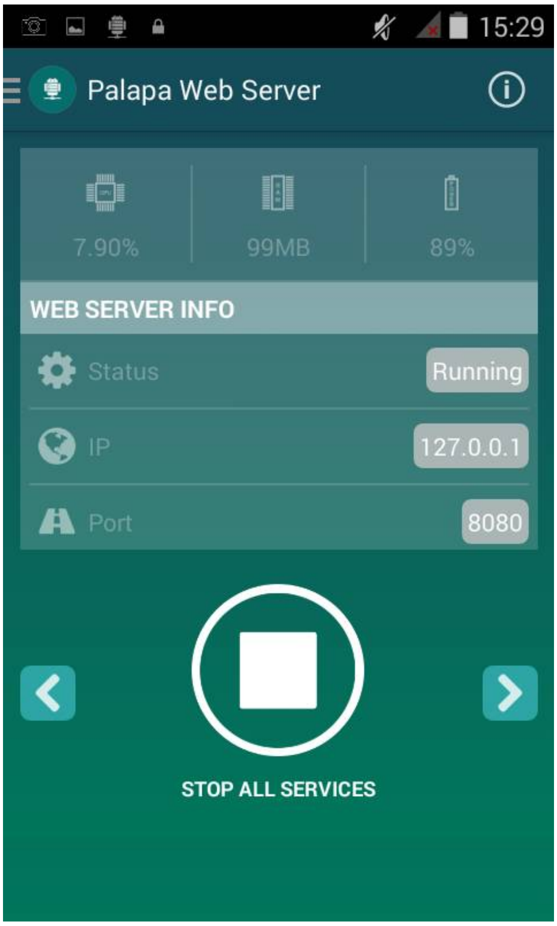
q2.php
Size 0.59 KB

q3.php
Size 0.6 KB

q4.php
Size 0.71 KB

q5.php
Size 0.78 KB

Figure 2 : Palapa Webserver console screen



15:29

Palapa Web Server

7.90%

99MB

89%

WEB SERVER INFO

Status Running

IP 127.0.0.1

Port 8080

Figure 3 : Numeric and cursor keypads available in TouchPal

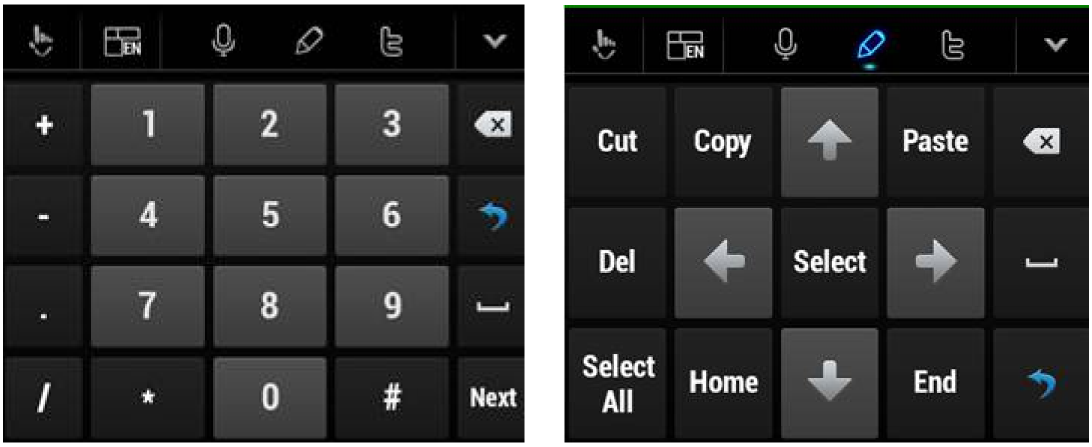
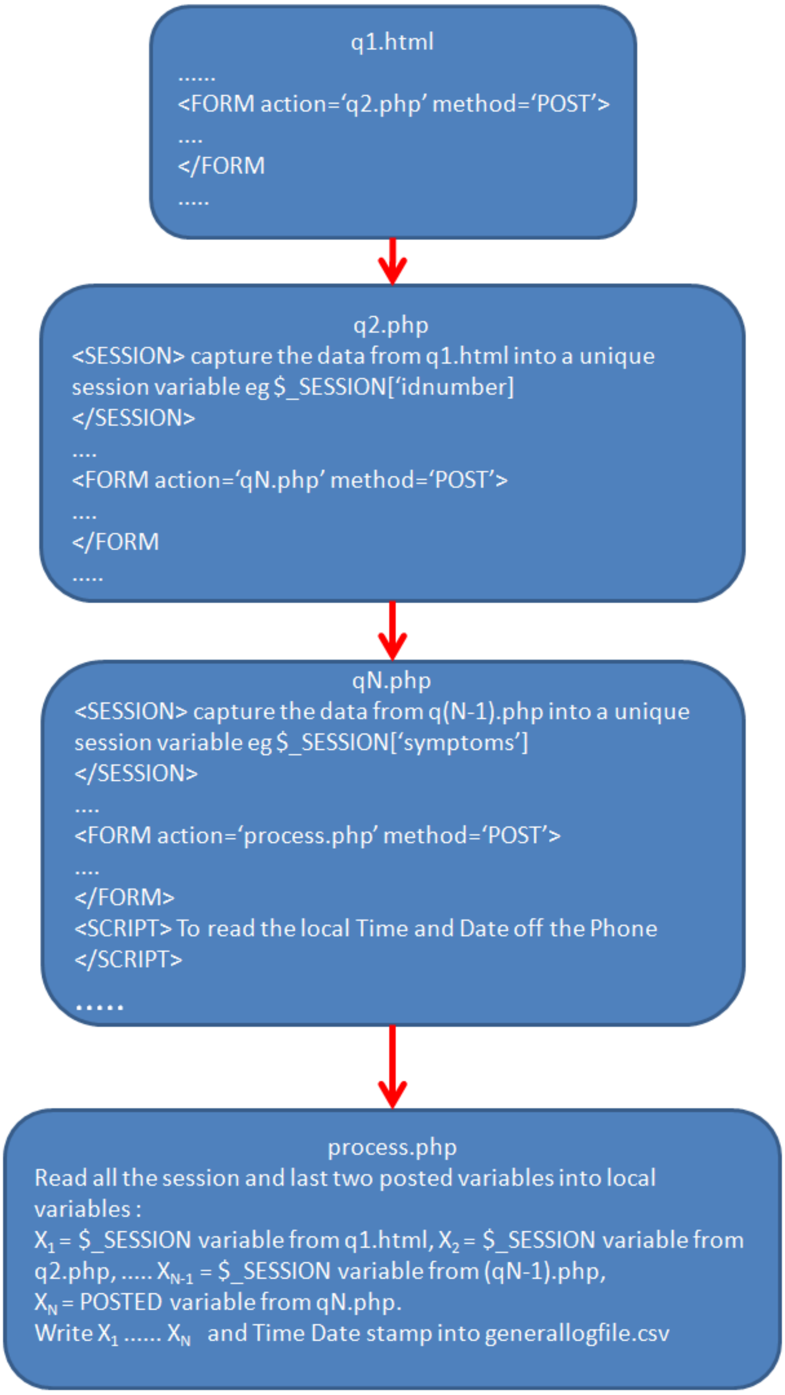


Figure 4 : Generalised schema for an ‘N’ questions feedback form.



```
q1.html
......
<FORM action=‘q2.php’ method=‘POST’>
....
</FORM
.....
```

```
q2.php
<SESSION> capture the data from q1.html into a unique
session variable eg $_SESSION[‘idnumber]
</SESSION>
....
<FORM action=‘qN.php’ method=‘POST’>
....
</FORM
.....
```

```
qN.php
<SESSION> capture the data from q(N-1).php into a unique
session variable eg $_SESSION[‘symptoms’]
</SESSION>
....
<FORM action=‘process.php’ method=‘POST’>
....
</FORM>
<SCRIPT> To read the local Time and Date off the Phone
</SCRIPT>
.....
```

process.php

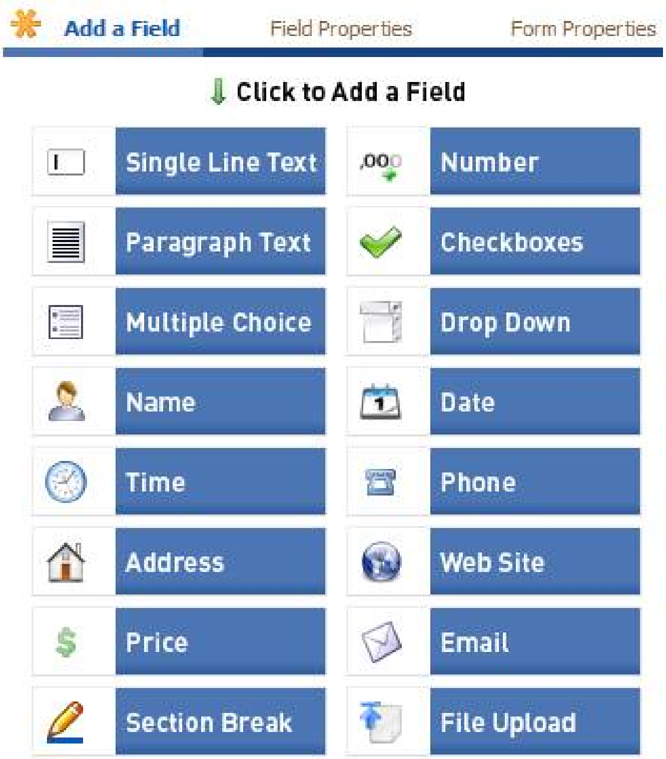
Read all the session and last two posted variables into local variables :

X_1 = $_SESSION variable from q1.html, X_2 = $_SESSION variable from q2.php, ... X_{N-1} = $_SESSION variable from (qN-1).php,

X_N = POSTED variable from qN.php.

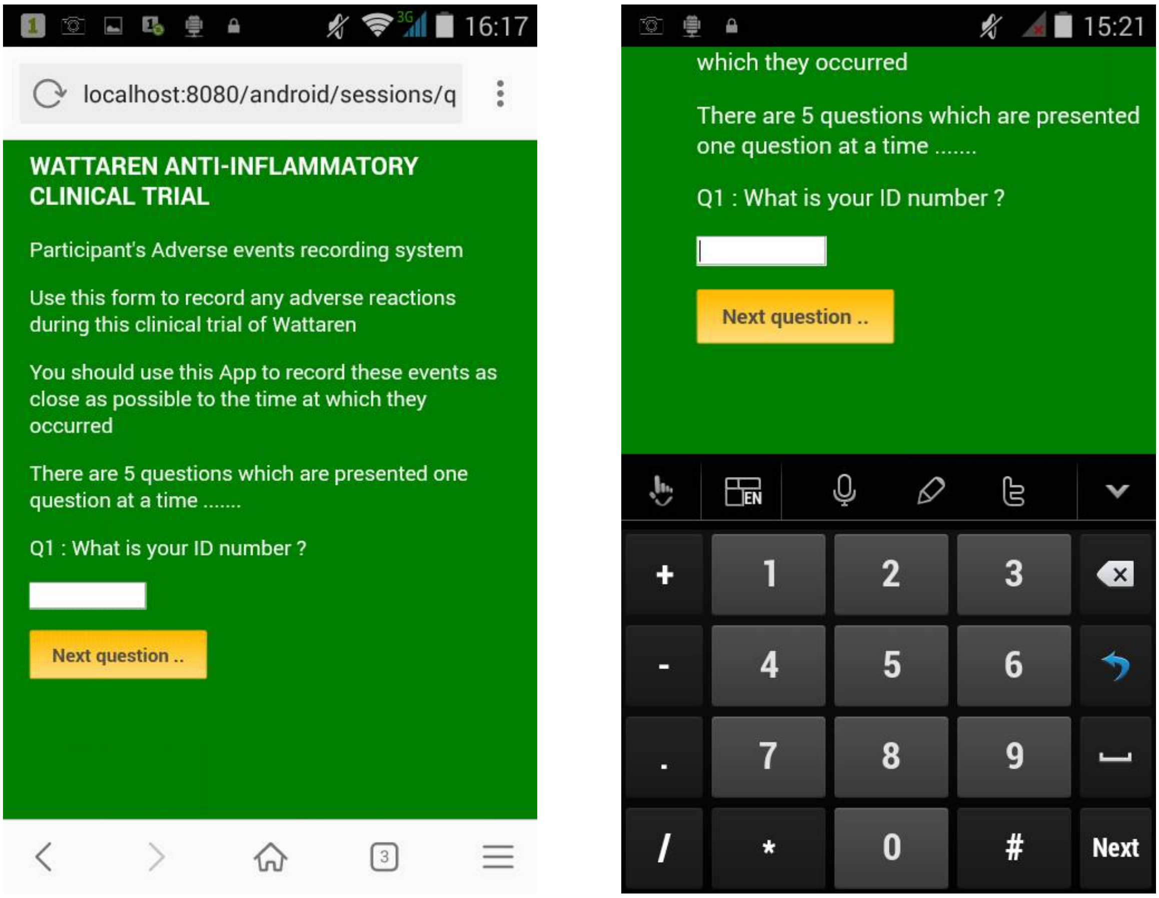
Write X_1 ...... X_N and Time Date stamp into generallogfile.csv

Figure 5 : The menu of HTML Form components available in the phpform.org 'wizard'



doi:10.20944/preprints201806.0078.v1

Figure 6 : Opening page of the clinical trial participant's feedback form. Numeric response is required and the TouchPal numeric keypad opens automatically.



WATTAREN ANTI-INFLAMMATORY CLINICAL TRIAL

Participant's Adverse events recording system

Use this form to record any adverse reactions during this clinical trial of Wattaren

You should use this App to record these events as close as possible to the time at which they occurred

There are 5 questions which are presented one question at a time ....

Q1 : What is your ID number ?

Next question ..

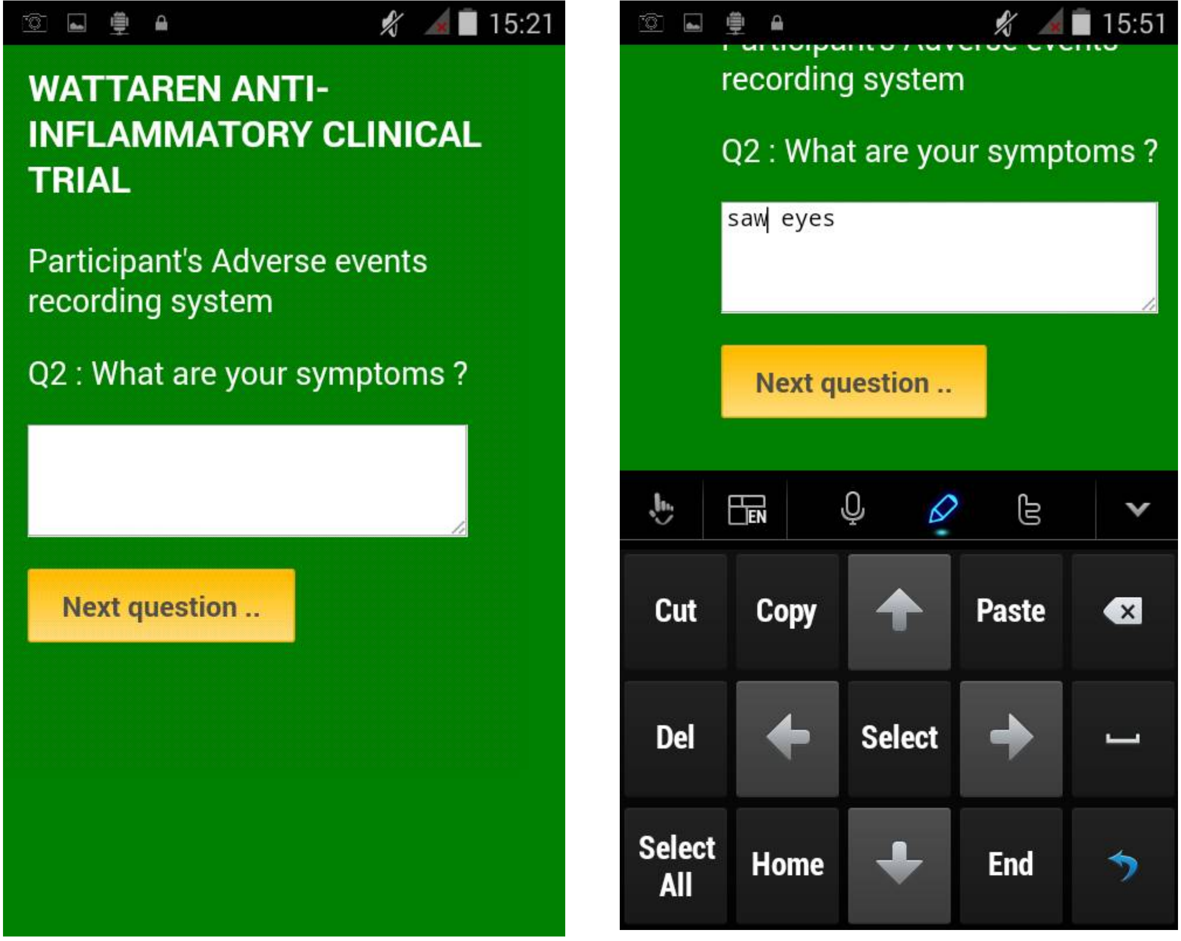
which they occurred

There are 5 questions which are presented one question at a time ....

Q1 : What is your ID number ?

Next question ..

Figure 7 : Webpage generated by the PHP script after the completion of the first question in the series



WATTAREN ANTI-INFLAMMATORY CLINICAL TRIAL

Participant's Adverse events recording system

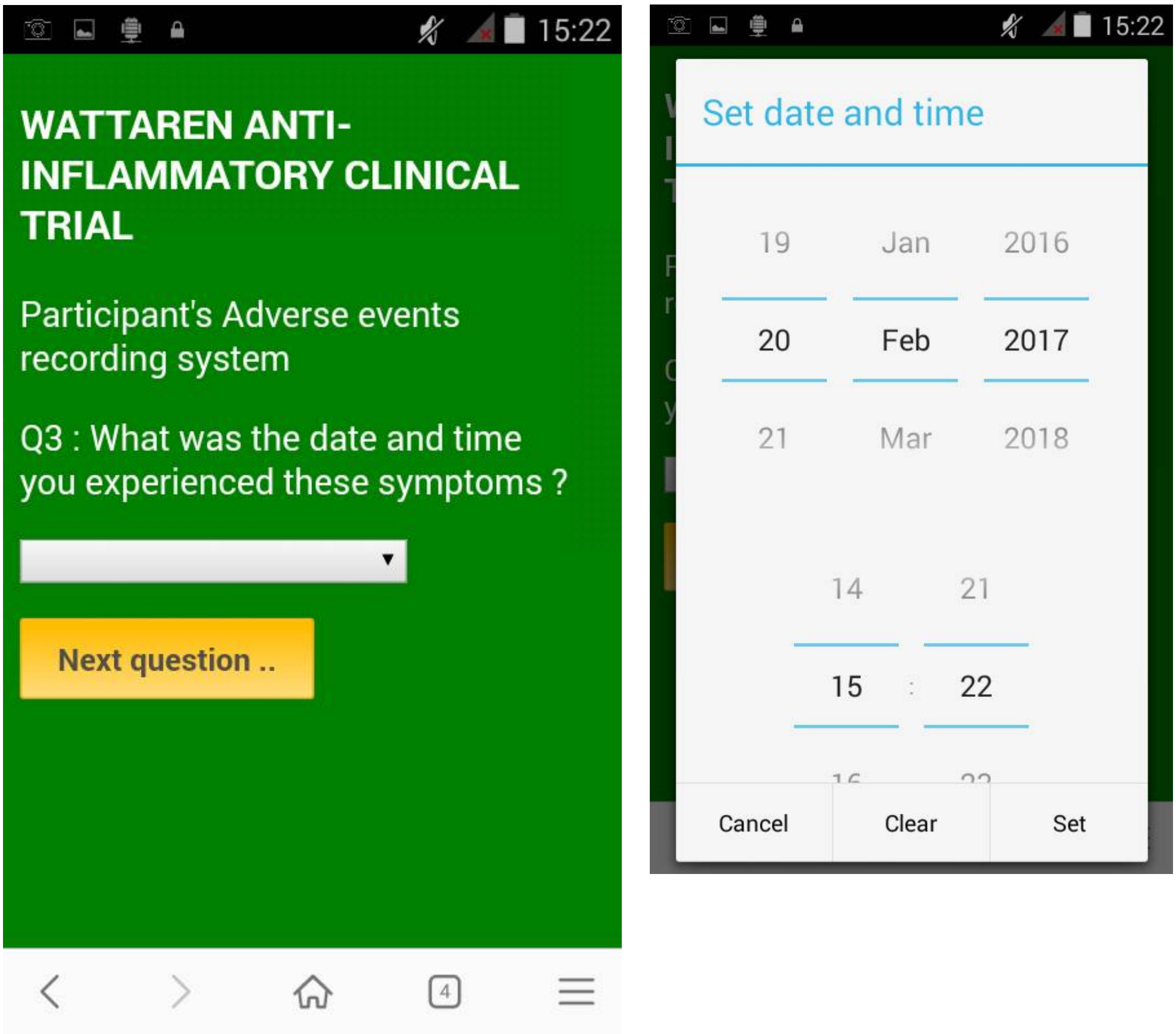
Q2 : What are your symptoms ?

recording system

Q2 : What are your symptoms ?

saw eyes

Figure 8 : This page requires a date response. The TouchPal keyboard opens the calendar keypad automatically in anticipation of this requirement.



WATTAREN ANTI-INFLAMMATORY CLINICAL TRIAL

Participant's Adverse events recording system

Q3 : What was the date and time you experienced these symptoms ?

Set date and time

20 Feb 2017

15 : 22

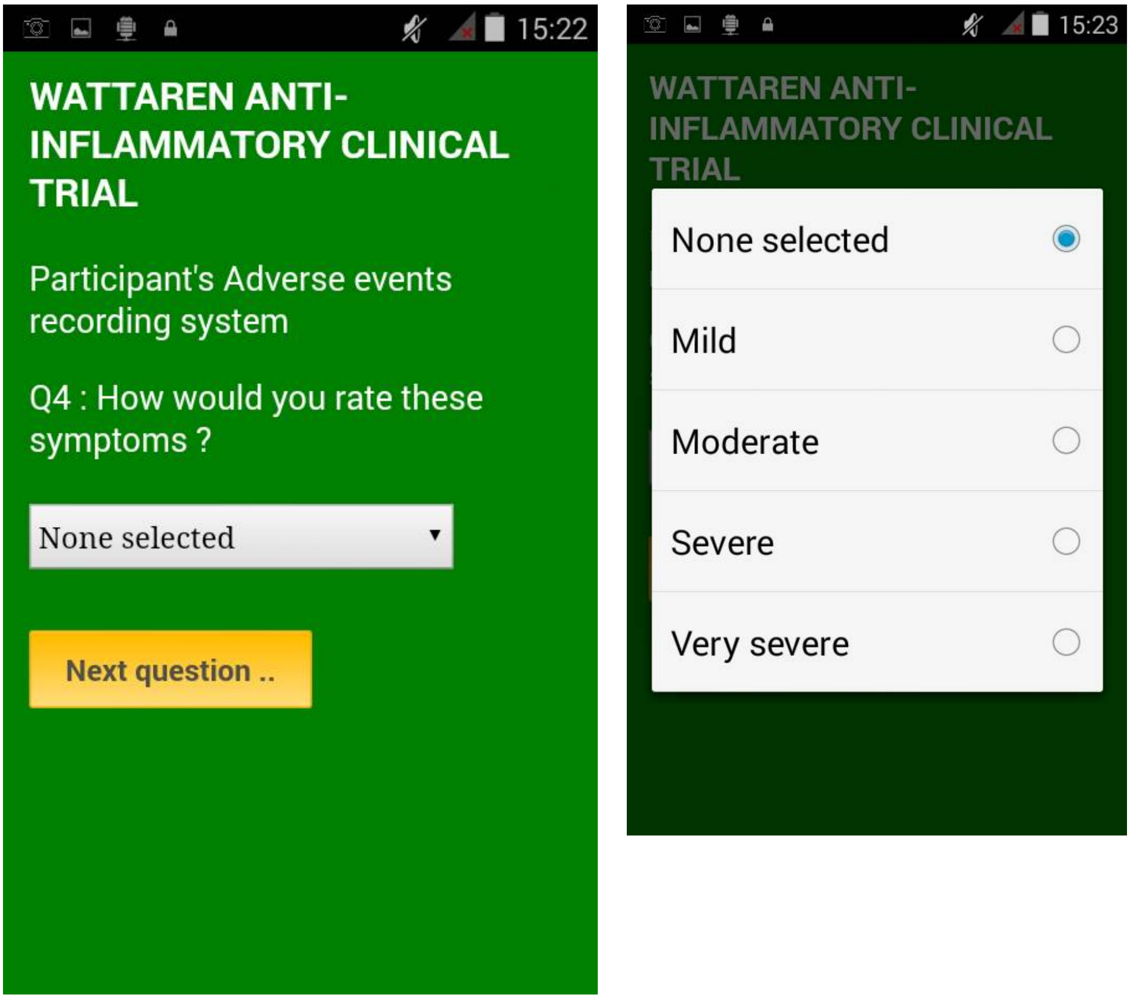


Figure 9 : Question 4 requires the participant to make a selection from a dropdown list. The TouchPal keyboard anticipates this requirement by displaying the list full screen.

15:22

WATTAREN ANTI-INFLAMMATORY CLINICAL TRIAL

Participant's Adverse events recording system

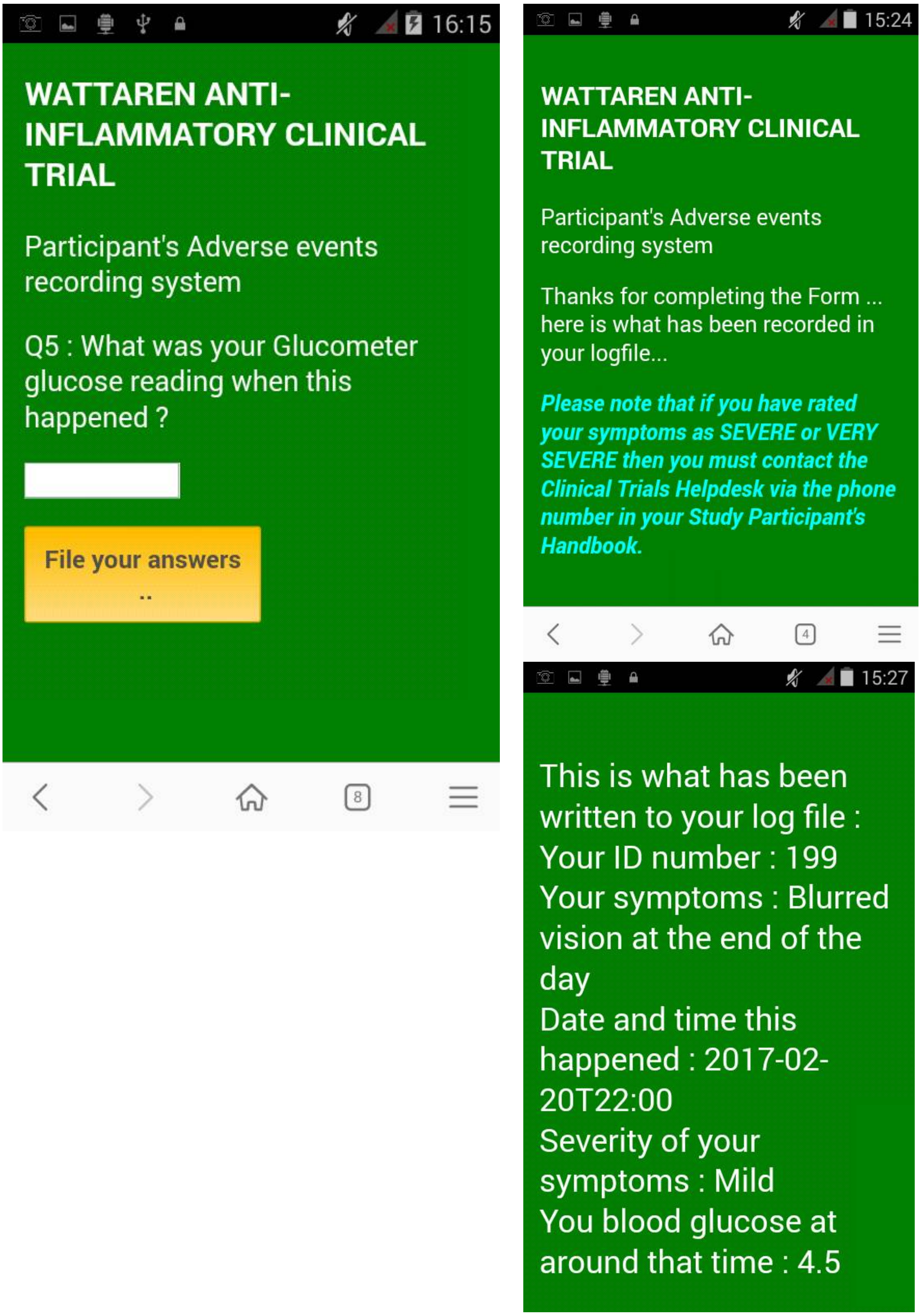
Q4 : How would you rate these symptoms ?

None selected ▾

15:23

WATTAREN ANTI-INFLAMMATORY CLINICAL TRIAL

Figure 10 : The final data entry screen and farewell message screen.



16:15

WATTAREN ANTI-INFLAMMATORY CLINICAL TRIAL

Participant's Adverse events recording system

Q5 : What was your Glucometer glucose reading when this happened ?

15:24

WATTAREN ANTI-INFLAMMATORY CLINICAL TRIAL

Participant's Adverse events recording system

Thanks for completing the Form ... here is what has been recorded in your logfile...

Please note that if you have rated your symptoms as SEVERE or VERY SEVERE then you must contact the Clinical Trials Helpdesk via the phone number in your Study Participant's Handbook.

15:27

This is what has been written to your log file :
Your ID number : 199
Your symptoms : Blurred vision at the end of the day
Date and time this happened : 2017-02-20T22:00
Severity of your symptoms : Mild
You blood glucose at around that time : 4.5