openXdata Mobile Charts: Instant graphic reports on mobile devices

Prototype software development and system usability survey

Owaisuddin Ahmed

Centre for International Health
Faculty of Medicine and Dentistry
University of Bergen, Norway

2011

openXdata Mobile Charts: Instant graphic reports on mobile devices

Prototype software development and system usability survey

Owaisuddin Ahmed

This thesis is submitted in partial fulfilment of the requirements for the degree of Master of Philosophy in International Health at the University of Bergen.

Centre for International Health
Faculty of Medicine and Dentistry
University of Bergen, Norway
2011

ABSTRACT

Researchers working in international health may require the use of charts to quickly visualize data and see trends. These may be required at frequent intervals. The openXdata Mobile Charts prototype was developed as software that can automatically create charts by reading a database and making the charts available for viewing through a password-protected web application tailored for mid-range mobile phones. A system usability survey was conducted to determine how user-friendly the software is. Participants rated ten items relating to user-friendliness using options from "strongly disagree" to "strongly agree." An eleventh item with adjective rating options for overall user-friendliness was added to determine how closely it matched with the score of the ten preceding items. The mean score for the ten-item system usability survey was 88.1 out of 100 while the mean score for the adjective rating was 85.0. A strong positive correlation (r = 0.914) was found between survey scores and adjective ratings. The prototype of openXdata Mobile Charts was given high ratings indicating that a production version of the software may be beneficial in fulfilling some visual reporting tasks.

Table of Contents

ABSTRACT 1
ACRONYMS AND ABBREVIATIONS 3
ACKNOWLEDGEMENTS 4
1. INTRODUCTION 5
2. OBJECTIVES 13
3. METHODS AND DATA COLLECTION 14
3.1 SOFTWARE DEVELOPMENT METHODS 14
3.2 DATA COLLECTION METHODS 24
3.2.1 QUESTIONNAIRE SELECTION 24
3.2.2 PARTICIPANT SELECTION 30
3.2.3 SURVEY CONDITIONS 31
4. RESULTS 32
4.1 RESULTS OF SOFTWARE DEVELOPMENT 32
4.1.1 DESCRIPTION AND USAGE 32
4.1.2 DEVELOPMENT NOTES AND ISSUES 37
4.2 RESULTS OF SURVEY - SYSTEM USABILITY SCALE AND ADJECTIVE RATING 44
4.2.1 PRE-SURVEY FINDINGS 44
4.2.2 SURVEY FINDINGS – SYSTEM USABILITY SCALE 45
4.2.3 SURVEY FINDINGS – ADJECTIVE RATING 47
4.2.4 SURVEY FINDINGS – PARTICIPANT COMMENTS 47
5. DISCUSSION 50
5.1 LIMITATIONS 51
6. CONCLUSIONS AND RECOMMENDATIONS 54
6.1 CONCLUSIONS 54
6.2 RECOMMENDATIONS 54
REFERENCE LIST 56
ANNEXES 57

ACRONYMS AND ABBREVIATIONS

API	Application Programming Interface
CSUQ	Computer System Usability Questionnaire
GIS	Geographic Information System
GUI	Graphical User Interface
GWT	Google Web Toolkit
IRD	Interactive Research and Development
KML	Keyhole Mark-up Language
OMEVAC	Open Mobile Electronic Vaccine Trials
oxd MC	openXdata Mobile Charts
QUIS	Questionnaire for User Interface Satisfaction
RPC	Remote Procedure Call
SQL	Server Query Language
SUS	System Usability Scale
UML	Unified Modeling Language
URL	Uniform Resource Locator
WAR	Web application Archive
XML	Extensible Mark-up Language

ACKNOWLEDGEMENTS

I would like to thank my supervisors Prof. Thorkild Tylleskär and Jørn Klungsøyr, as well as the administration at the Centre for International Health at the University of Bergen. Their great support and guidance throughout the Masters program in all matters, from the most general to the most technical, made it possible for me to succeed. I would also like to thank the Research Council of Norway who funded the Open Mobile Electronic Vaccine Trials (OMEVAC) project which made my studies and stay in Norway possible. Last but not least, I would like to express my heartfelt gratitude to my parents who made continuous sacrifices so that I could reach this point, but passed away before completion of my studies.

1. INTRODUCTION

Researchers conducting studies related to health rely on regular data updates to judge progress and to make relevant decisions. Updates presented in the form of graphs can quickly present key study variables and relationships between them. While it is an advantage to use graphs to present regular updates, all graphs must be constructed. This can be done by using one of many programs (such as Microsoft Excel) to load a dataset and define the parameters of the graph. The resulting output can usually be copied to other documents and shared as needed. While this is a standard way of making and sharing graphs, it can be a labour-intensive process. In a large and complex study requiring frequent updates (perhaps one or more times per day) in the form of graphs, assistants may have to dedicate significant amounts of time in preparing and distributing the graphs. This can put a strain on the limited time and budget of the study.

In order to get frequent updates without taking much time away from assistants, researchers may desire a system that can automate some of the tasks associated with making and distributing the graphs. Such a system may give users the ability to:

1. Define what graphs need to be created by identifying the database in which the study data is stored, as well as the tables and column names to use for graph data series and labels. This would be done once and edited only if the graph type, data series, and/or labels needed to be changed.
2. Automatically create graphs by directly accessing the study database at an interval defined by the user. This would avoid the need to have assistants spend time to load datasets, define the parameters of the graphs, and then create them.

3. Make the graphs available for use in a standard image file format. Such a feature would make it easy to distribute and view the graphs, as well as to use them in other documents and programs without the need of special software.
4. Access an interactive website with a username and password from where the graphs can be viewed.
5. Access the interactive website described above from a mobile phone in addition to doing so from a computer. This would allow users to access the website from almost anywhere in the world at any time. This assumes that users have a suitable mobile phone which has internet access, web browsing (including to websites written in the Java programming language), and image file viewing capabilities.
6. Easily and quickly see all the available graphs by way of a simple user interface.
7. See the time and date in which each graph was created. This would allow for a measure of quality control. If the user set the interval to make new graph at one day, then a time-date stamp that is more than a day old would indicate an error.

Some of the capabilities described above were implemented in a Geographic Information System (GIS) web application developed in 2009 by the information technology staff at Interactive Research and Development (IRD), based in Karachi, Pakistan. The researcher was heavily involved in this effort. The software was developed to help researchers based at IRD and the Bloomberg School of Public Health at Johns Hopkins University in Baltimore, Maryland, USA track study progress and make relevant decisions. Five specific applications were developed to work with a web-based instance of Google Earth to bring a rich, web-based data visualization experience to the researchers. One of the five applications is a web application interface that serves as the entry point to the GIS. It provides the authentication (via a username and password) and framework to access the GIS and host an instance of Google Earth. This web application was not developed by the researcher and will not be discussed. The remaining four applications were developed by the researcher in 2008 and 2009 and are no longer in active use.

A brief description of this work is necessary to better realize the significance of the software development that was done for this thesis. The names of the applications and some project details are:

1. PATH EPI KML Maker – this application was developed to display specific data regarding Expanded Program on Immunization (EPI) centres in the Karachi Invasive Pneumococcal Disease Surveillance Project, a project which was done in collaboration with the Johns Hopkins Bloomberg School of Public Health. The first and last parts of the application name stems from the Program for Appropriate Technology in Health (PATH) which funded the project and Keyhole Mark-up Language (KML, explained below).
2. PATH GP KML Maker – this application was developed to display specific data regarding participating General Practitioner (GP) clinics in the Karachi Invasive Pneumococcal Disease Surveillance Project.
3. DOTS KML Maker – this application was developed to display specific data regarding individual patients enrolled in the IRD-Indus Hospital Directly Observed Treatment, Short-course (DOTS) program. It complements a specific implementation of the open-source electronic medical records system OpenMRS [1] for the IRD-Indus Hospital DOTS program.
4. MDR-TB KML Maker – this application was developed to display specific data regarding individual patients enrolled in the IRD-Indus Hospital Multidrug-Resistant

Tuberculosis (MDR-TB) program. It complements a specific implementation of OpenMRS MDR-TB edition for the IRD-Indus Hospital MDR-TB program.

The four applications performed all the necessary steps required to retrieve relevant project data from the database and process the data into specially structured files for display on the web-based instance of Google Earth. They would also overwrite the necessary files to provide updated information for display by the server. In doing so, the newly created files represented data from individual patients and health centres in the form of unique place-marks (icons) on a map. Clicking on the place-mark resulted in a small window opening up within the GIS with the data displayed inside. The format of the files displayed was the Keyhole Mark-up Language (KML), which follows a tag-based structure like the standard Extensible Mark-up Language (XML) files [2] and can be used to display GIS data in multiple programs. [3]

The four applications described above provided the benefit of allowing researchers to track the progress of multiple studies in an easy to navigate and visual format. The applications ran on a built-in scheduler that updated the information at pre-defined intervals. While a considerable amount of information was available, the screen was not covered-up with it. Instead, data were intuitively categorized by centre or patient locations (represented as place-marks), which were clearly visible on an interactive map. In the case of health centre place-marks, literally seeing the health centre along with its associated study data may have helped researchers to note emerging trends quickly. In the case of patient place-marks, researchers could zoom out of the interactive map so that the entire study area was visible. Then they could quickly see where patients were, what the patient types were, and note any areas with a higher concentration of patients. This functionality was accessible from any internet-enabled computer with a web browser that supports the free Google Earth plug-in.

This was particularly useful when multiple researchers were operating internationally through collaborative agreements.

While there are many benefits to the four applications described above, the applications themselves had several limitations. These need to be described to clearly see the improvements and changes that went into the new software. The limitations can be categorized as relating to retrieving data from the study database, and processing the information for output in the GIS.

The limitations with regards to data retrieval arose from the platform on which the applications were developed, they were script-based. All four applications were developed using AutoHotkey, which is not a conventional programming language. It is an open-source program for Windows that allows automating tasks by way of scripts written by users that conform to a unique syntax. Its basic capabilities include sending keystrokes, mouse movements and mouse clicks to Windows, as well as assigning keys or a combination of keystrokes to perform a task such as opening a file or running an application (hotkeys). Advanced capabilities include generating windows, forms, menus, renaming/moving/deleting files, as well as reading/writing/parsing text files. A completed script that directs AutoHotkey to perform such tasks can be compiled into a standard Windows executable file (*.exe file) so that it is a stand-alone application that does not require AutoHotkey to run. [4] The four applications were compiled AutoHotkey scripts.

There are three major limitations to retrieving data from the database via AutoHotkey. The first is that since all interaction with the database is through sending keystrokes to the database command line client window, a user must be logged-in to the computer, and that computer will be unusable while the application is running (except for background tasks). If, for example, a user opens a text editor while the application is running, then many of the keystrokes will be sent to the text editor before the loop gets to the point where the database command line client window is made active again. Not only will several queries be typed-out in the text editor, but those queries will not be executed. This means that the resulting Google Earth file will have missing information, and the amount of missing information could be substantial. The second limitation is that because keystrokes can appear in a text editor, sensitive information such as the database username and password may become visible (in practice the application was run on a computer in a secure location).

The third limitation to retrieving data from the database via AutoHotkey is that since queries are sent as keystrokes to the database command line client and the results are stored as files, the whole process is very slow. Databases can perform transactions very quickly, but if keystrokes need to be sent and files need to be written, then the process slows down dramatically. For instance, both PATH EPI KML Maker and PATH GP KML Maker were written with a hard-coded delay of between 190 ms and 200 ms when sending queries in the main loops used for retrieving data. Decreasing this value by much caused the database command line client window to lag behind the keystrokes, so that the ENTER keystroke at the end of the query was sent before the entire query appeared in the window, thus causing incomplete queries to be sent. So while the database command line client window may show a database transaction time of less than 10 ms, sending keystrokes makes the process much longer.

All four applications did not perform the same tasks to process the data for output by the GIS. The file(s) containing the data were accessed directly by the application and the contents were read into memory. PATH EPI KML Maker and PATH GP KML Maker read multiple files and data were inserted into a pre-defined template that was hard-coded into the script. This template followed the Google Earth KML file structure. DOTS KML Maker and

MDR-TB KMZ Maker read a single file that was created using a pre-defined Data Export task within the specific implementation of OpenMRS that they each complimented.

All four applications made decisions based on the values of specific data, but the amount of decisions made varied considerably. For example, both DOTS KML Maker and MDR-TB KMZ Maker selected a different value for the place-mark icon based on the gender of the patient. MDR-TB KMZ Maker went one step further and changed the colour of the icon as well, depending on the patient type. PATH EPI KML Maker made many decisions when constructing a stacked bar graph showing the number of visits by month with up to eight series corresponding to pre-defined categories of patient visits. In the end, a Uniform Resource Locator (URL, a network or internet address) string with the necessary values was constructed as per the requirements of Google Chart Tools. [5] This string was embedded in the Google Earth KML file so that when the file was displayed on the GIS, the parameters of the URL string were sent to the Google Chart Tools server and a resulting image of the chart was retrieved.

With regards to limitations of the four applications in processing the information for output, there were two. Firstly, the procedures relied extensively on script arguments that were specific to the exact task. This meant that every query, table, and chart was made using script that could only be used to retrieve and process data in the exact same manner. If a new field needed to be added to a table or a new series to a stacked bar chart, then the script had to be changed. If the same functionality was needed to display data from a new study, then a new script was required. Only the basic concepts on which the script was based could be copied, which meant that changes would have to be made to almost every line of the script. This would require significant time and effort to make and test.

Secondly, all graphing functionality was done externally via Google Chart Tools. This meant sending study data to a third party which is not part of the study in any way. While a degree of anonymity could be achieved by leaving out certain information, this could still be unacceptable as per the ethics guidelines if done in another study. Graphing in this manner also required users to follow Google's Terms of Service, which state that Google can discontinue the service, even immediately and without notice under certain circumstances. [5] Also, lack of internet connectivity would shut down all graphing capability.

Due to the limitations described earlier, there was a need for software that could perform similar tasks in a more generic way and not rely on third-party services. Also, there was a need to shift emphasis towards graphing in general and not on the inclusion of geographic data. New software solution that has more capabilities and fewer limitations had to be developed and evaluated. Researchers, especially those collaborating internationally with each other, must be able to access and review the field data of their choice. This must be possible regardless of where the researchers are and at what time they choose to review data. They must also be able to access data in considerable depth with relative ease, and the data must be presented in meaningful ways.

2. OBJECTIVES

The objectives of this study were as follows:

1. To develop prototype software with the following specifications:
    a. Platform independent (independent of operating system)
    b. Independent of external software
    c. More generic than the researcher's previous software
    d. Ability to use the openXdata database
    e. Automatic graph generation through simple user-supplied definitions (plain text and basic Server Query Language (SQL) commands)
    f. Accessibility of graphs through a web application
    g. Easy graph viewing through a smart phone
2. To evaluate the web application for user acceptability by using a modified version of the system usability scale (SUS).

3. METHODS AND DATA COLLECTION

3.1 SOFTWARE DEVELOPMENT METHODS

The Java programming language was used for this software development and several resources and examples were used for Java programming in general and for developing the specific functionalities of openXdata Mobile Charts (oXd MC). One of the most consulted has been the Java programming textbook by Deitel and Deitel. [6]

Details of how the software was developed and how various resources and examples were used are listed in this section. The first step was planning exactly what oXd MC was supposed to be able to perform and how users and implementers would interact with it. This required the construction of a use case diagram, figure 1.

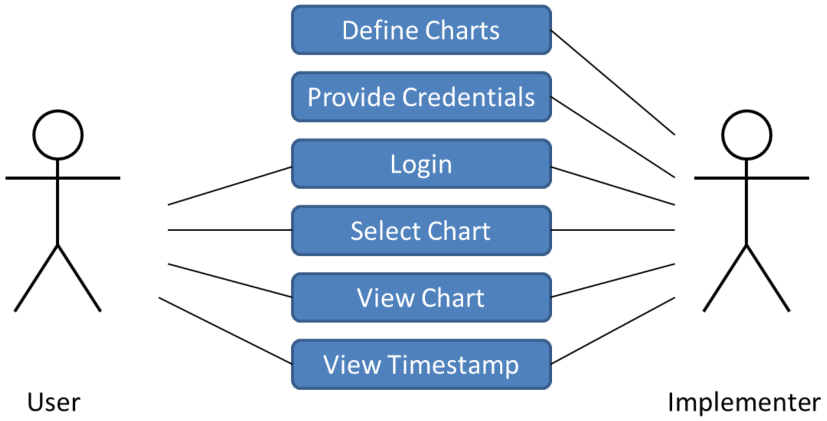


Figure 1: Use case diagram for the chart generation and display system from the user and implementer's perspectives.

The next step is usually the construction of class diagrams showing attributes, association, activity and states (state diagram). However, due to insufficient prior experience, it was difficult to know what classes would be required and how they would interact with each other. Instead, the researcher learned how to develop the different required functionalities in a learning-by-doing exercise by studying examples and explanations from the textbook and internet in an iterative process to produce the desired output in which the researcher was both the developer and user/tester. During this process, the researcher encountered many technical issues which did not have obvious solutions. These were addressed by studying proof-of-concepts in online resources, and changes were made based on those. The aim was to eventually combine all the functionalities required for generating charts into one project (with the resulting prototype being named oXd MC Chart Generator Component), and combine all the functionalities required for viewing charts online into another project (with the resulting prototype being named oXd MC Web Application Component). In some cases, class and package names were retained to aid the researcher in the process. Table 1 lists the software resources that were used. It is followed by a summary of the different functionalities required and what sources the researcher used in making them. It includes a description of what examples were supposed to do, what was learned from them, and what significant changes were made to adapt them to the objectives. The many issues and their exact solutions have been excluded. Technical details can be found in the code for the two projects that comprise oXd MC.

Table 1. Software/libraries used in the process

Software/libraries used in developing oXd MC	Version numbers
Java Development Kit	1.6.0_21
Eclipse Integrated Development Environment for Java Developers	1.2.0.20090619-0620
Google Plugin for Eclipse 3.5	1.3.3.v201006111302
Google Web Toolkit Software Development Kit	1.7.1
EXT GWT Java User Interface Component Library for Google Web Toolkit	2.2.0
JFreeChart Java Chart Library	1.0.13
MySQL Database Connector Library	5.1.13

The researcher selected JFreeChart as the Java-based chart library for creating charts/graphs and saving them in a picture format. The researcher then studied Google Web Toolkit and the remote procedure call (RPC) and made a web application that became the foundation for oXd MC Web Application Component. [7] The source code of the Contact List web application written by Masud Idris [8] was also studied for this purpose.

The researcher modified the Contact List application so that data would be read from a database. This was accomplished through published methods [6] and replicating the database connection/transaction examples. It was necessary for credentials to be read from a properties file and a solution was made based on examples in chapter 19 in [6]. A drop-down menu was added next. The researcher studied data models and how they are used in drop-down menus. The database was accessed to retrieve the following information (each from a separate database column as defined in the properties file): chart titles for populating the drop-down menu, timestamps (time and date labels) in which the chart was made, the file name for the associated chart image, and one more column for other information. This column was not used but accessing it was retained for possible future use. Upon making a selection, the associated chart image was displayed. This functionality was created and tested in a separate GWT web application project.

Extra features were added to make the user interface simple. The researcher added three Content Panels (aligned vertically) that display different information. The top one displays the drop-down menu, the middle one displays an image, and the bottom one displays a small amount of additional information associated with the selected menu item (intended for timestamps, but could be any string data so long as it fits in the space provided). The login screen was added at the end, and it was adapted from Addy Kapur's online tutorial. [9] The code presented in the tutorial was modified so that the username and password were checked against those stored in a database (in plain text form, which is a limitation that will be discussed later).

The researcher then followed the necessary steps required to build a Web application Archive (WAR) file [10] and deployed oXd MC Web Application Component in Apache Tomcat 6.0. The application was tested using the same steps that participants would follow during evaluation. When accessed through Mozilla Firefox 4.0, oXd MC Web Application Component worked as expected. It was accessed a second time using the Nokia 5230 phone and the oXd MC Web Application Component worked as expected. There were some cosmetic issues with the way oXd MC Web Application Component was rendered in the phone browser, but these were minor. Due to time constraints and the fact the oXd MC Web Application Component was working as expected, the researcher halted further development and debugging and moved on to the next phase.

As a quick test, the researcher tried accessing the chart image files by entering the chart image file name in the web browser (after the web application's address). This resulted in an error. No further testing was done to see if charts could be directly accessed. Also, work was started on a logout feature but halted in order to move on to testing and the evaluation phase. The researcher did check to see if it was possible to login, navigate away from the home page and then back to the home page. It was found that this was not possible. The login screen would be displayed instead. The development of oXd MC Chart Generator Component followed a similar path of prototyping, modifying, and adapting to meet objectives. It proved to be easier, as much of the knowledge and experience needed to develop it was gained in developing the oXd MC Web Application Component. For example, the functionality of reading a properties file and accessing/interacting with a database were quickly adapted from oXd MC Web Application Component. However, there were some major differences in intended functions that required further study. The first was how to make a program that would perform a task repeatedly according to a user-defined interval. This functionality was adapted from the online tutorial “Using the Timer and TimerTask Classes” [11]. The next step was studying and adding a chart generation capability. Two online tutorials with several examples of how to create and save charts were studied. [12, 13] The researcher made modifications so that user-supplied parameters in a database were used to create a chart. The most important feature that had to be implemented in oXd MC Chart Generator Component was support for SQL in selecting chart data and simple, pipe-delimited (the "|" character) labels to use as chart labels. The next step was adding support for creating multiple charts at once and to repeat the process based on the timer. To keep the use of hard-coded data at a minimum, the researcher added support for using values from a properties file for timer, database connection and access tasks. This included support for user-defined labels for the different chart types (for example, any label to correspond to “box and whisker plot”). The researcher tested the features by way of sample data that was entered through the openXdata user interface after creating sample studies and forms. For example, testing the box and whisker plot involved creating a sample anthropometry study and entering 103 length/height-for-age forms.

The researcher constructed activity and class diagrams after development. The following five diagrams show what activities oXd MC performs as a system and how it does so with respect to programming. Figure 2 shows how oXd MC Chart Generator Component schedules and executes a task, while figure 3 shows details of how the task is accomplished. Figure 4 shows how oXd MC Web Application Component shows charts to users. Figure 5 is a technical diagram showing the different components of oXd MC Chart Generator Component and their relationship to one another, and figure 6 is a similar diagram for oXd MC Web Application Component. All diagrams have been made according to details in chapter 5 of [6].

Figure 2 shows the activities carried out by the scheduler system of oXd MC Chart Generator Component. The system does the following: It loads the properties file or displays the appropriate error message if it is inaccessible. It then displays details on delay, repeat interval, and total number of iterations. After this, it schedules the task as per the delay and interval and then creates the timestamp. Next, it runs the task by connecting to the database, creating charts, and saving them. It will display the appropriate error message if the database is inaccessible. Then it will decrease the number of times to repeat the task by one and display the number of remaining iterations along with the timestamp. If the number of remaining iterations is greater than zero, then it will repeat the tasks starting from the step of creating the timestamp. When the number of remaining iterations is zero, it will display a message that the cycle is complete.

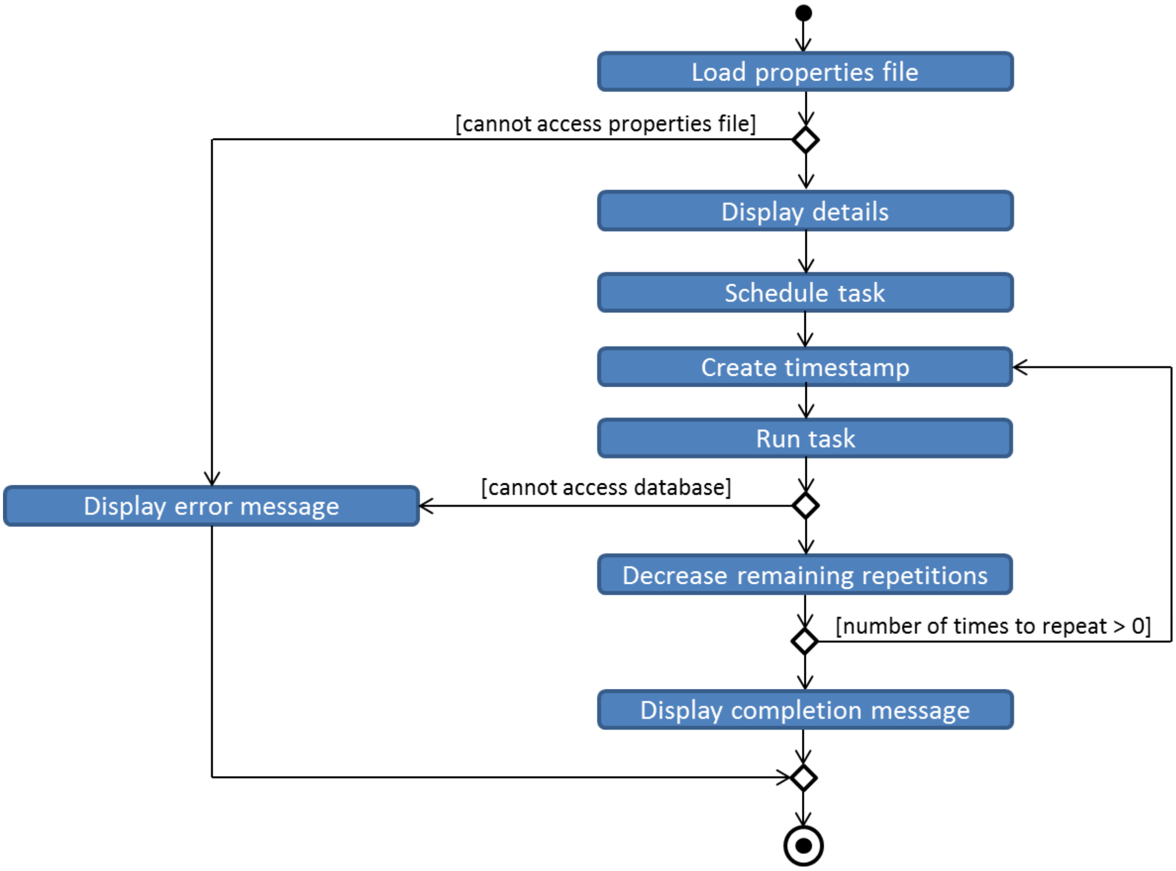

Load properties file
[cannot access properties file]
Display details
Schedule task
Create timestamp
Run task
Display error message
[cannot access database]
Decrease remaining repetitions
[number of times to repeat > 0]
Display completion message

Figure 2: Activity diagram for scheduler system of oXd MC Chart Generator Component.

Figure 3 shows the activities carried out by the chart generation system of oXd MC Chart Generator Component. The system does the following: It loads the properties file or displays the appropriate error message if it is inaccessible. It then connects to the database or display the appropriate error message if it is inaccessible. After that, it reads the chart definitions table in the database. If the chart definition is for a bar chart, box and whisker plot, line chart or pie chart, then the system will create and save the chart or display the appropriate error message if it cannot be saved. Then, the system will update the database with the timestamp, or display the appropriate error message if the database is inaccessible. Finally, it will repeat the process of creating more charts if the number of remaining charts is greater than zero.

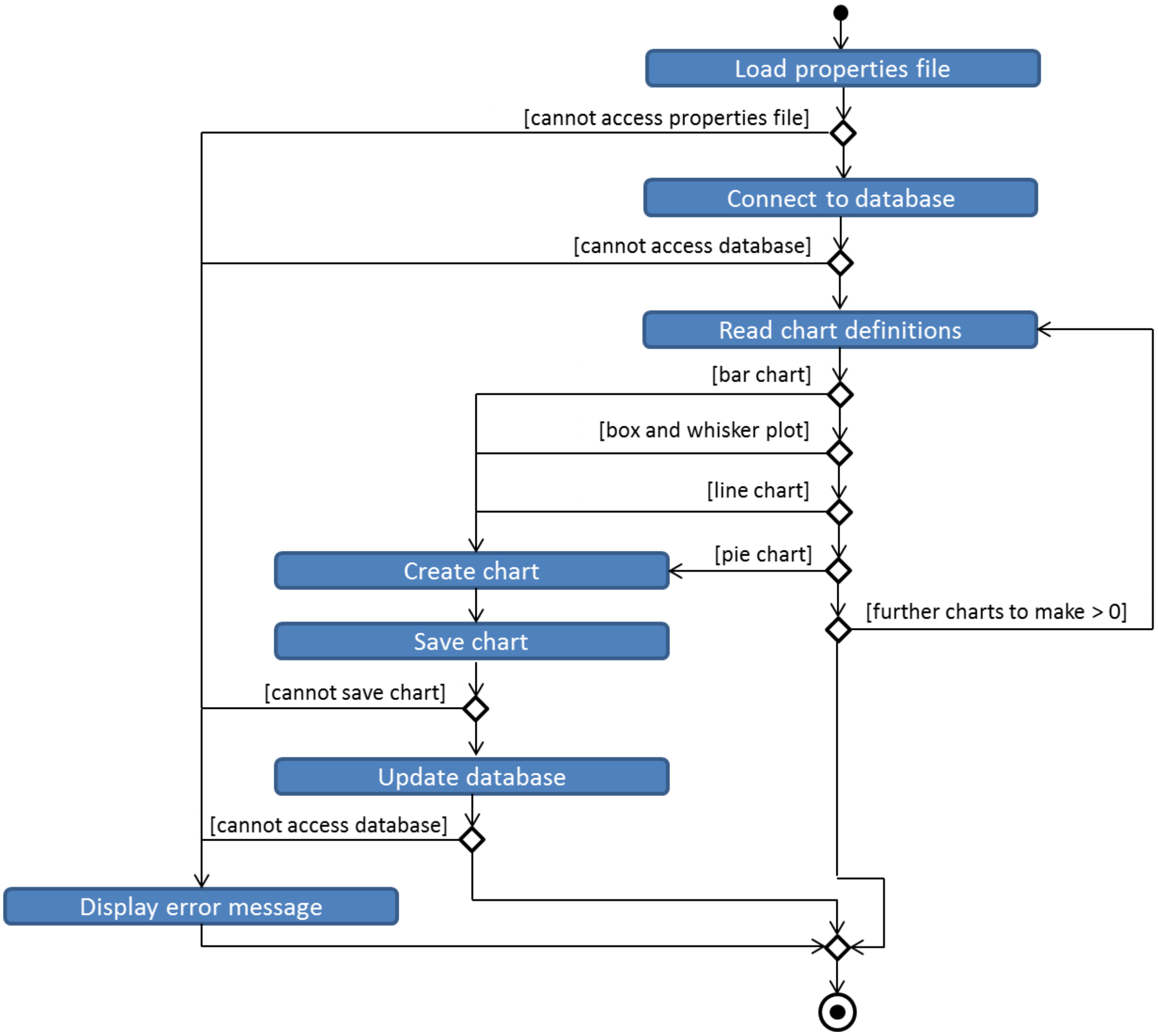


Figure 3: Activity diagram for chart generation system of oXd MC Chart Generator Component.

Figure 4 shows the activities carried out by oXd MC Web Application Component. The web application does the following: It displays the login screen and gets the username and password from the user. It then loads the properties file or displays the appropriate error message if it is inaccessible. Next, it connects to the database or displays the appropriate error message if it is inaccessible or if the username or password is incorrect. After that, it loads the properties file again or displays the appropriate error message if it is inaccessible, and then it connects to the database or displays the appropriate error message if it is inaccessible. In the next step it retrieves the data associated with the chart selection menu and displays the main screen. If the user selects a chart, then that chart along with its timestamp is displayed on the main screen. The user may repeat selecting and viewing charts.

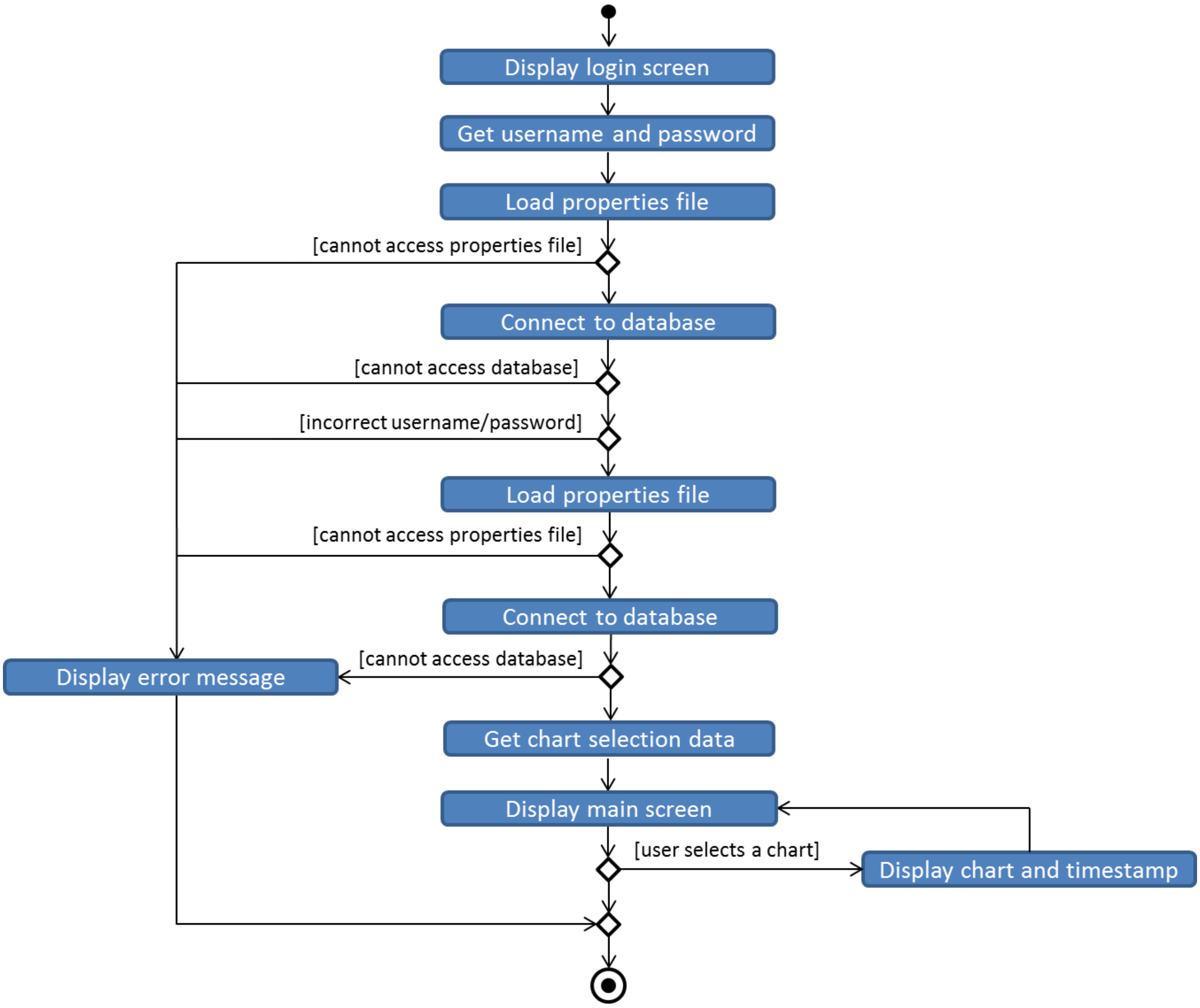


Figure 4: Activity diagram for oXd MC Web Application Component.

Figure 5 is a class diagram for oXd MC Chart Generator Component. It shows the various components (classes) and their relationship to each other. Figure 6 is a similar class diagram for oXd MC Web Application Component. Both diagrams have been constructed using Unified Modeling Language (UML) conventions.

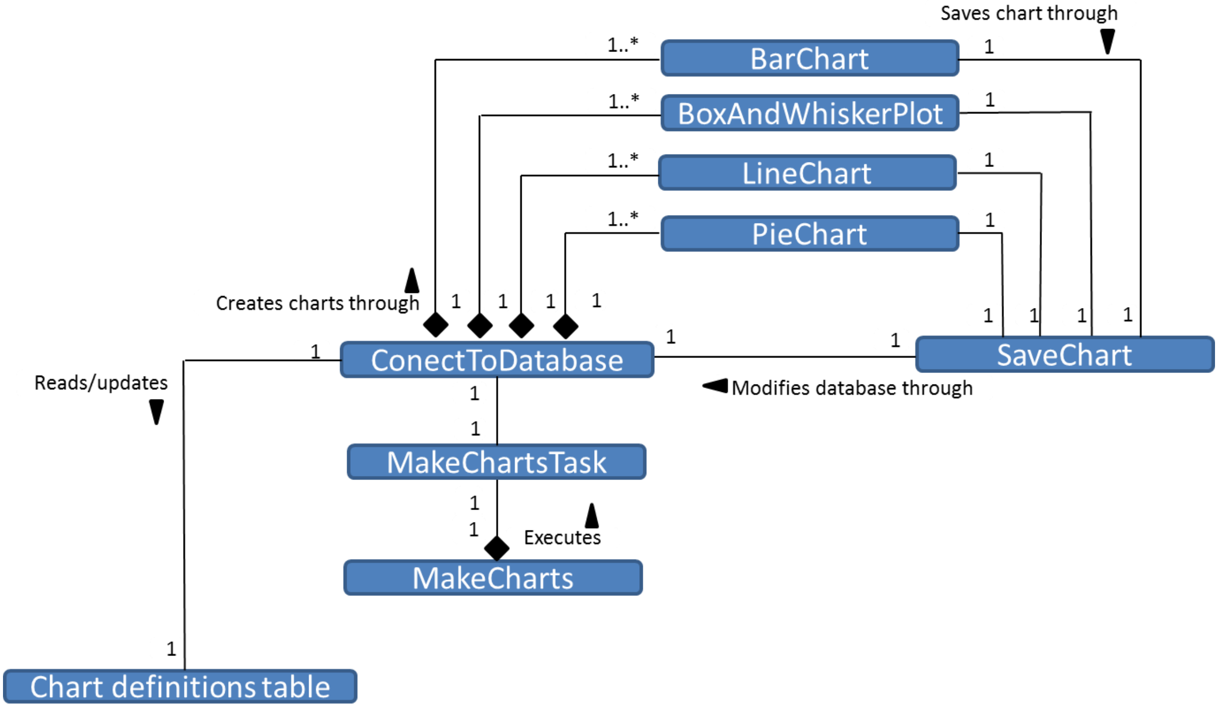


Figure 5: Class diagram for oXd MC Chart Generator Component.

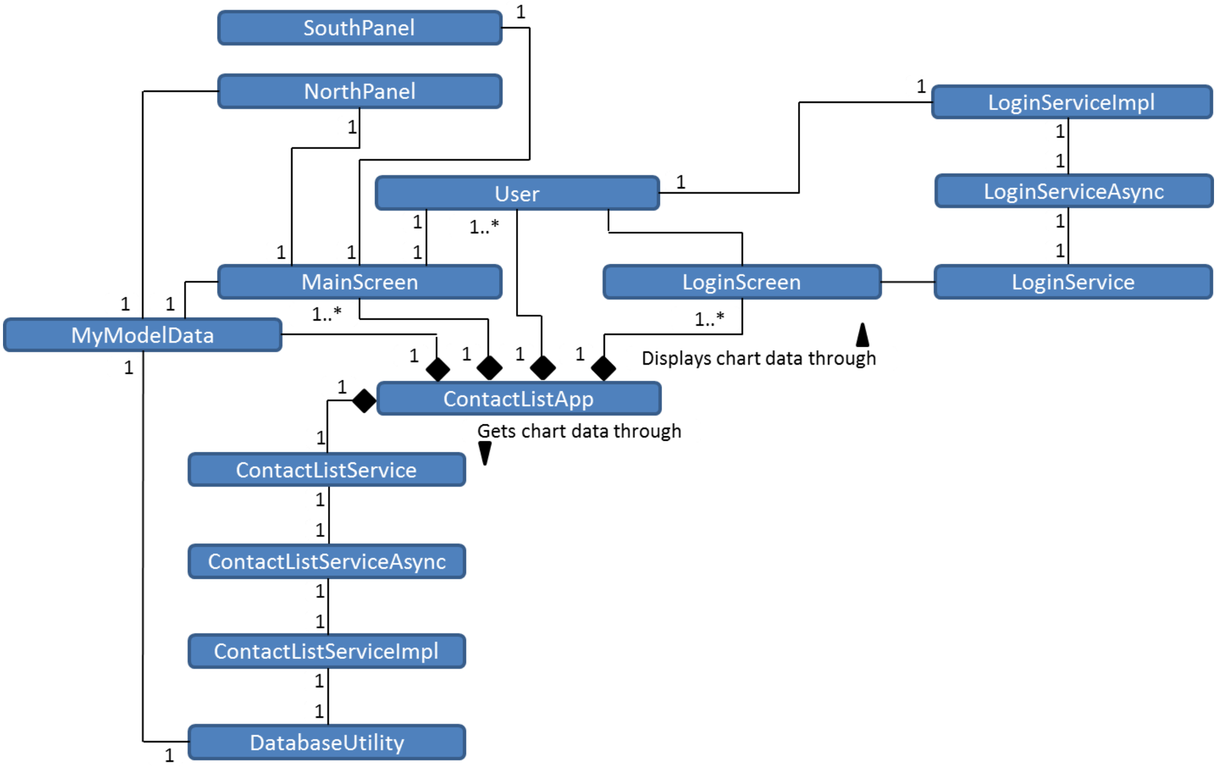


Figure 6: Class diagram for oXd MC Web Application Component.

3.2 DATA COLLECTION METHODS

For the purpose of the survey, an informed consent form was needed. Since the survey was anonymous without any personal or health-related questions and participants would only be asked to provide anonymous feedback on the usability of software in a multiple choice format, a simple informed consent form was considered. While this would have been acceptable in a pure computer science-related study, there was concern that it may not appear thorough enough under the rigorous ethics guidelines expected of health studies. For this reason, the researcher consulted the *Informed Consent Form Template for Qualitative Studies* published by the Research Ethics Review Committee of the World Health Organization. [14] This format was followed to ensure consistency with other surveys conducted at the Centre for International Health.

3.2.1 QUESTIONNAIRE SELECTION

A specific questionnaire was needed to collect data, and this would have to be one that was tested and accepted in the software development industry. The researcher consulted the paper *A Comparison of Questionnaires for Assessing Website Usability* [15] to learn which kind of questionnaire would be most suitable in evaluating oXd MC Web Application Component. The following five questionnaires were compared in the paper: [15]

1. System Usability Scale (SUS)
2. Questionnaire for User Interface Satisfaction (QUIS)
3. Computer System Usability Questionnaire (CSUQ)
4. Words (adapted from Microsoft’s Product Reaction Cards)
5. The questionnaire used by the authors (Tullis and Stetson) for several years in usability tests of websites.

Each questionnaire was selected based on its suitability to assess the usability of websites. The study involved 123 participants and each participant was randomly assigned to one of the five questionnaires, and this “yielded between 19 and 28 participants for each questionnaire.” [15] All participants performed two tasks on each of two financial information websites which were referred to as Site 1 and Site 2. [15] The authors state that:

> “The order of presentation of the two sites was randomized so that approximately half of the participants received Site 1 first and half received Site 2 first. After completing (or at least attempting) the two tasks on a site, the user was presented with the questionnaire for their randomly selected condition. Thus, each user completed the same questionnaire for the two sites.” [15]

The results were that “All five questionnaires showed that Site 1 was significantly preferred over Site 2.” [15] The authors then proceeded to the main objective, which was:

> “to determine what the results would have been at different sample sizes from 6 to 14. At a sample size of 6, only 30-40% of the samples would have identified that one of the sites was significantly preferred. Most of the data reach an apparent asymptote at a sample size of 12, where two of the questionnaires (SUS and CSUQ) yielded the same conclusion as the full dataset at least 90% of the time.” [15]

The following graph illustrates the results, and the caption describes it in the words of the authors. [15] For more details on the results for each questionnaire, the overall results and the statistical methods used, the results section of the paper should be consulted. [15]

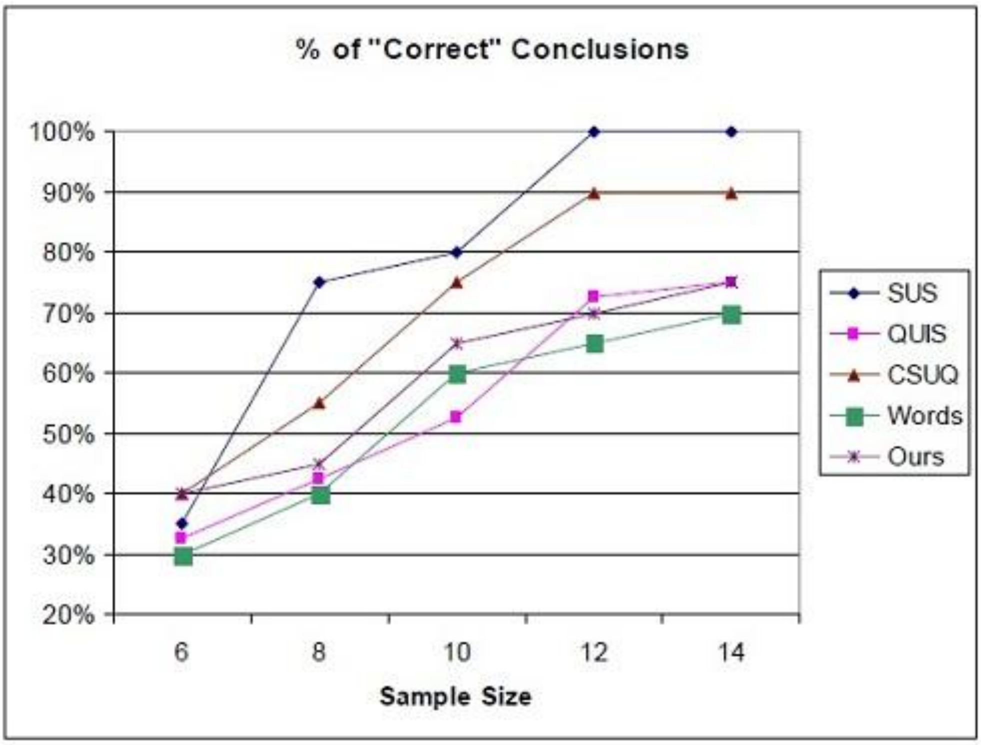


Figure 7: Data based on t-tests of random sub-samples of various sizes. Twenty sub-samples were taken at each sample size for each site and each questionnaire. What is plotted is the percentage of those 20 tests that yielded the same conclusion as the analysis of the full dataset (that Site 1 was significantly preferred over Site 2).

As one can see with a sample size of 12, the “apparent asymptote” for SUS was actually 100%. This meant that if the SUS were used with a sample size of 12 participants, the researcher could expect to get the most accurate results amongst the five questionnaires. For this reason, SUS was selected as the questionnaire, and the sample size was set at 12 participants.

The next step was to research the SUS so that a proper survey could be conducted and the scores presented in a form that was easy to understand. The researcher consulted the papers *SUS: a "quick and dirty" usability scale* [16] and *Determining What Individual SUS Scores Mean - Adding an Adjective Rating Scale*. [17] In the first of these two papers, the author describes the SUS as a 10-item scale in which respondents are given a statement and five options to choose from. The options range from "Strongly disagree" as the first choice, followed by 3 choices with no labels, and finally "Strongly agree" as the last choice. [16] The items are worded such that five are positive and five are negative, and placed in an alternating manner so that “the respondent has to read each statement and make an effort to think whether they agree or disagree with it.” [16] Completing the questionnaire “yields a single number [ranging from 0 to 100] representing a composite measure of the overall usability of the system being studied, [16]” but one should note that “scores for individual items are not meaningful on their own. [16]” Further, the author states that the SUS:

> “is generally used after the respondent has had an opportunity to use the system being evaluated, but before any debriefing or discussion takes place. Respondents should be asked to record their immediate response to each item, rather than thinking about items for a long time.” [16]

However, there is a conflict between the author's description and the requirements of the study. Firstly, informed consent is required, and this means clearly describing the study and the complete scope of participation. Secondly, any candidate that is approached must be able to ask questions, and those who choose to participate must be able to ask questions during the course of assessment. This means that some discussion will take place before the participant has had an opportunity to use the system. Further, the author states that “the usability of any tool or system has to be viewed in terms of the context in which it is used, and its appropriateness to that context.” [16] This means that participants would have to be asked to evaluate the usability of oXd MC WAC in the context of a website that is meant to provide graphical reporting based on study data in addition to discussion that ensures informed consent from the participant. In short, participants would have a good idea of what the software is supposed to do and what kind of output to expect, but no information on how user-friendly it is.

Four modifications were made to the SUS according to the directions provided by Bangor et al. [17] The first was to print directions on the questionnaire itself and not on any other page or to communicate them verbally. While Brooke gives clear directions for participants to follow in the first paper, these are not included in the actual SUS template that is provided. [16] To ensure that all participants got exactly the same instructions for completing the survey, the researcher printed the same directions on the questionnaire as that printed by the authors in the template, but the word “don't” was changed to “do not” (which is a minor change that will not regarded as one of the four modifications) :

> “Please check the box that reflects your immediate response to each statement. Don’t think too long about each statement. Make sure you respond to every statement. If you don’t know how to respond, simply check box “ [17]

The second modification was changing the word “system” in the original questionnaire to “product.” The third modification was changing the word “cumbersome” in the statement for item 8 to “awkward.” These two changes reflect those made by the authors, which are based on feedback from respondents and have been used in approximately 3,500 surveys in 273 studies. [17] The fourth modification was the addition of one more item. This was done because as the authors state, “While a 100-point scale is intuitive in many respects and allows for relative judgments, information describing how the numeric score translates into an absolute judgment of usability is not known.” [17] The authors addressed this issue by adding an eleventh item to the questionnaire. This item is similar to the other items in that it has seven response options but different in that all options are labelled with an adjective, as shown below:

11. Overall, I would rate the user-friendliness of this product as:

□	□	□	□	□	□	□
Worst	Awful	Poor	OK	Good	Excellent	Best
Imaginable						Imaginable

The authors added this to 964 SUS-based surveys [17] and found that the scores “correlate extremely well with the SUS scores (r=0.822)” and that the “addition of the adjective rating scale to the SUS may help practitioners interpret individual SUS scores and aid in explaining the results to non-human factors professionals.” [17] The following chart illustrates the relationship between adjective ratings and SUS scores: [17]

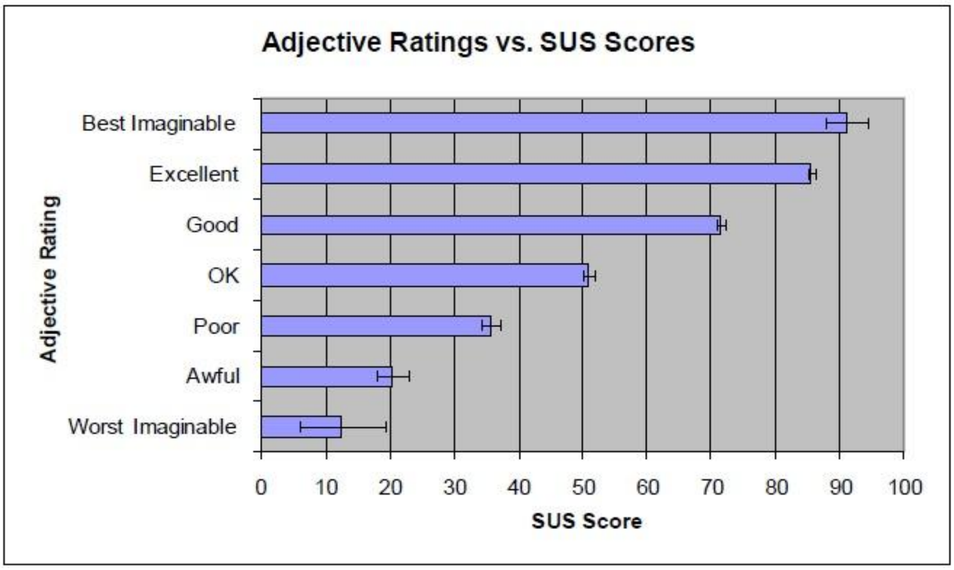


Figure 8: Graph showing the relationship between adjective ratings and numerical SUS scores.

The following figure shows four different rating scales plotted together in such a way that it is easy to see the relations across the scales: [17]

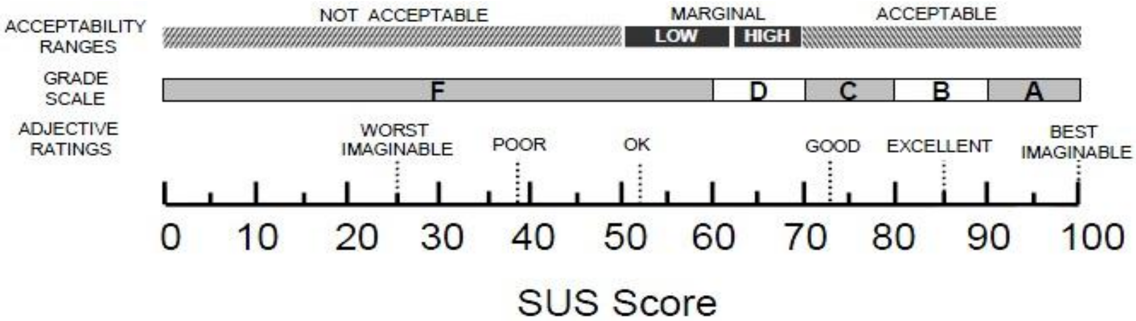


Figure 9: Chart showing three usability rating scales plotted with respect to SUS scores.

The “Acceptability Ranges” shown at the top of the figure is a scale that was proposed in a previous paper [18] with the aim of helping “practitioners determine if a given SUS score indicated an acceptable interface or not.” [17] The “Grade Scale” shown next is the common scale used in American schools, where scores are associated with letters. The authors state that they have “found that a useful analogue to convey a study's mean SUS score to others involved in the product development process has been the traditional school grading scale (i.e., 90-100 = A, 80-89 = B, etc.” [17] The “Adjective Ratings” and “SUS Score” refer to scales described earlier. The SUS as modified by Bangor et al was used by the researcher to assess the usability of oXd MC WAC and to interpret the results in a manner that is easy to understand.

3.2.2 PARTICIPANT SELECTION

Suitable candidates for the evaluation of oXd MC Web Application Component were researchers in the field of international health. To quickly and easily find a pool of candidates, the researcher searched at the Centre for International Health at the University of Bergen. An email call for candidates failed and the potential candidates were therefore approached personally by the researcher among participants in the on-going elective course (Globalization and Health), usually either Masters or PhD candidates. It was assumed that such candidates would occupy middle and senior-level positions in a study. All twelve candidates who were approached agreed to participate and all completed the survey. The candidates were briefly informed about the research. They were asked if they would be interested in participating in a 20-minute survey in which they would use oXd MC Web Application Component and answer questions regarding acceptability. Those who expressed interest were given a copy of the informed consent form with the relevant details.

With those candidates who agreed to participate after having read the informed consent form, a mutually agreeable time and meeting point was set. On the next meeting, they were asked to sign two copies of the informed consent form, with one copy to be kept by the researcher and one copy to be kept by the candidate.

3.2.3 SURVEY CONDITIONS

The survey was conducted in the researcher's office at the Centre for International Health at the University of Bergen in order to control the environment. In addition to keeping the location constant, other steps were taken to limit differences for participants. All participants were given the same set of written instructions for using oXd MC Web Application Component. All participants used the same Nokia 5230 mobile phone when following the instructions. The researcher made sure the mobile phone was working properly and accessed oXd MC Web Application Component. This was checked before each interview. The researcher handed the phone to participants with the login screen of oXd MC Web Application Component loaded on the phone. The researcher sat next to participants while they read and followed the instructions. They were told to feel free to ask questions if anything was unclear. After participants finished using the software as per the instructions, they were told that the researcher would step outside the office while they completed the questionnaire. When they were done, they could call out to the researcher or step outside the office. Nothing was said to make participants feel that the researcher had to re-enter the office before the survey could be considered complete.

4. RESULTS

4.1 RESULTS OF SOFTWARE DEVELOPMENT

4.1.1 DESCRIPTION AND USAGE

The resulting prototype, oXd MC, has two components. The first component is a graph generator. This is supposed to be installed on a computer of the user’s choosing. Some requirements must be met, such as that the computer must be able to run a Java program, the program must be able to access the desired database, the desired graphs must be defined (in terms of what data to use, titles, and labels), and a location must be defined to save the graphs. The second component is a web application, which is also supposed to be installed on a computer of the user's choosing. As with the first component, some requirements must be met, such as that the computer must be able to host a Java web application, and the location of where the graphs are saved must be defined. All the functions of oXd MC occur on the computer(s) of the user's choosing, so no data is sent to a third party. Once it is installed, most interaction with oXd MC is through the web application, which has been designed to be accessed through a mobile phone. Listed below are details of how the prototype is implemented and what it does.

The following steps must be followed to install and use oXd MC Chart Generator Component. It is assumed that the implementer has knowledge working with databases and Java programs:

1. Find and open the file “OXD_MC_DB.properties” using a text editor. This file contains the data necessary for oXd MC Chart Generator Component to function, and these are referred to as properties. It contains four types of properties:
    a. Properties needed to set the scheduler system and create the chart files.
    b. Properties to connect to the database.

    c. Properties needed to find information in the database on how to make charts.

2. All properties must be changed to reflect the settings that are in use. Only the value of properties must be changed (the data that is to the right of the equals sign) and not the label of the properties (the data that is to the left of the equals sign). It is not recommended to change the following properties: "chart_file_type", "chart_width" and "chart_height".
3. Changes will most likely have to be made to the database so that oXd MC Chart Generator Component can use it. This may involve adding new tables, columns and data. Details are listed with respect to the labels of properties:

The following properties are for getting data and saving charts. Values from the researcher’s test database are shown:

a. chart_table_name=contact_list2

b. chart_id_column_name=chart_id

c. chart_titles_column_name=graphTitles

d. chart_file_names_column_name=graphFileNames

e. chart_files_date_time_created_column_name=graphFilesDateTimeCreated

f. chart_other_info_column_name=graphsOtherInfo

g. chart_type_column_name=graphType

h. graph_data_query_column_name=graphDataQuery

i. chart_files_date_time_creation_prefix=Created:

j. path_for_saving_charts=C:\\Owais\\openXdata\\Tomcat6.0\\webapps\\oxdmcl\\

The values of the first eight properties must be the same as the table name and column names that contain chart information. The ninth property is for the prefix that is used in the footer section, just before the timestamp. The tenth property must be changed to reflect the path where oXd MC Web Application Component is installed as a web application (see below for more information).

The following properties are to identify different chart types in the database. Default values are shown, but they must be the same as the value decided by the user and the usage must be consistent:

k. bar_chart=bar chart

l. box_and_whisker_plot=box and whisker plot

m. line_chart=line chart

n. pie_chart_label=pie chart

Properties for a time series chart and scatter plot are also included as work on these chart types was started. They are not fully supported so the properties are not listed.

The following steps must be followed to install and use oXd MC Web Application Component. It is assumed that the implementer has knowledge working with databases and web applications:

1. Access the management utility of whatever web application server is in use.
2. Locate and deploy the file “oxdmcl.war”. If the web application server supports deploying files by placing them in a specific folder, then this file can be placed there. When using the Apache Tomcat web application server, the folder is called “webapps”.
3. Once the file has been deployed, it should have its own folder (called “oxdmcl”) within the web application server folder. Navigate to this folder and then to the “classes” folder. When using Apache Tomcat, the path could look like: C:\Tomcat6.0\webapps\oxdmcl\WEB-INF\classes

4. Find and open the file “OXDMC_SETTINGS.properties” using a text editor. It contains three types of properties:
    a. Properties to connect to the database.
    b. Properties needed to authenticate users against those listed in the database.
    c. Properties needed to find information in the database related to generated charts.
5. All properties must be changed to reflect the settings for accessing the intended database. Only the value of properties must be changed and not the label of the properties. Some changes may have to be made to the database (adding new tables/columns/data).
6. The oXd MC Web Application Component can be accessed by using the URL to the web application server followed by “oxdmcl” for example: http://localhost:8585/openXdata/oxdmcl

Following the steps above will install oXd MC Chart Generator Component and oXd MC Web Application Component, but the software will not create and display charts without sufficient data for defining and populating charts. Data can be entered into a MySQL database in different ways, and one way is to use an external program. The researcher used openXdata, which is a web application that allows users to easily perform highly organized data collection on computers and mobile devices. Users can create studies and design forms using a GUI. These forms can be very versatile and support the submission of images, audio, video and GPS coordinates as responses. They also support logic for validation, skips and branches based on responses. The forms can be filled-out via computer and a variety of mobile electronic devices including many inexpensive mobile phones. Forms can be submitted to the openXdata server in a delayed manner, allowing data collection in areas with not network access and later submission in an area with network access. After submission, data can be viewed, managed and exported.

The researcher used the export feature to get data ready for use by oXd MC Chart Generator Component. openXdata automatically creates a new database table for each form that is exported. It is a simple table with all form instances listed as rows and their responses listed as columns. The researcher did the following to define a box and whisker plot using data from a sample anthropometry study comprised of 103 length/height-for-age forms:

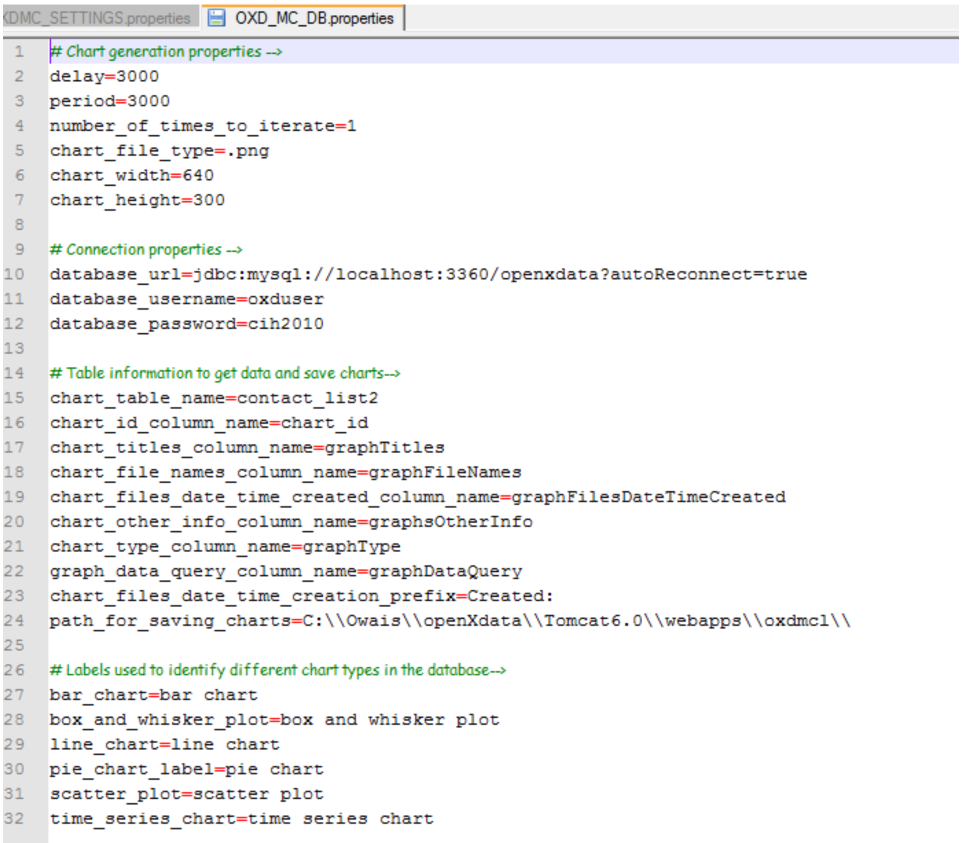
1. In the column for chart titles, the desired title was entered (“Length/Height for Age for Male Study Subjects, 20-24 Months”). Nothing was entered for the chart ID column, as that column was set to automatically to have the row number entered.
2. In the column for chart file names, the desired chart file name, as well as the path to the images folder of oXd MC Web Application Component was entered (“images/BoxAndWhiskerPlot.png”).
3. Nothing was entered in the column for the timestamp, as that information is entered by oXd MC Chart Generator Component after the chart has been successfully created.
4. In the column for other information, the desired titles for the value and series axes were entered in a pipe-delimited format. A third null parameter was added as this parameter is used by the pie chart but not for the box and whisker plot (“Age in Months|Length / Height (cm)|null”)
5. Query parameters were entered in the column for chart data query in a pipe-delimited format (“age_in_months|null|length_or_height_in_cm| lh_for_age_ex_lh_for_age_form1_v1|ORDER BY age_in_months ASC”). These parameters correspond to column keys, query parameter for row keys (not used in the

box and whisker plot), row keys, table name for the data and the ordering option for unique column keys (entered as a query parameter).

6. The appropriate label was entered in the column for graph type. It was the same as that defined in the properties file ("box and whisker plot"), but the case of letters is not important. The resulting chart is shown in figure 17.

4.1.2 SCREEN SHOTS OF PROTOTYPE AND REQUIRED DATA

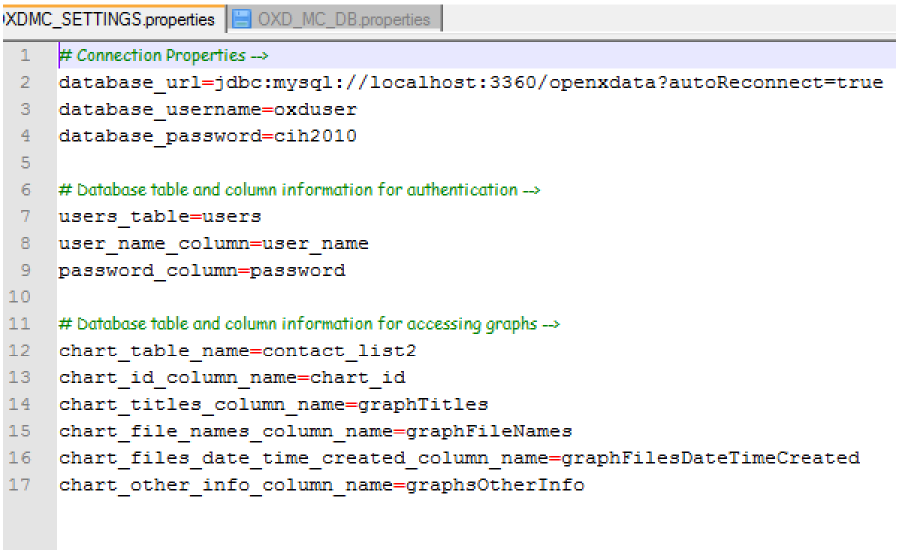
The following 11 figures are screen shots. Figure 10 shows the contents of the properties file needed to run oXd MC Chart Generator Component. There are four categories of properties: general, connection, database table information and labels for chart types. As stated in the description and usage section, all properties must be changed to reflect settings that are in use. Only the value of properties must be changed and not the labels.



```
# Chart generation properties -->
delay=3000
period=3000
number_of_times_to_iterate=1
chart_file_type=.png
chart_width=640
chart_height=300

# Connection properties -->
database_url=jdbc:mysql://localhost:3360/openxdata?autoReconnect=true
database_username=oxduser
database_password=cih2010

# Table information to get data and save charts-->
chart_table_name=contact_list2
chart_id_column_name=chart_id
chart_titles_column_name=graphTitles
chart_file_names_column_name=graphFileNames
chart_files_date_time_created_column_name=graphFilesDateTimeCreated
chart_other_info_column_name=graphsOtherInfo
chart_type_column_name=graphType
graph_data_query_column_name=graphDataQuery
chart_files_date_time_creation_prefix=Created:
path_for_saving_charts=C:\\Owais\\openXdata\\Tomcat6.0\\webapps\\oxdmcl\\

# Labels used to identify different chart types in the database-->
bar_chart=bar chart
box_and_whisker_plot=box and whisker plot
line_chart=line chart
pie_chart_label=pie chart
scatter_plot=scatter plot
time_series_chart=time series chart
```

Figure 10: Screen shot of the properties file for oXd MC Chart Generator Component.

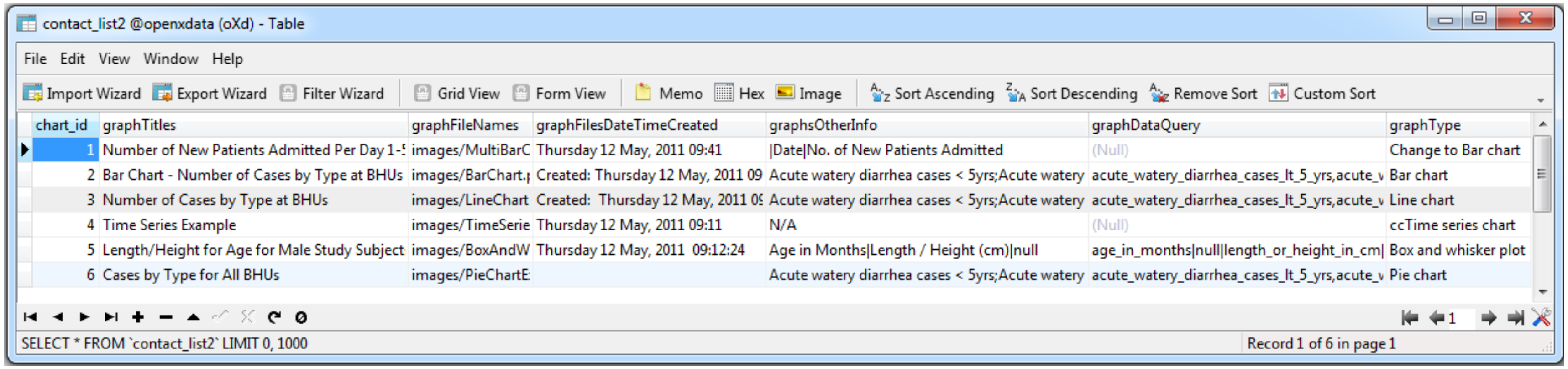
Figure 11 shows the contents of the properties file needed to run oXd MC Web Application Component. There are three categories of properties: connection, authentication for login to the system, and information for accessing chart definitions. As stated in the description and usage section, all properties must be changed to reflect the settings that are in use. Only the value of properties must be changed and not the label of the properties.



```
# Connection Properties -->
database_url=jdbc:mysql://localhost:3360/openxdata?autoReconnect=true
database_username=oxduser
database_password=cih2010

# Database table and column information for authentication -->
users_table=users
user_name_column=user_name
password_column=password

# Database table and column information for accessing graphs -->
chart_table_name=contact_list2
chart_id_column_name=chart_id
chart_titles_column_name=graphTitles
chart_file_names_column_name=graphFileNames
chart_files_date_time_created_column_name=graphFilesDateTimeCreated
chart_other_info_column_name=graphsOtherInfo
```

Figure 11: Screen shot of the properties file for oXd MC Web Application Component.

Figure 12 shows some of the data needed to create chart definitions that oXd MC Chart Generator Component can use. It shows the contents of the chart definition table, as seen through the Navicat database administration software. The upper-left corner shows the name of the chart definition table (“contact_list2”), which is the same as what is shown in figure 11 as the “chart_table_name” property. Five of the table columns also have the same names as those listed in figure 11 under the third category of properties.



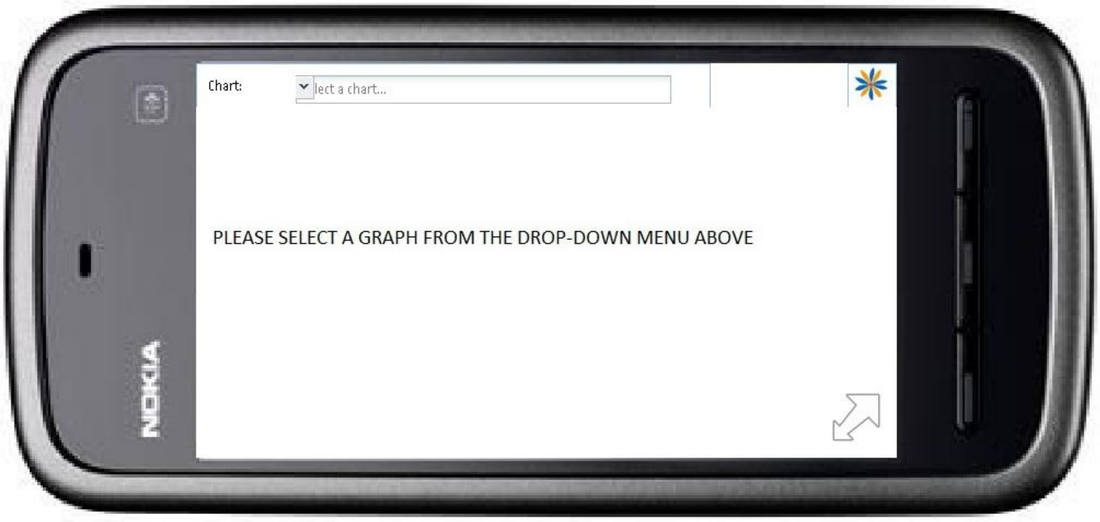
chart_id	graphTitles	graphFileNames	graphFilesDateTimeCreated	graphsOtherInfo	graphDataQuery	graphType
1	Number of New Patients Admitted Per Day 1-	images/MultiBarC	Thursday 12 May, 2011 09:41	|Date|No. of New Patients Admitted	(Null)	Change to Bar chart
2	Bar Chart - Number of Cases by Type at BHUs	images/BarChart.	Created: Thursday 12 May, 2011 09	Acute watery diarrhea cases < 5yrs;Acute watery	acute_watery_diarrhea_cases_lt_5_yrs,acute_	Bar chart
3	Number of Cases by Type at BHUs	images/LineChart	Created: Thursday 12 May, 2011 0	Acute watery diarrhea cases < 5yrs;Acute watery	acute_watery_diarrhea_cases_lt_5_yrs,acute_	Line chart
4	Time Series Example	images/TimeSerie	Thursday 12 May, 2011 09:11	N/A	(Null)	ccTime series chart
5	Length/Height for Age for Male Study Subject	images/BoxAndW	Thursday 12 May, 2011 09:12:24	Age in Months|Length / Height (cm)|null	age_in_months|null|length_or_height_in_cm|	Box and whisker plot
6	Cases by Type for All BHUs	images/PieChartE		Acute watery diarrhea cases < 5yrs;Acute watery	acute_watery_diarrhea_cases_lt_5_yrs,acute_	Pie chart

SELECT * FROM `contact_list2` LIMIT 0, 1000

Record 1 of 6 in page 1

Figure 12: Screen shot of the MySQL database table for chart definitions as required by oXd MC Chart Generator Component The image has been rotated to fit on the page.

Figures 13-19 show oXd MC Web Application Component being accessed via a mobile phone. All screen shots have been superimposed on a picture of the Nokia 5230 mobile phone so the reader can visualize oXd MC Web Application Component in the correct context. Figure 13 shows the login page, where the user can enter a username and password through the touch-screen keyboard that is supported by the phone. Figure 14 shows the home page, which appears only when the username and password have been successfully authenticated against those in the database.



PLEASE SELECT A GRAPH FROM THE DROP-DOWN MENU ABOVE

Figure 13: Screen shot of the login page.

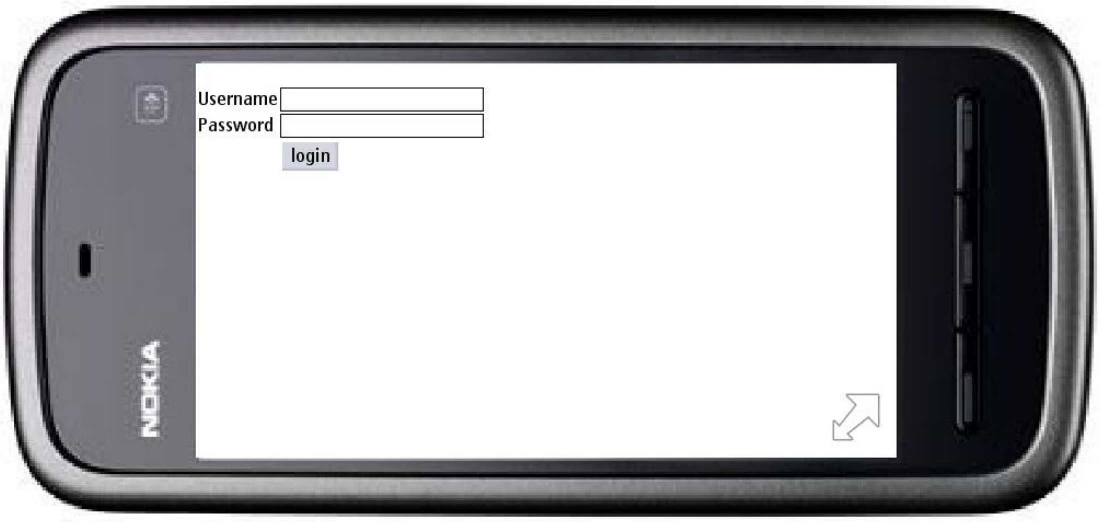


Figure 14: Screen shot of the home page.

Figure 15 shows the chart selection menu which is located along the top of the home page. The menu expands to show its contents when the user selects it. Each menu item corresponds to a different row in the chart definition table. The name that appears is the same as what is listed in the column corresponding to the “chart_titles_column_name” property.

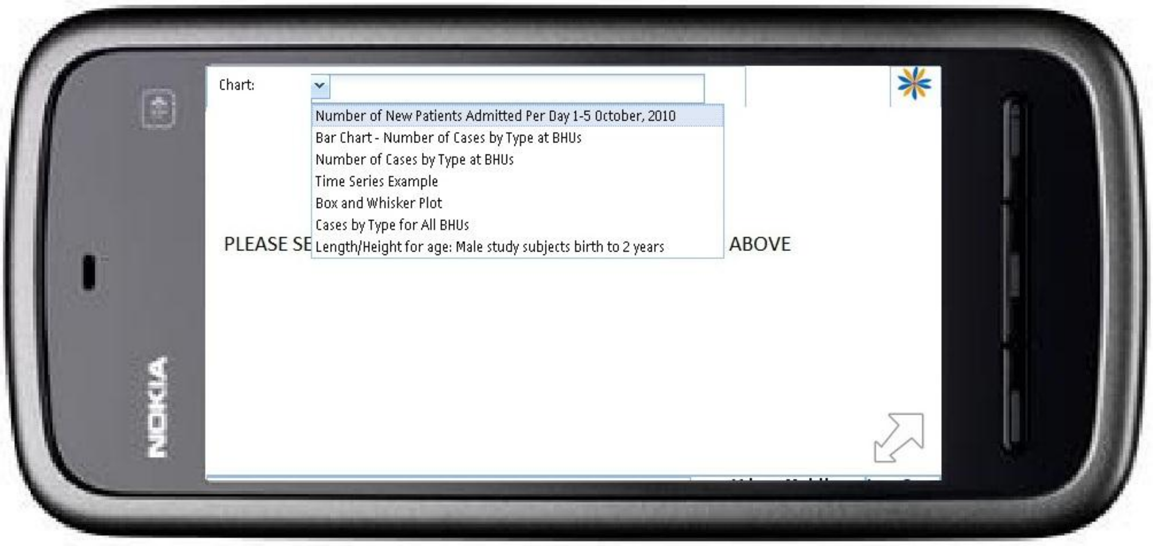


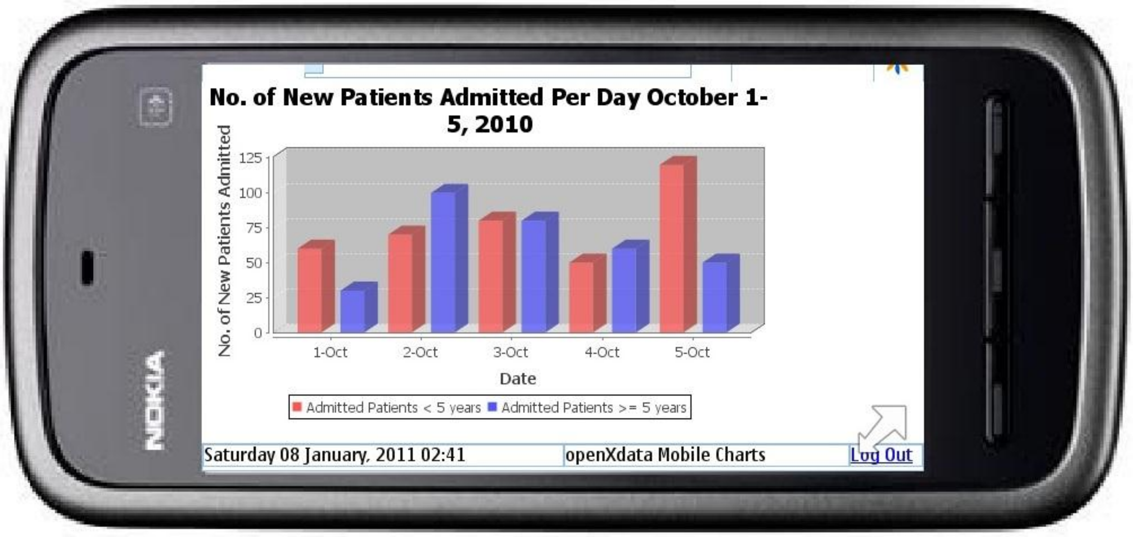
Chart:

- Number of New Patients Admitted Per Day 1-5 October, 2010
- Bar Chart - Number of Cases by Type at BHUs
- Number of Cases by Type at BHUs
- Time Series Example
- Box and Whisker Plot
- Cases by Type for All BHUs
- Length/Height for age: Male study subjects birth to 2 years

PLEASE SE ABOVE

Figure 15: Screen shot of the chart selection menu.

Figure 16 shows the chart that was highlighted in figure 15. After the user selects a chart, the chart image file that corresponds to it is displayed in the main viewing area. The timestamp is visible in the footer section.



No. of New Patients Admitted Per Day October 1-5, 2010
No. of New Patients Admitted
125
100
75
50
25
0
1-Oct
2-Oct
3-Oct
4-Oct
5-Oct
Date
🟥 Admitted Patients < 5 years
🟦 Admitted Patients >= 5 years
NOKIA

Saturday 08 January, 2011 02:41 | openXdata Mobile Charts | Log Out

Figure 16: Screen shot of the chart that was highlighted in the previous figure.

Figure 17 shows the box and whisker plot that was defined as an example in the description and usage section. It is comprised of data from a sample anthropometry study consisting of 103 length/height-for-age forms.

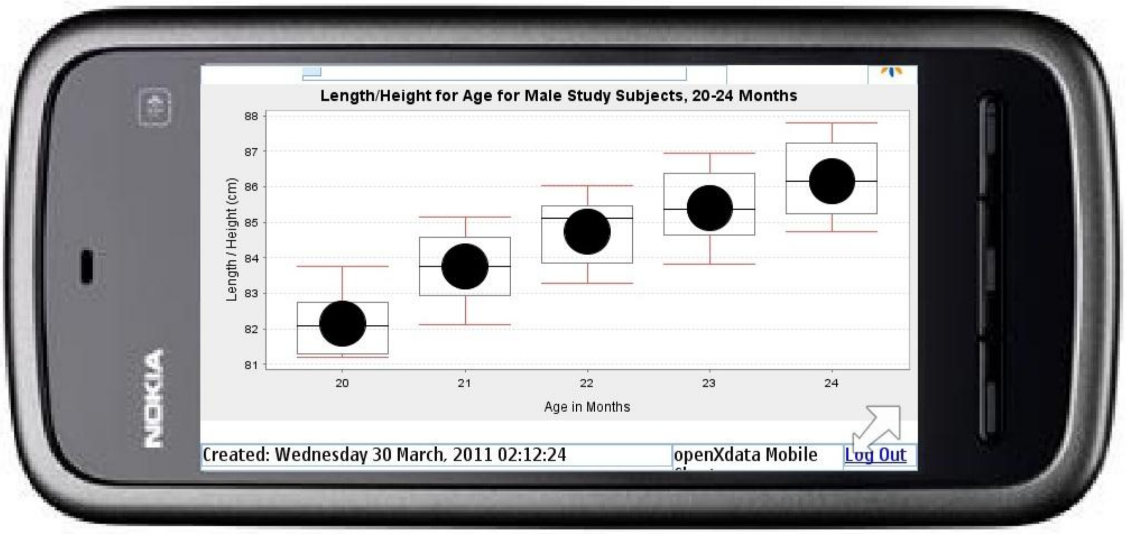

NOKIA
Length/Height for Age for Male Study Subjects, 20-24 Months
Length / Height (cm)
88
87
86
85
84
83
82
81
20
21
22
23
24
Age in Months
Created: Wednesday 30 March, 2011 02:12:24
openXdata Mobile
Log Out

Figure 17: Screen shot of a box and whisker plot made using a sample anthropometry study.

Figure 18 shows a pie chart with series labels, and figure 19 shows a box and whisker plot with multiple series and outliers.

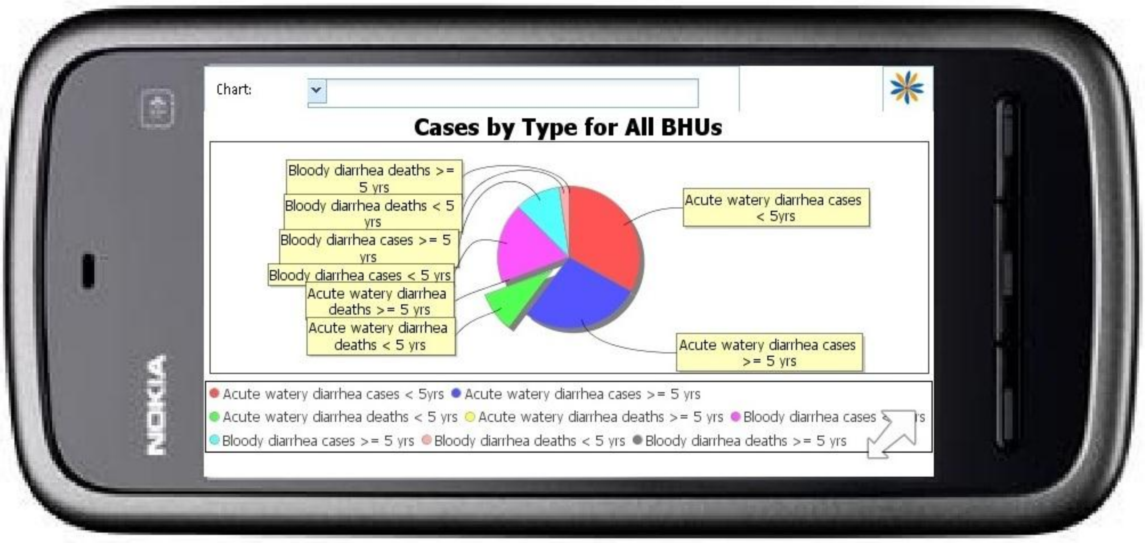
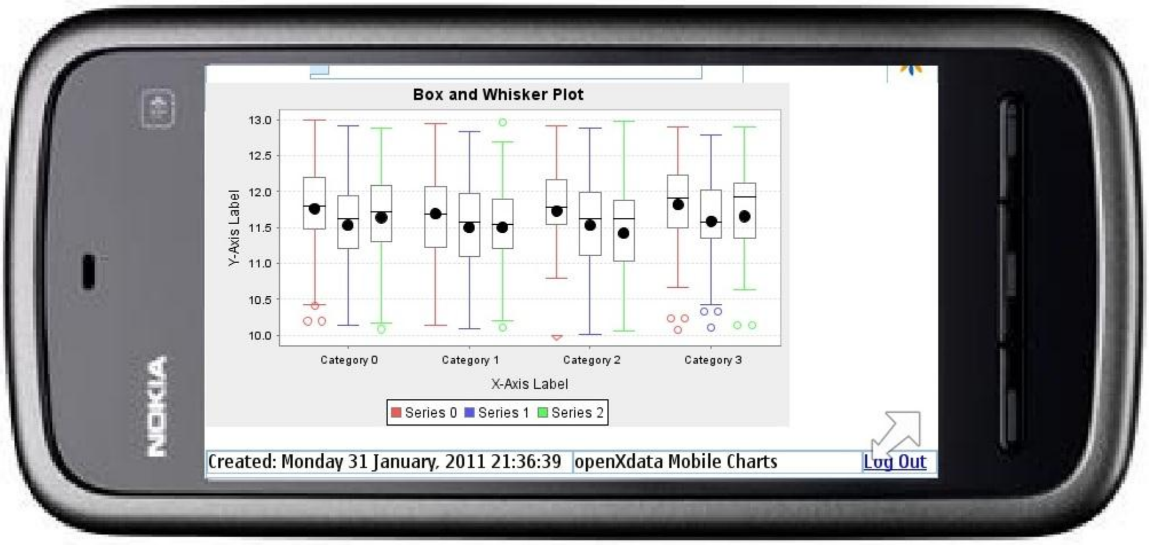

NOKIA
Chart:
Cases by Type for All BHUs
Bloody diarrhea deaths >= 5 yrs
Bloody diarrhea deaths < 5 yrs
Bloody diarrhea cases >= 5 yrs
Bloody diarrhea cases < 5 yrs
Acute watery diarrhea deaths >= 5 yrs
Acute watery diarrhea deaths < 5 yrs
Acute watery diarrhea cases < 5yrs
Acute watery diarrhea cases >= 5 yrs
● Acute watery diarrhea cases < 5yrs ● Acute watery diarrhea cases >= 5 yrs
● Acute watery diarrhea deaths < 5 yrs ○ Acute watery diarrhea deaths >= 5 yrs ● Bloody diarrhea cases
● Bloody diarrhea cases >= 5 yrs ● Bloody diarrhea deaths < 5 yrs ● Bloody diarrhea deaths >= 5 yrs

Figure 18: Screen shot of a pie chart with series labels.



Created: Monday 31 January, 2011 21:36:39 openXdata Mobile Charts

Figure 19: Screen shot of a box and whisker plot with multiple series and outliers.

4.2 RESULTS OF SURVEY - SYSTEM USABILITY SCALE AND ADJECTIVE RATING

4.2.1 PRE-SURVEY FINDINGS

While all 12 participants completed the SUS, they raised several issues and questions, and made several comments. None of these were noted verbatim at the time. The researcher did not anticipate the ranges of issues, questions and comments, so no formal consideration was given for recording them when the study was designed. Even so, it was important to note the issues, questions and comments. Firstly, it was found that all participants had questions regarding the survey before starting it. Many of these questions were basic, such as “What are you doing?” (in terms of the study) and “What kind of software have you made?” The researcher answered these questions and asked if participants understood what the study was about and what they were expected to do. In a few occasions, more questions were asked, which the researcher answered.

Due to the kinds of questions raised with the first few participants, the researcher decided to give all subsequent participants a brief introduction. This covered the “Purpose of the Research” section of the informed consent form. A sample database was opened in the (Navicat) database administration software. They were told that there were two distinct applications: one for making the graphs based on information in a chart definitions table (they were shown an example of such a table in the database administration software), and one for displaying the charts in a web application (they were not shown any examples of this). This introduction was in addition to what was done for all participants as listed in the Survey Conditions section. As a result, participants seemed to understand the research and their role in it.

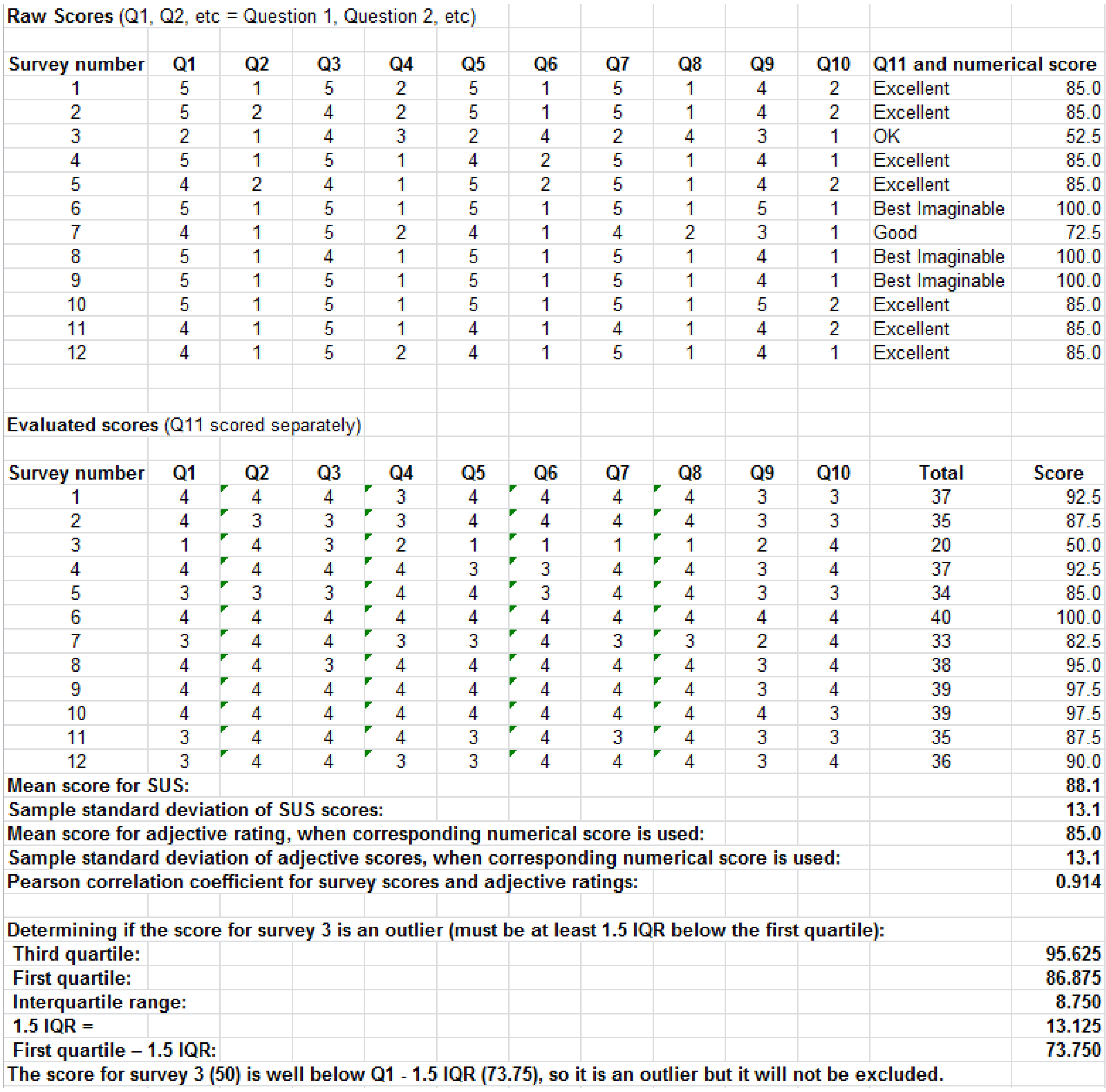
It was observed that after being shown the phone they would use for the evaluation, some participants expressed not being proficient in using touch-screens. The researcher told them that the touch-screen on the phone they would use may not be very sensitive. Another observation is that some participants had trouble following the instructions (see annex 2). All participants had difficulty with step 6. None of the participants encountered the “Java/ECMA script error” message. Even so, they seemed confused, as if they were expecting it. One participant was waiting for the error to appear, although it was listed in the instructions to follow that step only if the error appeared. The researcher made a note of this, and pointed out to all subsequent participants to skip that step if they did not see the error. The researcher did not anticipate the confusion this step would cause and kept the step on all surveys to ensure that all participants would read the same set of instructions.

4.2.2 SURVEY FINDINGS – SYSTEM USABILITY SCALE

After all 12 surveys were completed, a number was written on the back of each one to keep track of them. This was used to make sure all surveys had been entered into the spreadsheet to calculate scores averages, and totals. Each survey was scored according to the instructions provided by Brooke: [16]

1. Each of the first ten items has 5 possible answer choices. The left-most choice (“Strongly disagree”) has a score of 0, while the right-most choice (“Strongly agree”) has a score of 4. The remaining 3 choices have a score of 1, 2, and 3 when moving from left to right. This number was noted for each item.
2. For all odd-numbered items, the score is the number described above minus 1.
3. For all even-numbered items, the score is 5 minus the number described above.
4. Add the scores of all 10 items and multiply by 2.5 for the final score. SUS scores range from a minimum of 0 to a maximum of 100.

A spreadsheet was made to calculate scores, and the results are shown on the next page.



Raw Scores (Q1, Q2, etc = Question 1, Question 2, etc)

Survey number	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11 and numerical score	
1	5	1	5	2	5	1	5	1	4	2	Excellent	85.0
2	5	2	4	2	5	1	5	1	4	2	Excellent	85.0
3	2	1	4	3	2	4	2	4	3	1	OK	52.5
4	5	1	5	1	4	2	5	1	4	1	Excellent	85.0
5	4	2	4	1	5	2	5	1	4	2	Excellent	85.0
6	5	1	5	1	5	1	5	1	5	1	Best Imaginable	100.0
7	4	1	5	2	4	1	4	2	3	1	Good	72.5
8	5	1	4	1	5	1	5	1	4	1	Best Imaginable	100.0
9	5	1	5	1	5	1	5	1	4	1	Best Imaginable	100.0
10	5	1	5	1	5	1	5	1	5	2	Excellent	85.0
11	4	1	5	1	4	1	4	1	4	2	Excellent	85.0
12	4	1	5	2	4	1	5	1	4	1	Excellent	85.0

Evaluated scores (Q11 scored separately)

Survey number	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Total	Score
1	4	4	4	3	4	4	4	4	3	3	37	92.5
2	4	3	3	3	4	4	4	4	3	3	35	87.5
3	1	4	3	2	1	1	1	1	2	4	20	50.0
4	4	4	4	4	3	3	4	4	3	4	37	92.5
5	3	3	3	4	4	3	4	4	3	3	34	85.0
6	4	4	4	4	4	4	4	4	4	4	40	100.0
7	3	4	4	3	3	4	3	3	2	4	33	82.5
8	4	4	3	4	4	4	4	4	3	4	38	95.0
9	4	4	4	4	4	4	4	4	3	4	39	97.5
10	4	4	4	4	4	4	4	4	4	3	39	97.5
11	3	4	4	4	3	4	3	4	3	3	35	87.5
12	3	4	4	3	3	4	4	4	3	4	36	90.0

Mean score for SUS:	**88.1**
Sample standard deviation of SUS scores:	**13.1**
Mean score for adjective rating, when corresponding numerical score is used:	**85.0**
Sample standard deviation of adjective scores, when corresponding numerical score is used:	**13.1**
Pearson correlation coefficient for survey scores and adjective ratings:	**0.914**

Determining if the score for survey 3 is an outlier (must be at least 1.5 IQR below the first quartile):

Third quartile:	**95.625**
First quartile:	**86.875**
Interquartile range:	**8.750**
1.5 IQR =	**13.125**
First quartile – 1.5 IQR:	**73.750**

The score for survey 3 (50) is well below Q1 - 1.5 IQR (73.75), so it is an outlier but it will not be excluded.

Table 1: Results of survey.

It was found that the mean SUS score was 88.1 out of a possible of 100. Comparing this score to the three other rating scales (Figure 7) reveals that a score of 88.1 is considered “Acceptable” according to the Acceptability Ranges. Further, this score equates to a “B” in the Grade Scale and “Excellent” in the Adjective Ratings scale. A 95% confidence interval was calculated using the t-test due to the small sample size, and it is [79.8, 96.4]. Calculating the Pearson correlation coefficient for survey scores and the adjective ratings resulted in r =

0.914. This indicated a strong positive correlation between survey scores and adjective ratings.

4.2.3 SURVEY FINDINGS – ADJECTIVE RATING

An average score for the Adjective Ratings scale was calculated. This involved noting the SUS score that corresponds to the adjective ratings. Since the lowest adjective rating given was “OK” (survey 3), the researcher did not note the SUS score that corresponds to the two lowest adjective ratings. It was clear that the two highest adjective ratings, “Best Imaginable” and “Excellent” correspond to SUS scores of 100 and 85, respectively. The SUS scores corresponding to “Good” and “OK” are not as precise, so values of 72.5 and 52.5 were assumed. Using this method, the researcher found that the average SUS-equivalent score for the adjective rating was 85. This corresponds to an average adjective rating of “Excellent” and the letter grade “B.” A 95% confidence interval was also calculated for this estimate, and it is [76.7, 93.3].

4.2.4 SURVEY FINDINGS – PARTICIPANT COMMENTS

Participants raised several issues and questions, and made several comments. None of these were recorded verbatim as that was not mentioned in the informed consent form. The researcher did not anticipate the ranges of issues, questions and comments, so recording them were not a part of the study design. Even so, it was important to note the issues, questions and comments. Listed below are descriptions of participant comments. These were not recorded on the same sheet of paper on which the participant completed the SUS. They were recorded on a separate electronic document after the survey was complete and the participant had left. The comments are listed in the order in which they were made.

The first participant had no difficulty completing the tasks and gave a positive reaction. No comments were recorded, and the researcher was later advised to start noting these. Participant 2 said oXd MC Web Application Component was “really good.” Participant 3 was the most critical of all twelve. This was reflected in both the comments and the score. The participant complained about the drop-down menu, saying that the message on the drop-down menu, "Select a chart..." disappears when the user selects a chart. The researcher replied (thinking that it was a question on using the software) that it re-appears when touching/clicking outside the menu. The participant said that it was not consistent. After completing the survey, the participant specifically said that the software was not very consistent and that the participant was very critical in the survey. It was later found that this participant gave the lowest ratings and these were generally opposite to the ratings given by others.

Participant 4 started by saying that the participant is not a Masters candidate. This alerted the researcher to a mistake in the introduction of the informed consent form, which states that “Masters Candidates currently at the Centre for International Health, University of Bergen, are invited to participate.” The researcher thanked the participant for pointing it out and said that it was a mistake and to please overlook it. The survey was completed with incident and the participant said oXd MC Web Application Component was “really good” and “easy to use.”

Participant 5 liked oXd MC Web Application Component and told the researcher to contact the participant when it is released. Participant 6 gave the best ratings. This participant really liked it and said it was very easy to use and also asked the researcher to email the participant when it is ready for release. The participant suggested that the text and fields of the login screen be made bigger so it is easier to select the appropriate fields. Further, the participant asked what would be the reward for the researcher’s "intellectual investment." The researcher responded by saying naming the Master degree and future opportunities as rewards.

There were no comments from Participant 7. The participant seemed to be in a hurry, and left quickly afterwards. Participant 8 said oXd MC Web Application Component was "very nice" while using it, when the graphs appeared. The participant also wished the researcher had made and released oXd MC earlier so that the participant could have used it in the field. Participant 9 said (while using oXd MC Web Application Component) that “this is how I would like software to be.” The participant made a comment regarding item 9 (“I felt very confident using the product”), saying that the participant was not very confident, but only because it was the first time using the program. The participant said that the second time around would be very easy, and added that it was very self-explanatory. Participant 10 commented after the survey “that was fast,” and that the participant is waiting for it to be available. Participant 11 said “this is good” while using oXd MC Web Application Component. Participant 12 asked if oXd MC Web Application Component could also be accessed from a computer.

In addition to participant comments, there are some other points to note. The researcher quickly found that showing participants the contents of a sample database really helped them understand oXd MC in general. Another point worthy of note is that some participants were confused with step 6 in the instructions (how to deal with the Java-ECMA error if they encountered it).

5. DISCUSSION

In this study, the researcher aimed to prototype and evaluate software that fulfilled several conditions so that it could help health researchers through graphical reporting.

The end product, oXd MC, was developed in Java and does not rely on external applications to create graphs. Further, it is generic meaning it is not specific to one or a few studies. The use of simple properties files and SQL mean that no elaborate syntax is required to run the program. All configurations/definitions are limited to plain text labels, delimiters, and basic SQL commands, thus reducing reliance on technical personnel. The automatic updates through repetition mean that there is no need to constantly interact with the program. All data is retrieved from a MySQL database, meaning that oXd MC can use study data collected and stored using openXdata. Only a few changes are required in the openXdata database, and these do not interfere with the operation of openXdata in any way. Once configurations have been made, most interaction with oXd MC is via oXd MC Web Application Component, which makes graphs available for viewing through a password-protected web page. The user interface for this web page is simple and small in dimensions, meaning it can be viewed through a smart phone.

The results from the evaluation suggest that oXd MC Web Application Component will be beneficial to health researchers. A modified SUS which includes an eleventh item with adjective ratings was used in the evaluation, and the mean score for oXd MC Web Application Component was 88.1 out of a possible score of 100 (with a sample standard deviation of 13.1). Further, the mean score for adjective rating (when the corresponding numerical score is used) was 85 out of a possible score of 100 (with a sample standard deviation of 13.1). Only one survey gave oXd MC Web Application Component a low score (50 out of 100), and this was an outlier as it was well below 1.5 IQR (73.75). Finally, the eleventh-item adjective rating was found to be a good predictor of SUS scores. Calculating the Pearson correlation coefficient for survey scores and the adjective ratings resulted in r = 0.914. This indicated a strong positive correlation between survey scores and adjective ratings. In fact, the correlation was stronger than that observed by Bangor et al (r = 0.822). [17] To be clear, the researcher calculated this to see what the correlation was between survey scores and adjective ratings and not to support or refute the findings of Bangor et al.

5.1 LIMITATIONS

There were several limitations in this study. These were due to software design as well as the manner in which the survey was conducted. The oXd MC Web Application Component was developed in an incremental fashion. Functional areas were developed in separate projects and then added to another project, thereby allowing the researcher to test in isolation before adding to a larger and more complex program. For this reason, many package and class names may not seem related to the project. Supporting files were already there and many names were not changed to keep many files from having to be edited. This may not follow good software design practices, but it was done due to time constraints.

Further limitations due to software design were reflected in the server-side and web application components (oXd MC Chart Generator Component and oXd MC Web Application Component). Chart definitions have to be set by manually entering SQL queries or sub-strings of queries into the study database. Usernames and passwords also have to be entered manually and that too in plain text. This has obvious security implications, but this is not an area of emphasis for the prototype nature of the software. The researcher believes that it would not be too difficult to add encryption function which relies on a hashed password and salt. Information in the properties file also has to be entered manually. There is no Graphical User

Interface (GUI) for installing and configuring any part of oXd MC. This means some technical experience is required to install and run oXd MC.

Once it is installed, users must cope with other limitations. The initial delay before starting to make graphs and the repeat interval for making graphs are in milliseconds. The total number of times to iterate is also required. These are minor weaknesses that could be addressed by changing milliseconds to minutes and making it so that a certain number (for example negative one) would signal the software to iterate indefinitely. When oXd MC Chart Generator Component is operating, there are two main limitations. The first is that some parameters for creating the various kinds of charts are hard coded. For example all bar charts will have a vertical orientation and default colours, although JFreeChart allows these (as well as other parameters for other chart types) to be customized. The second limitation when oXd MC Chart Generator Component is operating is that the SaveChart class creates a new instance of the ConnectToDatabase class and calls its UpdateDatabase method to update the database (runs a query to update the date and time that graph was created). There is surely a more elegant approach to update the database than to call the main database connection/chart making class every time.

The oXd MC Web Application Component has one main limitation when operating. While there is a login capability, there is no logout capability. This was not addressed due to time constraints and because it is not a key function of the software. Two more issues arose when accessing oXd MC Web Application Component from the Nokia 5230. A Java/ECMA error is thrown when loading oXd MC Web Application Component. However, the software still works as expected. The phone browser also freezes on occasion, requiring the browser to be restarted. However, no errors were reported when two members of the audience accessed oXd MC Web Application Component during the researcher’s presentation on 31 March 2011.

The researcher does not know exactly what phones were used, but they were not the Nokia 5230. For this reason, error logging on the phone was disabled for the survey, and the errors are not considered to be major limitations.

There were also limitations due to the manner in which the survey was conducted. First, there was no pilot phase conducted. This had several consequences. First it caused the researcher to amend the instructions for the survey after it had already started. Some candidates had trouble following the instructions, especially as it included a step related to the Java/ECMA error (see annex 2) that did not apply since error logging was disabled. The researcher had to tell some candidates to skip that particular step when they expressed confusion. Second, the researcher failed to properly capture the qualitative comments made by the test users during the survey. The researcher did not plan to note participant comments. This meant that the first participant’s comments were not noted, but the researcher was advised by the supervisors to start doing so after this, but it is obvious that a more rigorous capture of these comments would have been beneficial.

Another important limitation is the calculation of sample size. In the article on which it was based, the question was very simple: which of two alternatives (web pages) is best in the view of the users? When the question is simple and if the difference between the alternatives is large, the sample size becomes very small. In our particular case, there was no comparison so that makes the question more difficult to formulate. But if we had wanted a smaller confidence interval around our estimate of 88 points out of 100, we would have needed more participants. And if we had looked at each component separately we may have needed an even larger sample.

6. CONCLUSIONS AND RECOMMENDATIONS

6.1 CONCLUSIONS

The researcher developed prototype software and evaluated its user acceptability. The following results and findings suggest that all study objectives were met and that participants generally had a very good experience using oXd MC Web Application Component:

1. oXd MC CGC is a Java application and oXd MC Web Application Component is a Java web application meaning both are platform independent.
2. oXd MC should run on computers with different operating systems.
3. Graphs generated oXd MC Chart Generator Component are available for viewing by means of the generic web application oXd MC Web Application Component.
4. oXd MC Web Application Component has a simple and small user interface for easy viewing via a smart phone.
5. oXd MC Web Application Component was evaluated for user acceptability by using a modified version of the SUS.
6. The mean SUS score was 88.1 out of 100, which equates to a "B" in the traditional grade scale and "Excellent" in the Adjective Ratings scale.
7. The Pearson correlation coefficient for survey scores and adjective ratings resulted in r = 0.914, indicating a strong positive correlation between the two.

6.2 RECOMMENDATIONS

While the researcher believes that all study objectives were met, there are several recommendations for similar studies in the future.

The following recommendations are for software development:

1. Future software could be made more secure through authentication against an encrypted password in the database.

2. The login screen for the web application could have large fields and buttons so users can easily enter the username, password, and press the login button.
3. The web application could have a logout feature.
4. A GUI could be added to allow users to define graphs and to even set specific intervals for each graph (to address possible requirements such as for some graphs to be made daily while others are made weekly).

The following recommendations are for survey design:

1. A small pilot study could be conducted for the researcher to know if participants understand the instructions. The researcher could note what action each participant took and encourage comments, and then make appropriate changes to the final version.
2. A script for introducing the study could be written. It could be done in such a way that no part of the software is shown and nothing said that could prejudice the participant's view regarding the software. However, any nuances related to the smart phone would have to be identified so that participants do not confuse those as being weaknesses in the software they are evaluating.
3. All participant questions and comments could be recorded (as the researcher is explaining the instructions and waiting while the participant goes through the instructions and uses the software). This may be complicated, as it may be best to record the conversation and not pause during the introduction to take notes. However, participants may give different responses if they know that everything is being recorded.

REFERENCE LIST

1. *OpenMRS overview*. 2010 [cited 2010 May 19]; Available from: http://openmrs.org/wiki/OpenMRS_Overview.
2. *Introduction to XML*. 2010 [cited 2010 May 19]; Available from: http://www.w3schools.com/xml/xml_whatis.asp.
3. *KML tutorial*. 2010 [cited 2010 May 19]; Available from: http://code.google.com/apis/kml/documentation/kml_tut.html.
4. *AutoHotkey - free mouse and keyboard macro program with hotkeys and autotext*. 2009 [cited 2010 May 19]; Available from: http://www.autohotkey.com/.
5. *Google Chart tools/image charts (aka Chart API) terms of use*. 2009 [cited 2010 May 19]; Available from: http://code.google.com/apis/chart/terms.html.
6. Deitel, H.M., & Deitel, P. J., *Java™ how to program, sixth edition*, 2004, Prentice Hall.
7. *Making remote procedure calls - Google Web Toolkit - Google Code*. 2011 [cited 2011 January]; Available from: http://code.google.com/webtoolkit/doc/latest/tutorial/RPC.html.
8. *Transport objects over RPC – GWT object serialization – tutorial*. 2007 [cited 2011 January]; Available from: http://developerlife.com/tutorials/?p=131.
9. Kapur, A. *"Create a login screen" GWT tutorials by Addy Kapur*. 2009 [cited 2011 January]; Available from: http://checkappointments.com/gwt/build-a-login-application-in-gwt-spring-and-hibernate/create-a-login-screen/ (no longer accessible).
10. *GWT and tomcat: Create .war using eclipse to deploy .war on tomcat*. 2009 [cited 2011 April 25]; Available from: http://blog.elitecoderz.net/gwt-and-tomcat-create-war-using-eclipse-to-deploy-war-on-tomcat/2009/12/.
11. *Using the Timer and TimerTask classes*. 2005 [cited 2011 January]; Available from: http://enos.itcollege.ee/~jpoial/docs/tutorial/essential/threads/timer.html.
12. *Chart « Java*. 2009 [cited 2011 January]; Available from: http://www.java2s.com/Code/Java/Chart/CatalogChart.htm.
13. *JFreeChart tutorial (reprinted) | Screaming Penguin*. 2005 [cited 2011 January]; Available from: http://www.screaming-penguin.com/node/4005.
14. *Informed consent form templates*. 2009 [cited 2011 April 8]; Available from: http://www.who.int/rpc/research_ethics/informed_consent/en/.
15. Tullis, T.S., & Stetson, Jacqueline N., *A comparison of questionnaires for assessing website usability*, in *Usability Professionals Association 2004*2004: Minneapolis, Minnesota USA.
16. Brooke, J., *SUS: a "quick and dirty" usability scale*, in *Usability evaluation in industry*, B.T. Patrick W. Jordan, Ian Lyall McClelland, Bernard Weerdmeester Editor 1996, CRC Press. p. 189-194.
17. Bangor, A., Kortum, P., & Miller, J.A., *Determining what individual SUS scores mean - adding an adjective rating scale.* Journal of usability studies, 2009. **4**(3): p. 114-123.
18. Bangor, A., Kortum, P., & Miller, J.A., *The System Usability Scale (SUS): An empirical evaluation.* International Journal of Human-Computer Interaction, 2008. **24**(6).

ANNEXES

1. Informed Consent Form - blank
2. System Usability Survey Instructions
3. System Usability Survey - blank
4. System Usability Survey – 12 completed

Informed Consent Form for openXdata Mobile Charts System Usability Survey
This informed consent form is for evaluating the usability of the software openXdata Mobile Charts. Masters Candidates currently at the Centre for International Health, University of Bergen, are invited to participate. The evaluation will be used in completing the Master Thesis of Owais Uddin Ahmed, who is the software developer of openXdata Mobile Charts and a Master degree candidate at the Centre for International Health, University of Bergen.

The study is a part of the OMEVAC (Open Mobile Electronic Vaccine Trials) project, which is "an interdisciplinary project to improve quality of vaccine trials in low-resource settings" (for more information, visit http://www.uib.no/rg/childhealth/projects/projects-child-health-nutrition/omevac-open-mobile-electronic-vaccine-trials). Funding for the OMEVAC project comes from the Research Council of Norway (for more information, visit http://www.forskningsradet.no/servlet/Satellite?c=Prosjekt&cid=1205852114030&pagename=ForskningsradetNorsk/Hovedsidemal&p=1181730334233).

This Informed Consent Form has two parts:

- **Information Sheet (to share information about the study with you)**
- **Certificate of Consent (for signatures if you choose to participate)**

You will be given a copy of the full Informed Consent Form

Part I: Information Sheet

Introduction

I am Owais Uddin Ahmed, a second-year Masters candidate at the Centre for International Health, University of Bergen. I have developed software for creating charts based on data from an existing database and displaying those charts in a form suitable for mobile phones. I am going to give you information and invite you to be a part of evaluating this software. You do not have to decide today whether or not you will participate in the evaluation. Before you decide, you can talk to anyone you feel comfortable with about the evaluation. This consent form may contain words or phrases that you do not understand. Please ask me to stop as we go through the information and I will take time to explain. If you have questions later, you can ask me.

Purpose of the Research

Researchers conducting studies related to international health rely extensively on regular data updates to judge progress and to make relevant decisions. In studies requiring collection of data from many individuals and/or locations, generating graphs of the data may help in interpretation and decision making. I have developed openXdata Mobile Charts, which is software that can create graphs from data that is in an existing database, and then make those graphs available on a website.

There are two components to openXdata Mobile Charts. The first component is a graph generator, which is supposed to be installed on a computer of the researcher's (meaning the user, not me) choosing. Some requirements must be met, such as that the computer must be able to run a Java program, the program must be able to access the desired database, the desired graphs must be defined (in terms of what data to use, titles, and labels), and a location must be defined to save the graphs. The second component is an internet application, which is also supposed to be installed on a computer of the researcher's choosing. As with the first component, some requirements must be met, such as that the computer must be able to host a web application, and the location of where the graphs are saved must be defined. All the functions of openXdata Mobile Charts occur on the computer(s) of the researcher's choosing, so no data is sent to a third party.

Once it is installed, most interaction with openXdata Mobile Charts is through the internet application, which has been designed to be accessed through a mobile phone. I want to learn how usable researchers find the internet application component of openXdata Mobile Charts, when accessed through a mobile phone.

Type of Research Intervention
This research will involve your participation in using openXdata Mobile Charts (internet application component) through a mobile phone, and then completing a survey. The participation will take about fifteen minutes.

Participant Selection
You are being invited to take part in this research because I feel that your experience as a researcher can contribute much to my understanding of the usability of openXdata Mobile Charts (internet application component).

Voluntary Participation
Your participation in this research is entirely voluntary. It is your choice whether to participate or not. You may change your mind later and stop participating even if you agreed earlier.

Procedures
A. I am asking you to help me learn about the usability of openXdata Mobile Charts (internet application component). I am inviting you to take part in this research. If you accept, you will be asked to follow written instructions on using openXdata Mobile Charts (internet application component). You will be asked to do this through a mobile phone that will be provided to you for the duration of participation. After following the written instructions, you will be asked to complete a survey on the usability of openXdata Mobile Charts (internet application component). The survey will be provided and collected by Owais Uddin Ahmed. The information recorded will be anonymous (your name will not appear on the form), and only a form number will be used.

Duration
Participation will take about fifteen minutes in total.

Risks
No personal information will be collected, and no opinions on sensitive topics will be sought.

Benefits
There will be no direct benefit to you, but your participation is likely to help me learn how usable openXdata Mobile Charts (internet application component) is.

Reimbursements
You will not be provided any incentive to take part in the research.

Confidentiality
The survey will be conducted anonymously, and no personal information will be collected.

Sharing the Results
The knowledge that I gain from this research will be published in my Master Thesis, which will be made available at the Centre for International Health, University of Bergen.

Right to Refuse or Withdraw
You do not have to take part in this research if you do not wish to do so. You may stop participating in the survey at any time that you wish. I will give you an opportunity at the end of the survey to review your remarks, and you can modify or remove any of them.

Who to Contact
If you have any questions, you can ask them now or later. If you wish to ask questions later, you may contact Owais Uddin Ahmed at: Owais.Ahmed@student.uib.no and +47 9090 6656.

Part II: Certificate of Consent
I have been invited to participate in research on the usability of openXdata Mobile Charts (internet application component).

I have read the foregoing information. I have had the opportunity to ask questions about it and any questions I have been asked have been answered to my satisfaction. I consent voluntarily to be a participant in this study

Print Name of Participant______________________________

Signature of Participant _______________________________

Date ___________________________
Day/month/year

Statement by the researcher/person taking consent

I have accurately read out the information sheet to the potential participant, and to the best of my ability made sure that the participant understands that the following will be done:

The participant will be asked to follow written instructions regarding the use of openXdata Mobile (internet application component).

A mobile phone will be provided (for the duration of the survey) which the participant will be asked to use when following the written instructions.

The participant will be asked to complete a survey evaluating the usability of openXdata Mobile (internet application component).

I confirm that the participant was given an opportunity to ask questions about the study, and all the questions asked by the participant have been answered correctly and to the best of my ability. I confirm that the individual has not been coerced into giving consent, and the consent has been given freely and voluntarily.

A copy of this Informed Consent Form has been provided to the participant.

Print Name of Researcher/person taking the consent______________________________

Signature of Researcher /person taking the consent_______________________________

Date ____________________________
Day/month/year

Instructions for Completing the openXdata Mobile Charts (internet application component) Usability Survey

Please follow the instructions below to complete the usability survey for openXdata Mobile Charts (internet application component).

1. You will be given a Nokia 5230 mobile phone for assessing openXdata Mobile Charts (internet application component) during the survey. Please use it to access the program.
2. The phone's internet browser will be open and the login page for the program displayed when the phone is given to you. The phone has a touch-sensitive screen. Touching the screen when the internet browser is open has the same effect as pointing and clicking on the screen using a mouse on a personal computer. Note that the touch screen may not feel very sensitive or responsive. This can lead to wrong characters being entered and wrong fields and buttons being selected. This is an issue with the phone and not the program you will assess. For this reason, please look past these issues and do not consider them in the evaluation.
3. Two labels will be visible on the login page: “Username” and “Password” Please touch the box to the right of “Username” and enter the characters in bold: **user** and touch the green check mark when complete.
4. Please touch the box to the right of “Password” and enter the characters in bold: **pass** and touch the green check mark when complete.
5. Please touch the grey button labelled “login.”
6. If the phone displays a “Java/ECMA script error” message, touch the “OK” button to proceed. If the message appears again, touch the “OK” button once again.
7. The home page will be displayed. There will be a drop-down menu at the top of the screen, with a small downward pointing arrow and the label “Select a chart...” Please touch the drop down menu. A list of available charts will be displayed. These charts have automatically been created from data in the database. For purposes of the survey, they are based on sample data. In actual use, they would be based on actual study data, with graphs being created as per the user's definitions.
8. Please select any chart displayed on the list. The selected chart should load on the screen.
9. Please touch the screen and slide your finger up on it. Touching and sliding your finger (either up or down) has the same effect as clicking and dragging on the screen using a mouse on a personal computer.
10. Please look at the bottom left of the screen. A date and time should be listed. This is the date and time in which graph was created.
11. Please touch the screen and slide your finger down on it. This will scroll back to the top of the home page.
12. Please view another chart by following the instructions in steps 7 through 11 (including step 11).
13. Please view another chart by following the instructions in steps 7 through 10 (including step 10).
14. Please read the instructions on the next page and complete the 11 item survey.

System Usability Scale for openXdata Mobile Charts (internet application component)

Please check the box that reflects your immediate response to each statement. Do not think too long about each statement. Make sure you respond to every statement. If you do not know how to respond, simply check box “3.”

	Strongly disagree				Strongly agree
1. I think that I would like to use this product frequently	□ 1	□ 2	□ 3	□ 4	□ 5
2. I found the product unnecessarily complex	□ 1	□ 2	□ 3	□ 4	□ 5
3. I thought the product was easy to use	□ 1	□ 2	□ 3	□ 4	□ 5
4. I think that I would need the support of a technical person to be able to use this product	□ 1	□ 2	□ 3	□ 4	□ 5
5. I found the various functions in this product were well integrated	□ 1	□ 2	□ 3	□ 4	□ 5
6. I thought there was too much inconsistency in this product	□ 1	□ 2	□ 3	□ 4	□ 5
7. I would imagine that most people would learn to use this product very quickly	□ 1	□ 2	□ 3	□ 4	□ 5
8. I found the product very awkward to use	□ 1	□ 2	□ 3	□ 4	□ 5
9. I felt very confident using the product	□ 1	□ 2	□ 3	□ 4	□ 5
10. I needed to learn a lot of things before I could get going with this product	□ 1	□ 2	□ 3	□ 4	□ 5

11. Overall, I would rate the user-friendliness of this product as:

□ Worst Imaginable □ Awful □ Poor □ OK □ Good □ Excellent □ Best Imaginable