

CSMCR SURVEY: AN ANDROID/WEB APP

A Dissertation Submitted to The University of Manchester for the Degree of Master of
Science in the Faculty of Science and Engineering

2016

Hizal Hanis Bin Hashim

SCHOOL OF COMPUTER SCIENCE

Table of Contents

Abstract	6
Declaration.....	7
Intellectual Property Statement	8
Acknowledgement	9
Chapter 1 : Introduction	10
1.1. Aim and Objectives	12
1.2. Deliverables.....	13
1.3. Structure of the Dissertation.....	14
Chapter 2: Background	16
2.1. Feasibility Study.....	16
2.2. Modes of Data collection	19
2.3. Survey Instruments	20
2.4. Mobile Web Vs Mobile App	21
2.5. Software Development Strategies	22
2.6. Software Development Methodology	24
2.7. Usability.....	26
Chapter 3: Method	29
3.1. Defining Requirements	29
3.2. Agile Development.....	30
3.3. Data Analysis	33
3.4. Risk management.....	36
Chapter 4: Design.....	38
4.1. Survey Questionnaire.....	38
4.2. Requirements Gathering.....	45
4.3. System Design	49
4.3.1. Mobile App	51
4.3.2. Web Application	55
Chapter 5: Implementation	59
5.1. Environment setup.....	59
5.2. Database.....	61
5.3. System Implementation	67
5.3.1. Mobile App	67
5.3.2. Web Application	73

Chapter 6: Evaluation	82
6.1. Technical Evaluation.....	82
6.1.1. Database	82
6.1.2. Hardware	83
6.1.3. Performance	85
6.2. Functional Evaluation.....	89
6.2.1. Requirement.....	89
6.2.2. Usability	90
6.3. Security Evaluation.....	93
6.4. Measure of Effectiveness	95
Chapter 7: Conclusion.....	99
7.1. Project Summary.....	100
7.2. Project Achievement.....	101
7.3. Issues Encountered	102
7.4. Future Work	105
Bibliography	107

Word Count: 19,878

List of Figures

Figure 1-1: Existing survey process	10
Figure 2-1: Survey Monkey Template	16
Figure 2-2: Existing approach and proposed cross-platform development model.....	23
Figure 4-1: Use Case Diagram	47
Figure 4-2: Data, System Management and Mobile App Layers	50
Figure 4-3: MVC Structure	51
Figure 4-4: Apache Cordova Architecture (55)	52
Figure 4-5: Android Activity Lifecycle States and Transitions (57)	54
Figure 4-6: Web Application Controllers	56
Figure 4-7: Web Application Models	56
Figure 4-8: Web Application Views.....	57
Figure 5-1: Overall System Set Up	59
Figure 5-2: Android Studio and AVD	60
Figure 5-3: phpMyAdmin Interface for Database Management	66
Figure 5-4: Survey App Homepage	72
Figure 5-5: Additional features in Homepage after Administrator Login.....	72
Figure 5-6: Survey Mode.....	72
Figure 5-7: List of Saved Responses	72
Figure 5-8: Administrator Login to the App	73
Figure 5-9: Sync App	73
Figure 5-10: Login Page.....	74
Figure 5-11: Homepage.....	75
Figure 5-12: Survey Mode.....	76
Figure 5-13: List of surveys	76
Figure 5-14: Analysis Dashboard.....	78
Figure 5-15: Contingency Table with Data Filtering	78
Figure 5-16: Add New User	79
Figure 5-17: Survey Question Settings.....	80
Figure 5-18: Data Backup and Restore	81
Figure 6-1: CSMCR Survey Mobile App in Various Devices	83
Figure 6-2: Performance trade-offs	85
Figure 6-3: Workload Model for Performance Testing.....	86
Figure 6-4: Data Analysis Performance Test.....	87
Figure 6-5: JMeter Results in Aggregate Graph	88
Figure 6-6: Appvigil Vulnerability Scanner Report.....	95
Figure 6-7: Mock Survey Session for Evaluation.....	96
Figure 7-1: CSMCR Survey System	99

List of Tables

Table 2-1: Comparison between Free Survey and Proposed Survey System	17
Table 2-2: Agile Development Methods	25
Table 3-1: Comparison of XP Principle and CSMCR Survey Project Setting	31
Table 3-2: Types of Data Analysis	35
Table 3-3: Project Risks, Critical Elements and Mitigation Plan	36
Table 4-1: Survey Questionnaire Summary	40
Table 4-2: Summary of Sprints, User Stories and Acceptance Test.....	45
Table 4-3: Summary of Project Requirements.....	48
Table 4-4: Summary of Web Application Functionalities	57
Table 5-1: Data Structure for CSMCR Survey Database.....	62
Table 6-1: Results from Hardware Evaluation for Mobile App.....	84
Table 6-2: Requirement Evaluation by Use Case	89
Table 6-3: Summary of Usability Evaluation with Users.....	91
Table 6-4: Types of Security Test and Result	94
Table 6-5: Parameters for Measurement of Effectiveness.....	96
Table 6-6: Results from the Measure of Effectiveness (Average for 10 Respondents).....	97

Abstract

The CSMCR Survey is a system comprising of an Android app and web application to conduct and manage surveys for events organized by the School of Computer Science, University of Manchester. As one of the reputable institutions for computer science, the school regularly welcomes visitors to its events such as the Open Day. Previously, visitors were handed with survey questionnaires' papers to obtain their feedback. The school administrative staff will then transfer the respondents' answers from the papers into digital format. This is done in order for the feedback to be efficiently analysed but the process is rather tedious. Consequently, data analysis capability becomes limited as the data collected is not properly structured. Therefore, to address these issues, this project developed a comprehensive system consisting of mobile app and web application for survey data capturing and analysis. Agile and extreme programming methodology was used in the overall development process. Developments were conducted in iterations with small releases. At the end of the project, a fully functional survey system was conceived and ready to use by the school. Extreme programming method proves to be effective and suitable for single programmer in managing the overall system development. Evaluation results indicate that the system is optimized, safe and fulfils all the project requirements whereby users are able to use the system for its intended objectives.

Keywords: *Questionnaire survey; Mobile apps; Hybrid mobile development; Agile software development; Extreme Programming*

Declaration

No portion of the work referred to in the dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Intellectual Property Statement

- i. The author of this dissertation (including any appendices and/or schedules to this dissertation) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.
- ii. Copies of this dissertation, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has entered into. This page must form part of any such copies made.
- iii. The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the dissertation, for example graphs and tables (“Reproductions”), which may be described in this dissertation, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.
- iv. Further information on the conditions under which disclosure, publication and commercialisation of this dissertation, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy, in any relevant Dissertation restriction declarations deposited in the University Library, The University Library’s regulations.

Acknowledgement

All praise to Allah the Almighty for giving me the strengths and blessings to complete this dissertation. To my parents, thank you for your love and support throughout my life. To my wife Anna Mastura, my son Hizal Hifzhan and my daughter Hana Humaira, thank you for encouraging and inspiring me to follow my dreams.

Special appreciation goes to my supervisor, Dr. Gavin Brown for his supervision and support throughout this journey. I also would like to thank everyone who contributed directly or indirectly towards the successful completion of this dissertation.

Chapter 1 : Introduction

This project aims to develop a systematic tool for conducting surveys and analysing the data for the School of Computer Science, University of Manchester. As one of the leaders in research breakthroughs and innovations, the school receives numerous requests from outside parties such as prospective students, researchers and academicians to visit the school. This is not surprising as some of the most important technological advancements took place here. For example, the world's first stored-program computer, dubbed the 'Manchester Baby' was developed here in 1948 and Alan Turing pioneered artificial intelligence during his time at the university (1). Therefore, to accommodate such requests, the school holds various events such as Open Day, Visit Day and Education Fair for people to learn about current research, programs as well as its facilities.

One of the most commonly used methods to measure the success of the events is by getting feedback from the visitors (2). Currently, the school administrative staff will carry out surveys by distributing survey forms. Visitors and prospective students will rate their satisfaction level on various aspects of the event by filling the hardcopy forms. These forms are then collected and transcribed into digital format. It is desirable to store the data in digital format, as it can be analysed more efficiently. However, the process of transcribing and translating hardcopy survey form into digital format is tedious and normally time-consuming. This process can sometimes take a couple of days just to process the hardcopy forms from a one-day event.

Figure 1-1 below shows summary of the existing survey process.

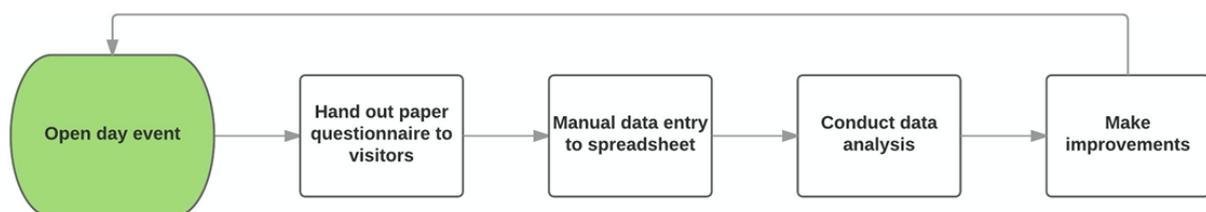


Figure 1-1: Existing survey process

In addition, only basic data analysis can be performed as the data is not systematically structured and lacking data analysis tools. Spreadsheet was used to perform basic data analysis such as percentage of response by questions. Unfortunately, this does not utilize the full potential of information that can be derived from the data (e.g. forecasting, trend, data filtering). Therefore, a more efficient way of conducting and analysing the survey data is urgently needed. It is crucial for the feedback, ideas, suggestions and comments collected in these events to be processed efficiently and intelligently, so that the school will be able to make improvements that can be best suited to the expectations of the visitors.

Attempting to solve these issues, this project developed a systematic system consisting of a mobile app and a web application. Cross-platform approach was used in developing the mobile app which resulted with a hybrid mobile app. The mobile app is primarily used for conducting surveys. On the other hand, the web application was developed to support extensive data analysis and manage the overall survey operations. The whole system was developed to be virtually 'maintenance-free' that allows for long term usage without the need for technical modification. Evaluation and testing was conducted on multiple devices with various form factors, screen resolutions and hardware specifications to ensure that the system is flexible and robust, and can deliver its intended objectives.

Aside from the system development, the project also involves the development of survey questions that can capture the broad aspect of visitor satisfaction factors. This includes research into the field of social sciences and humanities in order to understand the critical factors of visitors and prospective students' satisfaction. In a study conducted by (3), among the main factors considered by prospective students in choosing their place of study include the course content, academic reputation and the quality of academic facilities. Events such as Open Day and Education Fair serve as a platform to highlight what the school and university has to offer. Therefore, questions asked in the survey should be able to effectively measure the success of such events and contain questions that can lead respondents to further provide valuable suggestions and comments.

As the School of Computer Science draws a myriad of prospective students and visitors, it is crucial for events such as an Open Day or Education Fair to be organized effectively. The

events will create the first impression for visitors and prospective students and help shape their decision whether to study at the university. By utilizing the survey system developed in this project, feedback from visitors can be captured systematically and improvements can be made for future events. Ultimately, this will help to enhance the image of the school as well as the University of Manchester.

1.1. Aim and Objectives

This project aims to critically assess the survey requirements in the School of Computer Science in order to design and implement a suitable tool for data capturing and analysis, which could contribute to overall improvement of similar events in future.

The objectives of this project are:

- To critically assess various platforms and methods in data capturing and analysis;
- To develop a data capturing and analysis tool that can simplify data collection and analysis process; and
- To evaluate the tool developed and measure its effectiveness.

To achieve these objectives, the following goals have been set:

- Present extensive literature reviews and best practices in conducting surveys;
- The system should be designed for practical use in any of the events organized;
- Survey data collection can be easily performed and integrated with the current practice;
- Simple and prompt data retrieval;
- Collected data are relevant, standardized and validated; and
- The data is presented in such a way that makes it easy to interpret.

These goals put more focus on the usability design of the system and consider the nature of users, the tasks that they would need to perform, and the environmental setting where the system will operate. Usability involves the art of creating applications where users can perform their tasks more efficiently compared to the manual system (4). This project

intends to present a system that can efficiently perform the survey and data analysis which will overcome the existing problems as stated in the introduction and at the same time have a high level of usability.

1.2. Deliverables

The project deliverables are:

- **A fully functional CSMCR survey system**

The system consists of a web application and a mobile app. The web application is capable to manage the survey questions (e.g. add, edit, delete, view) and generate data analysis (e.g. frequency and contingency table). Alternatively, respondents can also choose to answer the survey from their device by logging into the web application.

The mobile app is primarily where the survey respondents will answer the surveys. Survey mode function on the app allows for a new set of questionnaire to be loaded automatically once the survey's response has been submitted (e.g. the tablet can be passed around during talks in lecture hall without needing the survey staff to monitor closely for every respondent). Survey responses are stored locally on the device and uploaded to the server when the network connection is available. Survey questions that are stored on the device can be updated by a simple sync to the web server.

- **Evaluation report for CSMCR Survey System**

This report outlines the functional and technical evaluation that was performed on CSMCR Survey System. In addition, security testing was also conducted and areas of vulnerabilities were identified and fixed. The overall evaluation measures whether the system meets its intended objectives.

1.3. Structure of the Dissertation

The aim of this dissertation is to provide comprehensive details on the development of CSMCR Survey System including the background research, method, design, implementation, evaluation and the overall conclusion. Below are the content summary for each chapter:

- **Chapter 2 – Background**

Background research on software development including suitable technologies, methods of conducting surveys, software development strategies and methodologies are discussed in detail. Platform comparison between mobile operating systems and the various cross-platform mobile apps developments approach are also described. Background studies and literature reviews helped in drawing subtle and comprehensive development plans.

- **Chapter 3 – Method**

This chapter illustrates the methods used for system development. This includes gathering project requirements, development process, data analysis and managing project risks.

- **Chapter 4 – Design**

This chapter will describe the project design, starting with survey questionnaires, fitting requirements into the system and overall design for mobile and web applications. The overall system architecture is presented in detail.

- **Chapter 5 – Implementation**

This chapter details the implementation process of developing the system. User Interface (UI) snapshots and source code snippets are also included for better understanding and visualization.

- **Chapter 6 – Evaluation**

Provide details on how the survey system were tested and evaluated. This evaluation stage consists of technical and functional evaluation, as well as system security and measure of effectiveness.

- **Chapter 7 – Conclusion**

This chapter offers conclusion, highlights the project achievements, issues that were encountered during the project and potential further works that could be carried out to enhance the system.

Chapter 2: Background

This chapter will highlight and discuss the feasibility study in developing the system, various modes of data collection available, survey instruments and various technical options that are relevant to the development of this project.

2.1. Feasibility Study

A feasibility study was conducted to objectively evaluate the project idea and compare it with solutions that already exist in the market. The overall concept of a system for collecting and managing survey questionnaires may not appear as something new since there are various readily available templates and off-the-shelf systems. For example, SurveyMonkey (Figure 2-1) provides survey templates for creating customized questionnaires and gathering respondents' feedback.

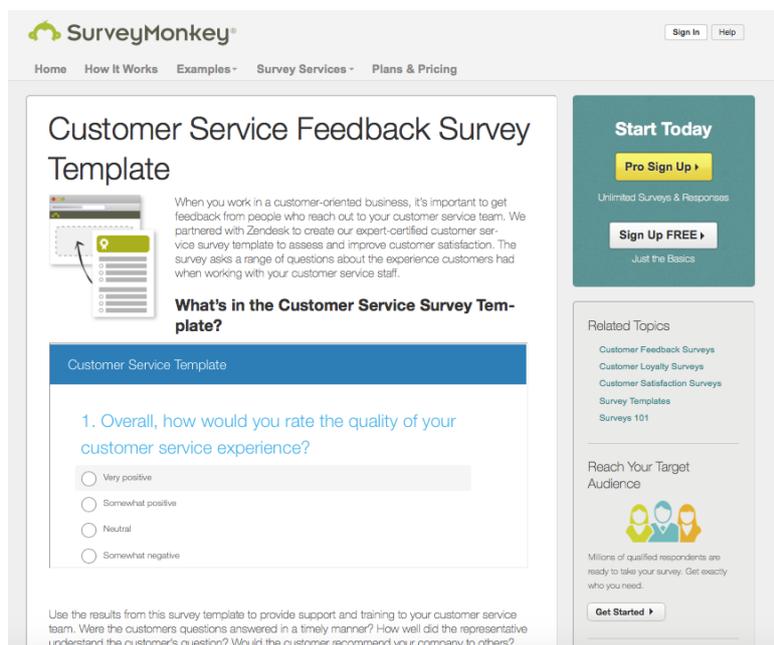


Figure 2-1: Survey Monkey Template

Therefore, to ensure the project does not replicate existing systems and will be practical as well as desirable, a comparison between the features of existing systems and requirements of the proposed system was made. Table 2-1 below explains the differences between major survey systems and the proposed system.

Table 2-1: Comparison between Free Survey and Proposed Survey System

Name/Features	SurveyMonkey	Google Forms	Survey Gizmo	Survey Planet	Proposed System
Survey Question Types	15 question types. Free version limited to only 10 questions per survey	9 question types	Over 25 question types	10 question types	Unlimited question types can be developed. No limit to the number of questions per survey
Connectivity	Require Internet connectivity	Require Internet connectivity	Require Internet connectivity	Require Internet connectivity	Can be done offline
Input method	Web browser and mobile app	Web browser	Web browser	Web browser	Mobile apps and web browser
Analysis	Frequency table and basic graphs. Data filtering and cross-tabulation only for paid subscription	Frequency table and basic graphs. Data filtering is not supported	Frequency table and basic graphs. Data filtering and cross-tabulation only for paid subscription	Frequency table and basic graphs. Data filtering and cross-tabulation only for paid subscription	Frequency table, cross-tabulation and graphs. Allow for data filtering.
No. of response	First 100 is free	Unlimited	Unlimited	Unlimited	Unlimited
Cost	Free version available. Subscription ranging from £26 - £85 per	Free. Combined with Google Apps for work for £3.30 -	Free version available. Subscription ranging from \$25 - \$75 per	Free version available. Subscription cost \$20 per month	Free

	month	£6.60 per month	month		
Third Party Integration	Possible only through established partner platform/applications	API available to use	Only for paid subscription	Embedded/ Integration through popular social media	Possible using the source code
Customization	Very limited and mostly available only to paid subscribers	Limited to only available templates	Not available for free version	Choose from 10 available templates	Can be customized as needed
Support	Through email. Quick support only for paid subscription	Search from the knowledgebase	Free version support only through online documentation	Free version support only through online documentation	Configurations and troubleshooting can be done by the school itself

One of the main arguments for developing a customized system is that respondents' answers can be validated through the use of built-in validation rules. The system can automatically check respondents' answers and prompt for verification as soon as it detects any inconsistencies in the answers. This can motivate respondents to read the questions carefully and give some thoughts before answering them. Each of the survey response will be identified with a unique ID number that can help in organizing and detecting any duplicate and fraudulent entries. This will result in higher overall response quality compared to using system without these functionalities.

In addition, critical system requirements such as ability to work offline and flexibility in data analysis can only be achieved through the proposed system. The school will have the flexibility of customizing the system and full control on how the system is managed and run. This will also help propel the school forward as it promotes better system development as well as appreciates the work submitted by the students.

Although more advanced functionalities are available by subscribing to major online survey systems, the subscription cost in the long run would be relatively high. The school would also have to depend solely on the functionalities offered by the service providers. Therefore, based on the feasibility study, it can be concluded that the proposed system is more feasible and practical compared to any existing systems available in the market.

2.2. Modes of Data collection

Qualitative research is normally used when data collection involves gathering people's views. One of the methods under this approach involves surveys, which can be viewed as a system to collect and gather usable information about people that could describe, compare, or explain their understanding, feelings, and behavior (2). This system comprises interdependent activities which involve defining survey objectives, establishing applicable questions and answering choices, respondents' selections, and preparing reliable and valid survey means. To ensure its objectives are met, the questions and answers should be specified and meticulously organized.

In the context of this project, surveys are conducted to gauge the satisfaction level for the events held, to measure the success of the events and gain valuable feedback and suggestions. Questionnaires are used to elicit objective and subjective information from individuals about their thoughts and perception (5). In order to achieve all these, the objectives for holding the event must first be established. They will be reflected in a set of questionnaires that will lead to the established objectives.

2.3. Survey Instruments

Self-administered pen and paper questionnaires have been traditionally used to collect survey data including at the School of Computer Science. Presently, studies have shown that with the advent of mobile technology, electronic instruments such as tablet computers, iPads, and laptops are increasingly becoming the alternative choice in collecting surveys data (6), (7), (8), (9). These devices are used in various places and diverse settings such as remote locations, schools, primary care clinics, and universities (7), (10), (11), (12), (13), (14). For the purpose of surveys during events organized by the school, the use of electronic instruments such as tablets might be a better and more appropriate choice.

Electronic survey instruments are useful to simplify the process, starting from development of questionnaires up to the data analysis. They can be classified into two types, namely those that rely on network connection to run the questionnaire and those that can run locally (15). For the former, the questionnaire is developed on a computer and transferred onto electronic devices or uploaded to a web application. The latter typically requires portable devices, such as tablets, that can store survey questions locally and users may complete the surveys offline. This option is more favorable when administering surveys in places with unreliable network connection.

One of the advantages in choosing electronic instruments is that they can increase usability and improve the accuracy of responses (15). In cases where the survey does not require every question to be answered, respondents may use the skip option and move to the other

relevant portions of the survey. For example, visitors can be grouped into categories such as prospective students or family members and based on the grouping, only the relevant questions would be asked.

In addition, survey questions can be intelligently developed to ensure valid and consistent answers. Once the survey is completed, data can be uploaded automatically into the database, thereby eliminating the need for manual data entry. This can eliminate possible inaccuracies with multi data-handling steps (7), (14). Studies also show that electronic questionnaires reduce the problem of missing data and errors in responses (16), (17). Therefore, based on previous studies, it can be concluded that switching from pen and paper to electronic instruments will improve the overall survey data quality and responses.

2.4. Mobile Web Vs Mobile App

One of the most common dilemmas when developing for mobile users is whether to choose between a mobile web and mobile app. Often, the answer depends on the main purpose of an application. For example, for this project, the main purpose is for data collection whereby respondents will have direct access to the survey system. Therefore, usability and convenience will be the key factors in choosing between the two approaches.

In general, the main advantage of mobile apps is that it is developed in its native programming language. This will allow the app access to and utilizes the device memory and processing power that makes the application run smoothly (18). In addition, as the app is installed on the device itself, it may run sans network or internet connection. This is a great advantage for applications that needed to be worked on offline as data can be stored locally and uploaded or downloaded when connection is available.

Mobile apps can also utilize some of the device specific features such as its camera, accelerometer and GPS. This will help enhance the functionalities and capabilities of the developed application. In addition, the design and UI can benefit from the dedicated framework of UI features. Developers may use the platform specific SDK such as

placeholders for adding icons, backgrounds and splash screens, in addition to access to the full library of UI elements. This will help save development time as well as ensuring the app and design will work seamlessly with the device operating system.

On the contrary, mobile web is usually considered as a backup solution for mobile developers (18). One of the key advantages of mobile web is that almost every mobile device already has a browser installed and can use the application straight away. This eliminates the need to download and install additional app. Moreover, updates or changes can be done directly on the server. This would ensure that each time a user accesses the application, he/she will use the latest version.

Several work-arounds have been established for mobile web to allow its functionality to be at par with mobile app. For example, mobile web utilizing HTML5 can now use Application Programming Interface (API) that allows access to the device specific feature, such as customized keyboards. Moreover, it also allows access to load page components or elements into the phone's local memory, which in turn enables a user to read content offline without having to download the mobile web content. This is considered a hybrid approach and will be explained in the following sub-chapter.

2.5. Software Development Strategies

In any mobile development, the aim is to target as many users as possible. One of the ways to achieve this is by enabling the app to work on various platforms; hence the concept of cross-platform solutions. Previously, in order for the app to support various platforms, developers either developed an app for a specific platform at a given time or a team of developers were divided into smaller teams and each team would develop the app for specific platform (19). Although this approach enabled the app to run on cross-platform, it required more developers, was time consuming and often costly.

In solving the issue, mobile development with cross-platform solutions has been proposed. The idea behind this concept is to develop an app that can run concurrently on different

platforms. Various cross-platform solutions for mobile development are still undergoing research and development phase. Figure 2-2 below compares the concepts between the traditional mobile development and the proposed cross-platform solutions (20).

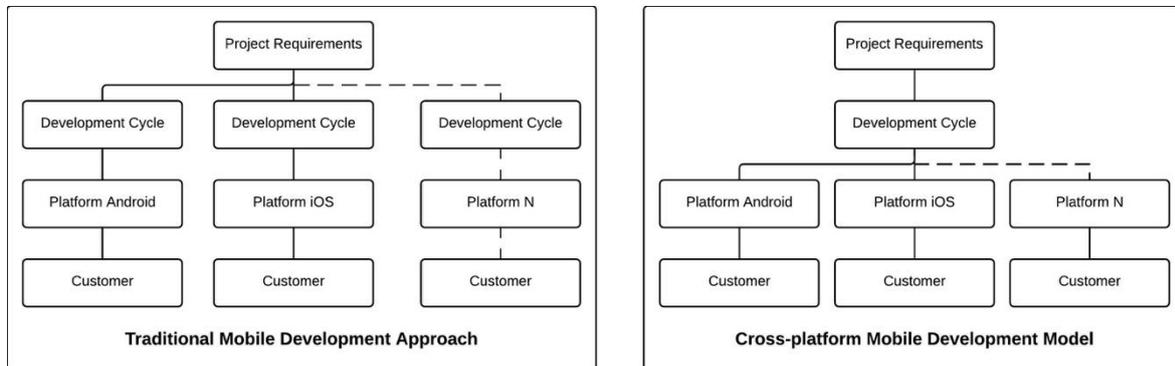


Figure 2-2: Existing approach and proposed cross-platform development model

Traditional mobile development is platform-dependent where apps are written for specific platform. Thus, for the app to run on a different platform, the whole development cycle has to start again. On the other hand, in the proposed cross-platform development model as shown in Figure 2-2, the solutions outspread the software development lifecycle by allowing the app to be written once and ported to various platforms. Common approaches in the proposed cross-platform mobile apps development are Cross-Compilation approach, Virtual Machine approach, Web-Based approach, Component-Based approach, Merged approach, and Cloud-Based approach (19). Recent approach includes Component-Based approach, Cloud-Based approach, and Merged approach. In addition, mobile apps can also be categorised according to types, namely web app, native app and hybrid app (21).

Compilation is one of the cross-platform approaches in mobile development which consists of two sub-approaches, namely the cross-compiler and trans-compiler (22). Compiler transforms source code that was written in the source language (usually high-level programming languages) into the desired language (usually lower-level language such as the assembly language or machine code) and into an executable program.

On the other hand, the virtual machine approach intends for mobile app to be developed with a platform-independent language. The most common virtual machine is Java Virtual

Machine (JVM) which comes with its own all-inclusive hardware architecture including the central processing unit, register, stack and corresponding instruction system (23). Via this approach, the developed app runs on the dedicated virtual machine that is installed on various platforms.

The web-based approach makes use of the web programming language and not platform-dependent programming languages to write an application that is then built into native applications. They are called Hybrid apps. Hybrid apps are developed with web and mobile technologies. It is installed, launched and used just like native apps, and it can access a range of device APIs. The Web-Based tools mainly use the web technologies like HTML, HTML5, Javascript, and CSS to develop mobile apps that can run on different platforms. To allow access to the native platform features, the web-based applications are packaged in a small native container, which have access to the native APIs that provides access to the hardware such as the camera, storage space and accelerometer. They possess most of the benefits of web app and native app.

2.6. Software Development Methodology

The Software Development Lifecycle (SDLC) provides guidance for successful software development. In recent years, the industry has gradually shifted from traditional waterfall model to Agile method. An annual survey conducted by (24) revealed that 95% of respondent organizations uses Agile in their software development projects. The main reason is to accelerate product delivery, enhance the capability to manage changing priorities and increase productivity (24).

For projects where development is done at a single site involving small groups, evidence has shown that Agile can improve both productivity and quality (25). Agile aims to reduce excessive and labour-intensive activities when conducting implementation and focuses more on quick responses to various and ever-changing project requirements (26). The concept of Agile is in line with requirements for this project which are subjective and may need to be changed frequently. However, the challenge is to determine which kind of Agile

to use as there are various kinds and Agile can simply be defined as any method that adheres to the Agile manifesto (27). Table 2-2 below shows a summary of various established Agile methods.

Table 2-2: Agile Development Methods

Agile method	Description
Extreme Programming (XP)	Aim to assist developers working in small teams that have to deal with ambiguous and frequently changed requirements. Considered as a lightweight methodology, the focus is on cost reduction, unit test done before and after coding, pair programming, regular integration with system, simple designs and regular releases (28).
Scrum	Provide process framework to manage product development that are under complex problems or circumstances (29). Scrum make use of empiricism theory that focuses on knowledge, experience and known information (29). Products are developed by cross-functional teams by utilizing iterative and incremental approaches (30).
Lean Software Development (LSD)	Originally used in manufacturing, it is adapted in software development settings when Agile gained popularity (31). The main idea is to deliver only the features requested by the customers and other activities not adding values for the customer requirements are considered as a waste (32), (33).
Kanban	Originally established as a lean tool in managing manufacturing activities (32). The main principles are to visualize workflow, limit the amount of work-in-progress and estimate time needed to complete tasks.
Feature-Driven Development (FDD)	FDD is considered as Agile and Adaptive approach that is only used during the design and implementation stage of SDLC (34). The method emphasizes on process quality and makes frequent deliveries. Previous study claims that FDD is better suited for development of critical systems as compared to other Agile methods.
Dynamic Systems Development	Defined as open-framework for the Rapid Application Development. The main ideas are that project flow should be determined by resource availability, prefix timing in establishing features and the amount of

Method	features released should not be fixed as it is based on productivity (34).
Adaptive software development (ASD)	ASD intended to solve problems and issues that are usually in large and complex systems development settings by providing guidance that can prevent project failure while still maintaining flexibility that allows for creativity (34). Main ideas in ASD are to conduct iterative and incremental approach for the development of features and prototypes.
Rational Unified Process (RUP)	RUP is an iterative process created to complement UML in object-oriented system development by utilizing UML features like the use cases. The four phases of RUP lifecycle are inception, elaboration, construction and transition (34).

Choosing the right Agile methodology is crucial for a project to be completed successfully. Unfortunately, in a comparative review done by (35), empirical evidence to show the effectiveness of one Agile method over another is scarce. A systematic review study done on 1,996 Agile developments using its various methods found weak evidence suggesting the advantages and limitations for each of the Agile methods used (36). Therefore, deciding on which method to choose for this project will have to be based on the suitability of Agile's specific method principles as well as the project requirements and settings.

2.7. Usability

Usability is defined by the International Organization for Standardization as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" (37). Towards the project completion, usability evaluation was conducted to identify any issues pertaining to the design and UI usability and changes were made for improvement.

In a typical usability study, potential system users will be asked to perform the task using the system while they verbally comment on their experience (38). The developer will make observation and along with the users' feedback on performance measures of accuracy, efficiency, and subjective satisfaction, the developer can better understand the issues that

users have with the product and can suggest improvements to the design. In addition, the five principles in system acceptability are useful in evaluating the system. The principles consist of (39):

- **Retention:** How easy it is to memorise the steps on using the system
- **Learnability:** The time taken for users to be familiar and acquainted with the system
- **Efficiency:** time taken to accomplish tasks using the system
- **User satisfaction:** user feedback and general opinion of the system
- **Errors:** Number of mistakes or errors made by user

In general, there are two types of evaluating usability, which are qualitative (formative) and quantitative (summative) (40). Qualitative is typically done at the early stage of development or before a new release. Qualitative data is typically used to characterize findings and describe properties and behaviours while quantitative research generally collects more data and they are normally used for statistical analysis.

Methods used to evaluate usability plays a crucial role to ensure successful evaluation and therefore the evaluator would normally pick which method to use (41). It is important that the evaluator obtains as much information as possible to ensure that the right decisions are made (42), (43). Based on previous studies (43), (44), (45), these four principles are recommended in guiding the method selection process:

- **Identify types of information required** – By considering the types of analysis that will be performed beforehand, it will provide some guidance on the types of data that should be collected. For example, user interaction times and number of errors made are beneficial to have for analysis. In addition, a mixture between objective and subjective methods may generate a sensible assessment of usability;
- **Stage of evaluation** – Throughout the different design process and stage, evaluation should be conducted and appropriate methods should be selected based on their suitability at that particular stage. By doing so, any usability issues at each design

stage can be fixed before progressing to the next stage which can make it more difficult and expensive to fix;

- **Resources required** – Balanced between availability of resource requirements and methods' used should be achieved; and
- **Involving people in method selections** – The number and types of people needed for conducting each method will need to be identified carefully as it will influence the suitability and outcome of the evaluation. Typically, evaluators will use methods and their knowledge and experience in making predictions on the usability of a system (46). Involving people is also useful in evaluating the subjective aspects of usability.

The four principles as described above are closely connected to each other and appropriate trade-offs should be made in identifying appropriate methods in line with this guidance.

Chapter 3: Method

This chapter will discuss the methods used to gather the project requirements, project development, data analysis, evaluation as well as factors to consider when choosing the methods. In addition, the project risks and mitigation plan are also discussed.

3.1. Defining Requirements

A strong foundation for embarking on a successful system development is created through defining requirements. They involve gathering expectations and documenting what is to be built. It is important to understand the environment, constraints and tools used by the school administrative staff in conducting the survey. The primary goal is to define the set of requirements that will be used to design, build, and implement the project solutions within an agreed-upon project scope and timeline. The process starts by defining project requirements, followed by defining data, functional and technical requirements as explained below:

- **Defining project requirements:** Project requirements were obtained from system owner and prospective users. Gathering requirements was done primarily through interviews. Project requirements are typically associated with process flow and users involved. This association identifies who and what will be affected by the project solution and might potentially be expanded to include stakeholders. Possible integration with other relevant systems that exist in the school were identified and considered in this stage.
- **Defining data:** Identify data sources and other dependencies needed to build the project solution. Defining data requirements involve looking at potential data sources to identify the types and kind of information that will be collected. This may also influence the quality of data that will be captured by the proposed solution and potential analysis that the system can come out with using the data.

- **Functional requirements:** Generic requirements initially provided with project descriptions were detailed out. Among the tools used in this process were use cases, analytical process workflow and user interaction.
- **Technical requirements:** Specification on items such as new technology directions (e.g. school is moving towards cloud data, software preference), IT policies, hardware and software restrictions are collected so that the proposed system can be developed to meet these requirements.

The requirements will be gathered and consolidated into a cohesive project requirement. The consolidation process will spot overlapping, redundant/conflicting requirements and identify gaps and missing elements based on dependencies that might not have been seemingly clear before the process. These issues will be resolved or included as part of the project requirements.

3.2. Agile Development

As presented in the background chapter of this report, the project adopted the Agile methodology as it fits with the project requirements and environment setting. After careful consideration between Scrum and XP, the latter is chosen as the specific agile development methodology. Both methods are quite similar in principle as some concepts in Scrum are actually “borrowed” from the XP (27). Scrum specifies more on team interactions between themselves and with the organization. Since this project does not involve much team effort, XP would therefore be a more suitable choice. The focus of XP is on code writing and system testing, which are in line with the project focus.

Table 3-1 below shows the principle of XP and how it can fit into this project setting.

Table 3-1: Comparison of XP Principle and CSMCR Survey Project Setting

Principle	XP	CSMCR Survey Project Setting
Team size	Intended for small teams, typically between two to ten members, and have to work on regular and unfamiliar assignments.	Small team consisting of single programmer and project supervisor/owner.
Communication	Regular communication between team members.	Project owner is directly involved and consistent communication with prospective system users is possible.
Change in requirements	Allows for frequent change in requirements. Quick response to any changes before the project completion.	Require frequent change in requirements to best suit an ever changing situation.
Iterations	Short and frequent iterations.	Project split into modules and released as soon as each module is completed.
Testing	Continuous testing.	Conduct test at every module release with possibility of automating the test.

XP is seen as a flexible Agile method that emphasizes close links between proposal stage and project implementation. The fundamental activities are planning, managing, designing, implementation and testing. The project adopted the set of practices as originally introduced by Beck in 1999 (47) to suit the project setting as follows:

Planning game – The project started with writing user stories, explaining functionalities that the user wants in the system. Estimation for the development time for each story was made. Stories were grouped into releases or iterations and for each iteration, they were limited to only the scope in these stories.

Small releases – New operational functionality was released in every two or three weeks. The iteration length was planned to be at maximum of one month for each release.

Metaphor – Explanation on the project uses as minimal jargon as possible for the non-technical members involved in the project. Common and simple metaphors were used when explaining the details to give better understanding across the board.

Simple design – Any code that could potentially become complex were broken into small units and simplified. The code was made easy to understand by incorporating simplicity and avoiding redundancy or duplicates in codes.

Tests – Unit test framework is created to help in planning for testing. Ideally, unit tests are created before the functional coding process itself. Automated unit test will be made and the entire test is ensured to run correctly.

Refactoring – Refactoring were done consistently throughout the process to ensure that the design is simple and unnecessary clutter and complexity is avoided. This will also help to keep the code clean and concise so that it will be easier to understand, modify and extended if needed.

Pair programming – This practice was not applicable as there is only a single programmer working on this project.

Continuous integration – New lines of codes were added regularly to the existing code to ensure that the entire system work as expected. For issues or problems that arise, immediate fix in coding were made or the changes made were reversed.

Collective ownership – In principle, all programmers working on the system are allowed to make changes to any part of the system, as there is no division of ownerships in codes. However, in the context of this project, as there is only one programmer, this practice was not applicable.

On-site customer – Any ad-hoc or clarifying questions were asked directly to the project supervisor as both are from the same school.

40-hour week – Constant overtime is usually accompanied with stress and is a sign of problems in the project that lead to rushing and low-quality solutions. Therefore, this project was planned with reasonable amount of workload to ensure that the quality of output is sustained at the high level and all functionalities planned for are realistic and can be completed within the allocated time.

Open workspace – The practice suggested that the entire development team should have a big room to facilitate open and efficient communication between members. As this project does not involve team development, this principle was not applicable.

Just rules – In this practice, everyone should follow and agree on the rules set. Change can be made provided that everyone is aware of its consequences. Again, as this project does not involve team development, this principle was not applicable.

In developing the mobile apps, the project followed the Mobile App Development Lifecycle, which comprises the evaluation of the app idea, designing the UI, programming the app using languages of targeted platform, testing the app on various devices, and publishing the app on the targeted platform store (19). Mobile app development can be considered as a special case of software development due to different aspects than the normal software development. This includes short development lifecycle, mobile device functions, mobility, mobile device specifications such as screen dimensions, navigation and app UI design, mobile security and user privacy (19).

3.3. Data Analysis

There are various ways for the data collected from the surveys to be analysed to provide the information required. However, lack of analysis process may produce undesirable output and jeopardise other potential beneficial analysis methods due to aggregation or

unavailability of data. Therefore, the types of data collected should carefully consider the type of data analysis needed.

Data source for this project primarily comes from the questionnaires. There are two types of questions asked in the survey questionnaires, which are open-ended and closed-ended questions. Open-ended questions are typically exploratory in nature and provide qualitative data, providing the ability to gain better insight on the feedback and opinions. On the other hand, closed-ended questions are typically conclusive in nature and intended to be easily quantifiable. There are four types of closed-ended questions, namely categorical/nominal data, interval data, ordinal data and ratio data.

For the purpose of this project, closed-ended question only use categorical/nominal data and interval data and they are described as below:

Categorical/nominal data – typically respondent will be asked to select from the choices provided to provide the answer or state their preference. For example:

What was your main mode of transport to get here? (Please tick the ones that apply)

Train	<input type="checkbox"/>	Bus/Coach	<input type="checkbox"/>	Tram	<input type="checkbox"/>
Car	<input type="checkbox"/>	Walking	<input type="checkbox"/>	Plane	<input type="checkbox"/>

Interval data – This type of question provides measurement and information about the order by utilizing intervals. For example, in the questionnaires for this project, Likert scale is used to rate the experience of respondent.

Once the data is collected and stored, it will be used in the data analysis stage. In general, there are six types of data analyses ranging from the least complex to the most complex, and they vary in terms of the knowledge involved, costs, and time taken. The six types are descriptive, inferential, exploratory, causal, predictive and mechanistic (48).

Table 3-2 below provides a summary for each type of data analyses.

Table 3-2: Types of Data Analysis

Data Analysis Types	Description
Descriptive	Usually the first type of analysis performed on a dataset, especially for data in large volume. Quantitatively describes the main features for a set of data. This type of analysis generally requires the least amount of effort to conduct.
Inferential	Analysis on the sample set of population with the aim of testing theories about the nature of the population in general.
Exploratory	A type of data analysis to find previously unknown relationship that lies within/between datasets. It is suitable for discovering links or connections as well as for defining future studies.
Causal	Conduct analysis of data to find the effect of changing one variable to the other. The implementation of this study would typically require randomized studies.
Predictive	Conduct analysis on current and historical data to make predictions about something in the future.
Mechanistic	Arguably requiring the most amount of effort, this analysis tries to understand the particular changes in the variables that can cause changes or difference in the other variables for the individual object.

Based on project requirements and suitability, descriptive type of analysis was most suitable for quantitative data (closed-ended question). However, qualitative data obtained from open-ended questions will need to be coded and analysed manually. Common answers found in this type of questions will be proposed as the candidate choice of answers for the new closed-ended questions set. Having most questions in closed-ended type will simplify the overall data analysis process as the process can be made automated.

3.4. Risk management

As with any software developments, given their abstraction and diverse nature, they are often exposed to unique challenges and risks. This project is no exception. Among the critical factors that have been identified in this project are the project requirements, project complexity and project planning and control. These factors and their risks should be managed systematically.

Table 3-3 below provides the details for the project risks, critical elements and mitigation plan.

Table 3-3: Project Risks, Critical Elements and Mitigation Plan

Project Risks	Critical Elements	Mitigation Plan
Project requirements	Ever changing requirements/ Requirements inadequately identified	Agile method employed in this project anticipate this and frequent releases will make it possible to accommodate changing requirements.
	Unclear requirements	Prioritize face-to-face meeting to encourage clear communication.
	Wrong system requirements	Frequent and small releases will make it possible to detect and change as early as possible.
	User reject system prototype	Regular delivery of system prototypes for immediate feedback and input.
Project complexity	Involved with high technical complexity	Alternative methods will be explored or experts in related fields will be consulted.
	Use of technology incompatible with existing school infrastructure	Potential technologies to be used are identified during requirements study and matched with existing school infrastructure.
	Advanced programming	Seek for tutorials and help from experts to

	skills needed	reduce the learning curve. Avoid building complex code and keep it simple, which is aligned with Agile principles.
Planning and control	Poor project planning	Project supervisor will monitor the project closely. Project plan was made with realistic duration and project buffer was added for unforeseen circumstances.
	Inadequate resource estimation	Maximize the use of existing hardware and software. Loan from other schools should they are needed.

As the project risks have been identified at the earlier stage of this project, any issues that may lead to the specified risk were given special attention and actions were taken to avoid them. The risks list provided important consideration factors during the planning and implementation phase of the project. Failure to comprehend and administer these solutions may have resulted in severe delays in the project delivery or may even lead to project failure. Therefore, understanding the variations in project scope and requirements will provide the basis to understand the risks better and enable development of effective risk mitigation strategies.

Chapter 4: Design

The application development is based on the Agile approach of the software development lifecycle. This chapter discusses the survey questionnaire, project requirements and the overall system design.

4.1. Survey Questionnaire

Questionnaire is a data collection method where respondents are requested to provide their opinions and feedback. Although a rather challenging task, it is one of the most popular research instruments that can yield high quality and usable data (2). Questionnaires should be designed to be “respondent friendly”, and “easy for respondents to complete, avoid confusion about what or how to answer and result in respondents feeling neutral or positive” (49). Therefore, by having clear instructions, systematic layout of questions and answers, suitable font size and type and suitable number of questions may help to achieve this.

In designing the survey questionnaire, it is recommended that the questions to be short and focused, possibly with no more than 12 words (50). The structure of questionnaire will determine the data analysis capability, as the more structured the questionnaire are, the easier it will be to interpret the result as the data is quantitative as opposed to qualitative. In addition, sequencing of questions by starting with easy and non-sensitive questions will help to encourage better feedback from respondents. The order of questions is also an important as biasness in answers may likely occur if responses are ordered in a consistent manner (51).

In designing the questionnaire, various question types exist. In summary, they are:

- **Open questions** – responses are expected in words, where typically qualitative information is needed. When conducting data analysis, they are usually converted and generalized into groups, allowing for the data to be quantitatively measured;

- **Closed questions** – respondents are offered a choice of answers;
- **Quantity** – responses are expected to be in numbers. For example, how many times have you attended undergraduate Open Day in the past year?;
- **List** – respondents can choose more than one response, in which the responses are not commonly exclusive;
- **Ranking/scales** – respondents may choose from a ranked scale of options to indicate their degree of agreement or disagreement with regards to the provided statement. Depending on the scale used, it may produce ordinal, interval/ratio or nominal data. For example, Likert scale will produce ordinal data, where a clear order can be established (e.g. strongly agree, agree); and
- **Grids** – not commonly used, it is only useful when measuring more than one dimension in the question. Alternatively, the question can be divided into two or more separate questions.

Based on the project goals, appropriate questionnaire that may help to gather opinion and feedback from the visitors was developed. There are 3 types of visitors whom mainly attend the open days, namely prospective students, parent/family members of the prospective students and academicians/researchers. Therefore, the questionnaire developed must take into account the different types of visitors so that only relevant questions will be asked. In addition, past experiences in organizing open day events and literature reviews were useful in developing the questionnaire.

The questionnaire comprised predominantly of quantitative questions using five point Likert scale and close-ended question. The use of quantitative questions may simplify the data analysis process as it can be automated and results are represented in figures. On the other hand, a minimal number of qualitative questions are also included to allow respondents to express their opinion and feedback freely. As for now, qualitative answers need to be analyzed manually from the system. Table 4-1 below shows the summary of questionnaire that has been developed for this project.

Table 4-1: Survey Questionnaire Summary

No.	Question	Question Type	Answer option(s)	Question rationale/Example use of data
1.	Type of visitor	Closed	<ul style="list-style-type: none"> • Prospective student • Family member of prospective student • Academician/researcher • Other 	Identify specific target audience in talks – e.g. parents vs prospective students.
2.	What was your main mode of transport to get here?	Closed	<ul style="list-style-type: none"> • Car • Bus/coach • Train • Tram • Plane • Walking 	Student ambassador can be stationed at popular arrival spot. Direction signage can be put at strategic places.
3.	Which course have you applied to?	Closed	<ul style="list-style-type: none"> • Artificial Intelligence • Computer Science • Computer Systems Engineering • Computer Science with Business and Management • Computer Science and 	Emphasize more detail on popular courses during talks.

			<p>Mathematics</p> <ul style="list-style-type: none"> • Computer Science (Human Computer Interaction) • Software Engineering • Foundation Year • Not Applicable 	
4.	How did you hear about this school?	List	<ul style="list-style-type: none"> ▪ University prospectus ▪ School prospectus ▪ Website ▪ Open Day ▪ Personal Recommendation ▪ General Reputation 	To know which medium is the most effective in reaching out.
5.	How would you rate the campus tour?	Scale	<ul style="list-style-type: none"> • Excellent • Very Good • Good • Fair • Poor • Did not good 	Measure the effectiveness of the campus tour.
6.	How would you rate the school talk?	Scale	<ul style="list-style-type: none"> • Excellent 	Measure the effectiveness of the school talk.

			<ul style="list-style-type: none"> • Very Good • Good • Fair • Poor • Did not attend 	
7.	How would you rate the lunch and refreshments?	Scale	<ul style="list-style-type: none"> • Excellent • Very Good • Good • Fair • Poor • Did not eat 	Measure the satisfaction level for lunch and refreshments served.
8.	How would you rate the school tour?	Scale	<ul style="list-style-type: none"> • Excellent • Very Good • Good • Fair • Poor • Did not go 	Measure the effectiveness of the school tour.
9.	How would you rate your interviewer?	Scale	<ul style="list-style-type: none"> • Excellent • Very Good 	Measure the effectiveness of the interviewer.

			<ul style="list-style-type: none"> • Good • Fair • Poor • Not applicable 	
10.	Did you find it useful to talk to the current students?	Scale	<ul style="list-style-type: none"> • Very useful • Useful • Somewhat useful • Need improvement • Not useful • Not applicable 	Measure the effectiveness in making current students available to talk to the prospective students.
11.	How would you rate the student project demonstration?	Scale	<ul style="list-style-type: none"> • Excellent • Very Good • Good • Fair • Poor • Did not see 	<p>Measure the effectiveness of student project demonstration.</p> <p>Know specific projects that visitors normally have more interest into.</p>
12.	Before this visit, how likely are you to accept any offer we make?	Close	<ul style="list-style-type: none"> • Definitely • Probably • Intended to reject 	To know if usually prospective students who come to the Open Day have made their decision or not.

			<ul style="list-style-type: none"> • Not applicable 	
13.	After this visit, how likely are you to accept any offer we make?	Close	<ul style="list-style-type: none"> • Definitely • Probably • Intended to reject • Not applicable 	To know if the visit has changed their decision. In data analysis, filtering options can be used to measure those who initially intended to reject and change to definitely/probably and vice versa.
14.	What is the main reason for any change of intention, if any?	Open		To know the specific reason for their change of decision.
15.	Overall, how would you rate the Open Day?	Close	<ul style="list-style-type: none"> • Excellent • Very Good • Good • Fair • Poor 	Measure the overall effectiveness of the open day.
16.	Do you have any suggestions/ideas to make the Open Day better?	Open		Gather suggestions and ideas from the visitor to make the event better.

4.2. Requirements Gathering

In Agile development, the process of gathering requirements from the customers is done iteratively (52). Initially, the general overview of system functionality was established. It does not have to cover every single detail as it is meant to serve as the starting point for the project. Later, in order to build the complete project requirements, user stories method was used. Potential users and stakeholders wrote the things that they want the system to perform. The use of user stories is almost similar to use cases or usage scenarios but user stories are not limited to only describing the UI. It highlights the project requirements in the form of necessary tasks needed to be performed by various levels of users.

Table 4-2 below provides the summary of user stories with the development sprint number and acceptance test.

Table 4-2: Summary of Sprints, User Stories and Acceptance Test

Spr. No.	User Story	Acceptance test
0	Setting up of development environment	Developer has Android studio working
		Developer has Github account set up
		Developer MySQL database set up
1	As an administrator, I want to be able to add questions to the system, so that during the survey we can use the updated survey questions.	Administrator can specify how many new questions to create
		Administrator can specify the type of answers for the questions
	As an administrator, I want to be able to log in and log out from the system, so that only authorized users can modify the questionnaires	Administrator can add users based on privilege and access level
		Administrator can specify what tasks can be performed for each access level
		Administrator can edit and remove users from the system

2	As a user, I want to answer survey questionnaires so that I can view my answers before submitting	User can view the survey questionnaires
		Users can answer various types of questions
		Users can review their answers
		Users can edit their answer
3	As an administrator, I want to be able to analyse the survey data, so that report can be prepared based on data provided from the system	Administrator able to generate frequency table based on questions selected
		Administrator able to filter data for analysis
		Administrator able to generate graphs from the data
		Administrator able to export generated analysis into separate file
4	As an administrator, I want to be able to generate trend overtime so that reports can be produced based on trend over a series of survey events	Administrator able to select date of data for analysis
		Administrator able to select criteria for data analysis
		Administrator able to generate frequencies based on selected survey date
5	As an administrator, I want to be able to download all the survey data so that I can use other software to generate analysis	Administrator able to choose the format of data export
		Administrator able to download the data into specified location
6	As an administrator, I want to be able to search for survey data based on time and date so that I can view the specific responses	Administrator able to select the desired time period
		Administrator able to view survey data based on selected time period
7	As a user, I want to answer survey questionnaires on the web so that I can use my own device	User able to view the questionnaires online
		User able to answer the questionnaires online
		Users able to review and submit their answers online
8	As an administrator, I want to be	Administrator able to download template data

<p>able to import previous survey answers into the existing system so that I can perform analysis using the new system</p>	to match the formatting with the system format
	Administrator able to upload and store data to the system
	Administrator able to view the recently uploaded data
	Administrator able to query and analyze the recently uploaded data

Based on the user stories gathered, use cases were created to further visualize the different types of users and tasks involved. Figure 4-1 below shows the use case diagram.

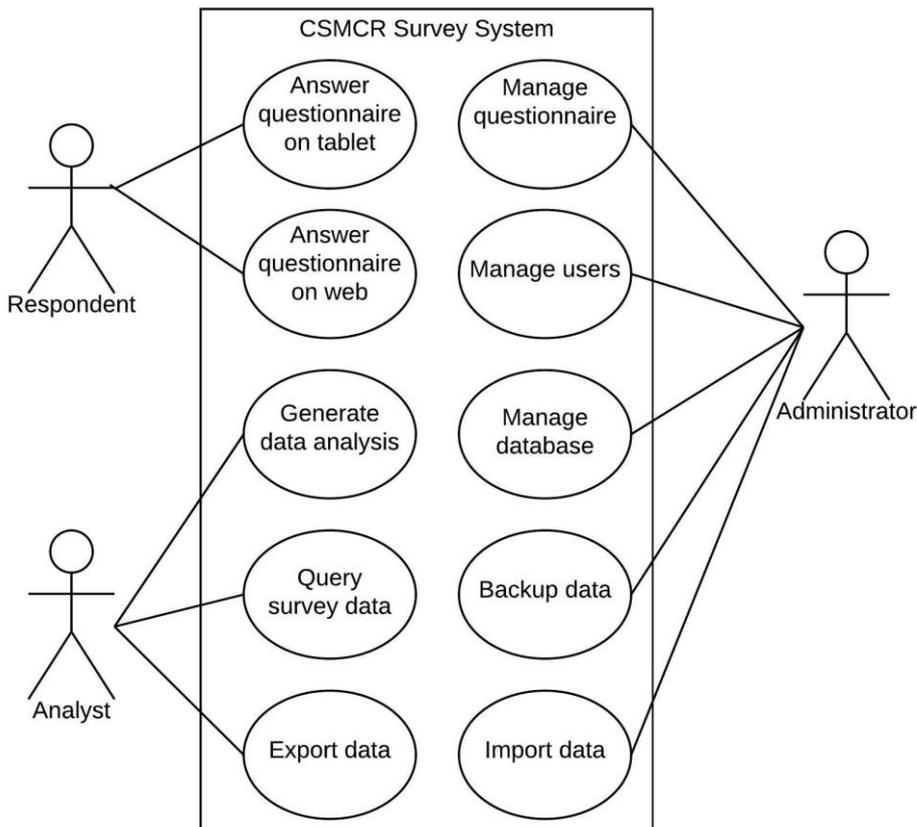


Figure 4-1: Use Case Diagram

The requirements extracted from user stories and use cases were grouped according to functional and non-functional requirements. Every requirement identified was classified on their degree of risk, complexity and priority with a rating of low, medium or high.

Table 4-3 below shows the summary of project requirements.

Table 4-3: Summary of Project Requirements

ID	Description of Requirement	Risk	Complexity	Priority
FR1	The system must be able to authenticate based on user levels.	Low	Low	High
FR2	User must be able to answer survey questionnaires using mobile apps.	Medium	Medium	High
FR3	Administrator must be able to edit questionnaires, conduct analysis	Medium	Medium	High
FR4	Administrator able to add, edit and delete users and questionnaires	Medium	Medium	High
FR5	Administrator can query and view the survey response by ID or date of survey	Low	Low	Medium
FR6	Administrator able to conduct data analysis in the form of frequencies and cross-tabulation	High	High	High
FR7	Administrator are able to filter dataset when performing data analysis	High	High	High
FR8	Analysis can be produced in table and graph	Medium	Medium	High
FR9	Analysis dashboard generates percentage of answers for all questions	Medium	Medium	Medium
FR10	Data analysis can be exported into Excel	Low	Low	Medium
NFR1	Design interface should be simple, clear, easy to use and looks professional	Medium	Medium	Medium
NFR2	Learning time for using the system is expected to be not more than 3 hours for administrator and not more than 5 minutes for general user	Low	Medium	Medium
NFR3	Data shall be kept safe at all time	High	Medium	High
NFR4	The system available at least 95% of all times	Medium	Medium	Medium

*FR denotes Functional Requirement and NFR denotes Non-Functional Requirement

By prioritizing project requirements and considering their risk and complexity, essential functions that are crucial to the project success can be treated accordingly. In addition, appropriate implementation sequence can be planned and executed based on the most important functionality.

4.3. System Design

This section detailed the application of systems theory in terms of design, architecture, data and modules to the system implementation.

Design layer

A three-layer design encompassing Data Layer, System Management Layer and Mobile Application Layer was created. Description for each layer is as follows:

Data Layer – Consists of Database Management System (DBMS) that will store and manage the database schema, survey data, user data and questionnaires data. The system and mobile app will interact with the data here by using SQL language. Only users with superuser access level can login to this layer.

System Management Layer – Utilizes mobile web development strategy. Provides interface for the administrators to interact with data at the data layer. This layer provides administrator with user-friendly interface to manage system users, manage data (add, edit, delete, view), analyse data and manage questionnaires. System administrator will have the convenience of using their web browser to perform these tasks.

Mobile application Layer - Hybrid mobile app development strategy is used. Hybrid mobile app allows the app to work smoothly on mobile devices and allows data to be stored locally on the device, thus eliminating the need for constant connectivity to the network. In addition, it simplifies the majority of the coding process due to familiarity with web development languages. In addition, it will provide the potential for cross-platform usage in the future.

Figure 4-2 below illustrates the overall design of the three layers and their interactions during the system lifecycle.

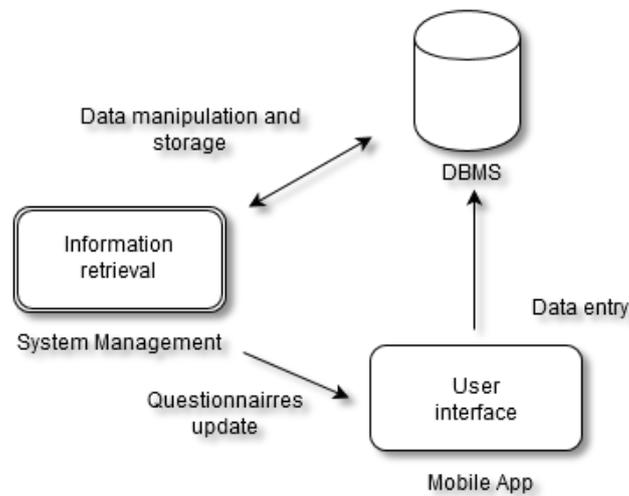


Figure 4-2: Data, System Management and Mobile App Layers

Architecture

In designing the application architecture, Model-View-Controller (MVC) pattern was employed. Trygve Reenskaug first envisioned the MVC design pattern back in 1970s at the Xerox Parc with the main objective “to bridge the gap between the human user’s mental model and the digital model that exists in the computer” (53). MVC pattern separate the responsibilities for an application into three main components to allow for efficient collaboration. The three main components in MVC are:

- **Model** – manages tasks related to the logical structure of data, state of the app and the business logic of the application
- **View** – collection of classes representing elements in UI and presentation. By splitting the application design from application logic, the task of modifying UI will be simpler and reduce the possibility of making errors due to inconsistency with the application logic

- **Controller** – logic for handling user actions and communicate between classes in the model and view

Figure 4-3 below depict the MVC structure used and the interaction between them.

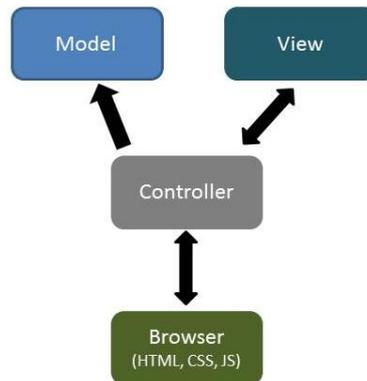


Figure 4-3: MVC Structure

Interaction between these components will be done according to user actions. The resulted implementation is a client-side hybrid app, with HTML and CSS forming the view, objects including the state of the UI as the model and JavaScript as the controller. Based on studies done by (54), segregation of responsibilities in MVC structure provides several benefits:

- Logic for each MVC components can be changed or evolved independently without affecting the other components;
- Separation of view from the model allows for designing various user interfaces (to accommodate user needs) to represent the same data; and
- Enhance application testability due to loose coupling that exists between view and controller in the structure.

4.3.1. Mobile App

Mobile devices such as tablets and smartphones provide a convenient and easy-to-use tool with context-sensing capabilities and flexibility for mobility. In designing for the mobile app,

factors that are commonly considered are the user’s requirements, user’s expectations, project specifications, developer’s capabilities, resource availability, security, budget, and duration.

Architecture Framework

Apache Cordova was employed as the mobile development framework for the hybrid app. Apache Cordova acts as a web-to-native wrapper, enabling deployment as the app is native to the device thus allowing access to the device specific capabilities. It is an open-source framework, which allows development to be conducted by using the standard web languages such as HTML, CSS and JavaScript. Mobile apps will be executed within wrappers of the targeted platform, in which they would depend on the standards-compliant API bindings that has access to each of the specific device capabilities such as graphics, sensors, and data network.

Figure 4-4 below show the overall architecture for Apache Cordova framework.

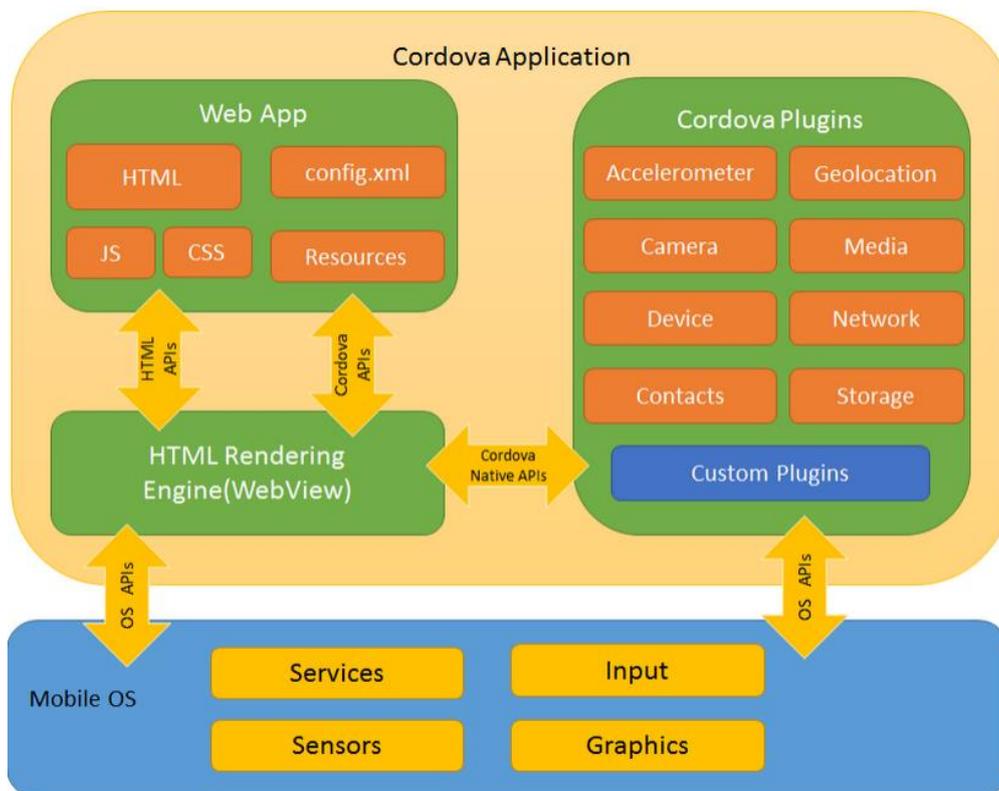


Figure 4-4: Apache Cordova Architecture (55)

Web app is where the code for this project resides in the framework. The implementation was conducted in HTML for displaying the UI, CSS for style and formatting and JavaScript for performing functionalities as specified in the project requirements. Images and media files are stored in the resources folder. The config.xml file provides the app configurations and specifies parameters such as the touch sensitivity and orientation shifts. The information is passed to the WebView that will provide the web app its complete UI. This is communicated using the APIs for HTML and Cordova.

WebView executes the app within the native application wrapper, and makes use of the Cordova plugins to access the device features. The plugins provide an interface for Cordova and the native device components to connect and bind using the standard device APIs. This allows for specific device features such as storage to be accessed and used by the mobile app.

User Interface

Choosing hybrid app approach for the development presents a challenge in designing for the UI, as apps developed and run in their native code would have the best access to the device capabilities thus assuring the best possible UI. However, current mobile devices support WebKit-based rendering machine that is commonly used in browser engine such as Firefox and Chrome, ensuring support for HTML5 and adaption to the latest W3C specifications (56). Therefore, it allows the hybrid apps UI to be designed using the common interface seen on web. Users will see the familiar design and this will bring benefit for cross-platform implementation, as the interface will always be consistent across the platform.

On the contrary, native app developers will not be able to reuse the same UI for different platforms. In addition, they are also required to learn the skills for each platform language (e.g. Objective-C, Java) and use platform specific development kit and environment to deal with its own conventions. From the user's perspective, the hybrid app essentially feels like web content accessed from a mobile device, with the ability to access the mobile device's capabilities using native APIs.

Resource Management

When designing for mobile app, design constraints such as the processing power, screen size and system memory availability should be considered. Application resources must be managed efficiently to ensure smooth experience. This can be achieved by understanding the activities in the Android development environment lifecycle and implementing it in the app development.

Figure 4-5 below shows the states and transitions for each Android activity lifecycle.

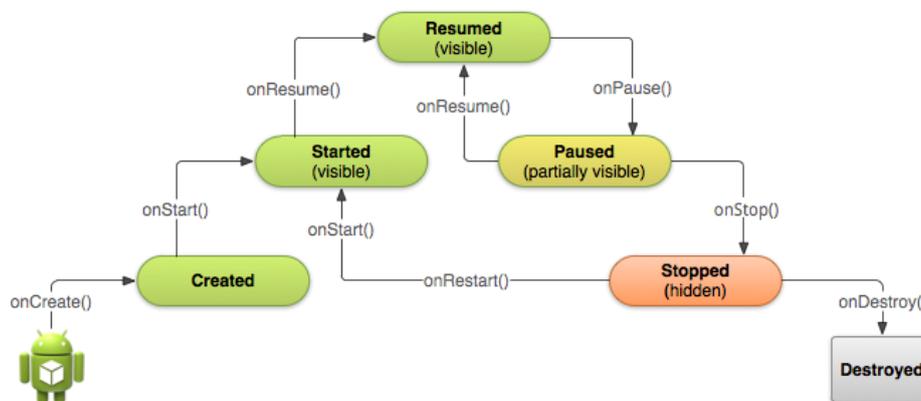


Figure 4-5: Android Activity Lifecycle States and Transitions (57)

When the application is first instantiated through the `onCreate` method, it will reside at the **Created** state. The basic operational logic will be performed only once for the entire lifecycle activity. Next, `onStart` method will move to the **Start** state where the application will now be visible to the user. Upon receiving external input, `onResume` method will move to the **Resume** state where this will become the main state for the application.

As the user may open other apps and partially consume the main activity, the `onPause` method will put the application into **Pause** state. This is an important state where special attention should be given to ensure that only the main resources for the device are kept running. The `onStop` method will put the application to the **Stop** state where the application will be hidden, all resources will be released and the other application activity will run and presented to the user. Method `onDestroy` put the application at **Destroy** state in which it will be the last state where application activity will be released from the device's memory.

Functions

The main function that will be performed by mobile app is survey data collection. However, in order to support the main function and project requirements, additional features must be added. Below are the features included in the mobile app:

- **Survey mode** – Visitors will be able to view the survey questionnaires and answer them directly from the app. Data will be saved locally before uploaded to the server when network connection becomes available.
- **Authentication** – Certain features such as data sync with server or survey questionnaires management would require authentication before the user is allowed to perform the action.
- **Settings** – Options such as updating the user list for administrator or updating survey questionnaires.
- **Data upload** – Administrator will be able to view survey responses, delete the response or upload the responses to the CSMCR survey system server.

4.3.2. Web Application

The CSMCR Survey web application provides the interface for system administrators to easily manage the survey questionnaires, responses and conduct data analysis. It is accessible from any browser including from mobile devices.

Web application architecture

MVC architecture pattern was employed in the web application. The model component signifies the report information, test cases, outputs and all operations involved in generating and accessing such information. In addition, the model includes classes to handle user

requests (e.g. the inputs for the survey data or data analysis). A single folder acts as the controller, containing all the functions related to user requests handling, model operations execution and presenting results by interacting with the views. The view folder contains all the UIs and the model folder contains files controlling the logic of the application.

Figure 4-6, Figure 4-7 and Figure 4-8 below show the MVC files structure for the web application.

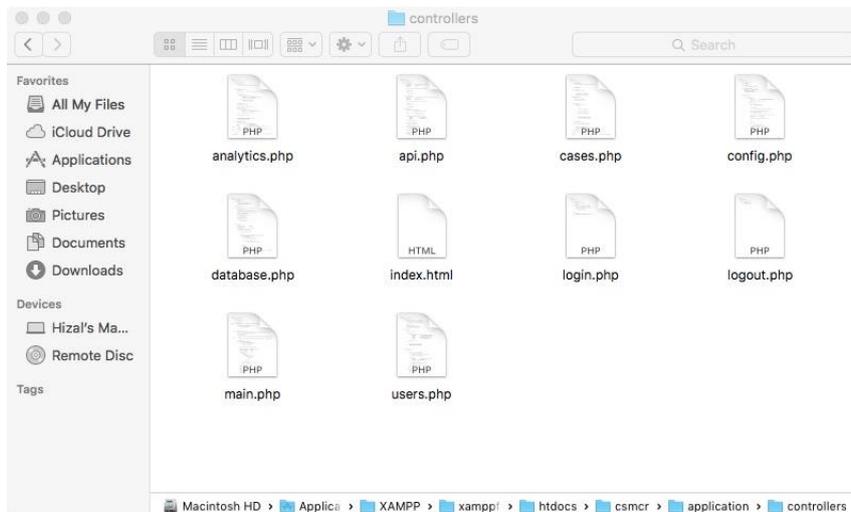


Figure 4-6: Web Application Controllers

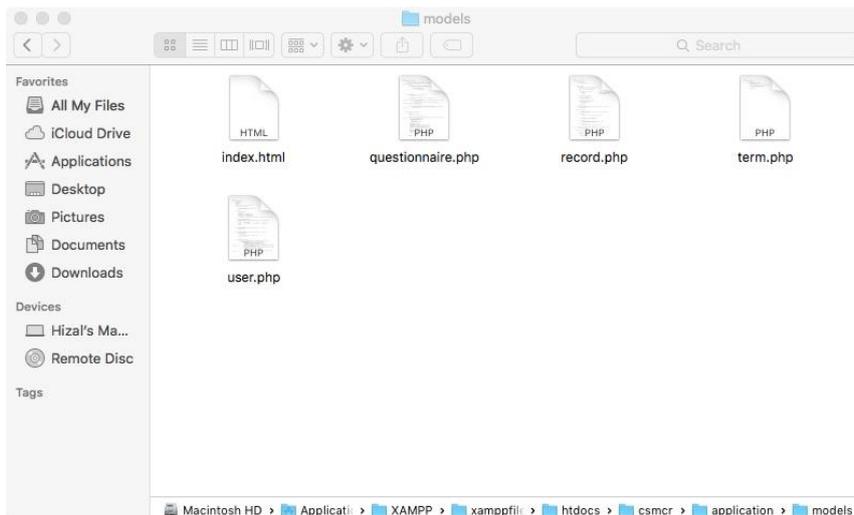


Figure 4-7: Web Application Models

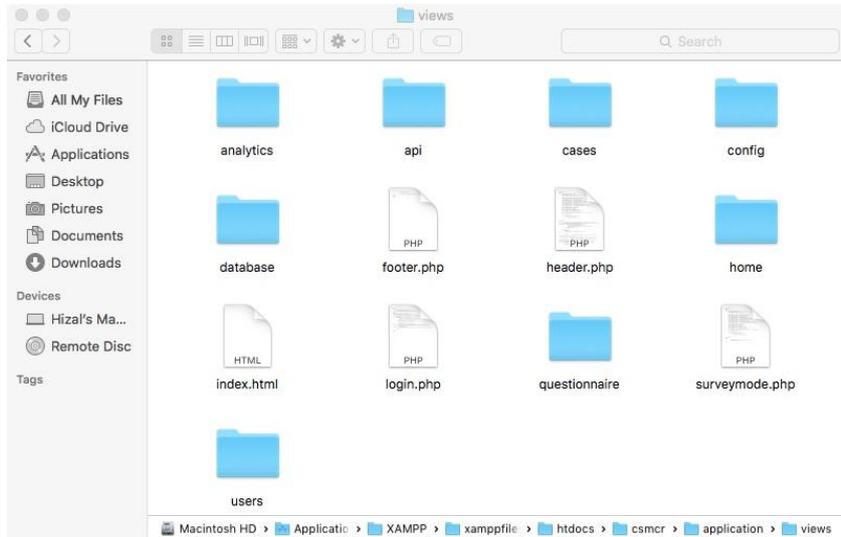


Figure 4-8: Web Application Views

The web browser sends the user input as HTTP requests and routed to the correct controller class in the following format:

“http://domain/{controller}/{action}/{parameters}”

Controller represents the name of the specific controller, **action** represents user initiated request and **parameters** are the values required by the controller to carry out the specified action. Design of the main features for the web application is described in Table 4-4 below.

Table 4-4: Summary of Web Application Functionalities

Module	Function	Description
Survey	Home	Main page of the web application
	Survey Mode	Display the survey questions. This serves as a backup solution should the mobile app is unavailable to use. Respondents may answer the survey just as they would on the app. However, this would require constant connectivity to the network.
	List of surveys	Provide the capability to query for survey records based on the survey ID or date of the survey. Administrator may view or update the completed

		survey question and answers.
Data Analysis	Dashboard	Provide overview of the survey data analysis.
	Frequency	Create frequency table based on the selected question from the dropdown menu. Data is also represented in a graph view.
	Contingency	Create contingency table based on two selected questions from the dropdown menu. In addition, administrator may also specify filtering for the dataset they want to create the table from. The filtering works by both 'and' and 'or' selection. For example, a contingency table of 'How would you rate the refreshments' and 'How would you rate the talk after lunch' may be generated. Filtering can be added to limit the dataset based on the needed criteria (e.g. only respondents who took the lunch).
Users	User list	List all the users registered with the system. Administrator may update or delete specific users.
	Add new	Add new users to the system.
Settings	Survey Questions	Provide a list of all survey questions. Administrator may add, edit, delete or rearrange the order of questions they want to appear in the survey. In addition, administrator may also further customize the answer options they want to appear in the survey by specifying the HTML or JavaScript code inside the provided box.
Database Management	Backup	Administrator may download or upload the data from the system database by using the csv format.

Development of the system UI and design for the web application take careful consideration of the usability aspect. The main goal is to achieve the balance between aesthetically pleasing design and intuitive and simple user to system interaction.

Chapter 5: Implementation

This chapter will provide details of software development and its complex features; based on research of best practices and technology selection as described in the background chapter. It will also explain the implementation of technologies that are vital for mobile app and web application development, including the server-side implementation and client-side implementation of the system.

XP was chosen as the software development method for this project. Therefore, the software follows iterative and incremental release. The implementation process was divided into a series of time-boxed iterations. Each iteration involves software design, programming and testing. A set of requirements for each iteration that will be implemented must be specified in advance before it can begin. Deliverables for each iteration were in incrementally functional prototype of the whole system.

5.1. Environment setup

Based on background studies and project requirements conducted earlier, an ideal hardware and software environment was set up and deployed for the project implementation. Figure 5-1 shows the overall set up. The data and system management layers are deployed within the School of Computer Science server, while the mobile app was installed inside the tablet provided by the school.

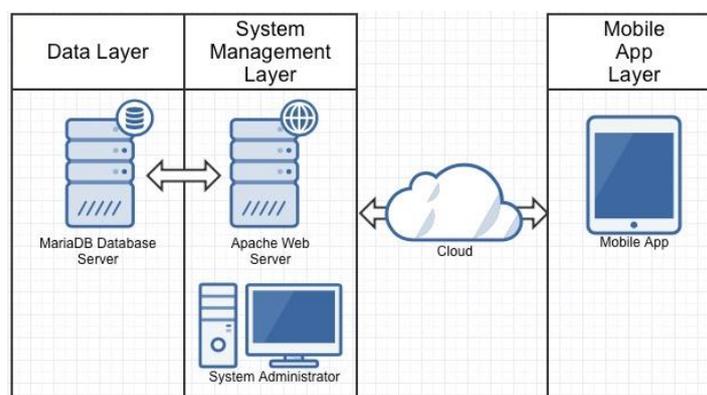


Figure 5-1: Overall System Set Up

The data layer contains database server on a MariaDB v. 10.1.13 instance that was configured specifically for this project. During the development phase, phpMyAdmin v. 4.5.2 was used as the interface from the web browser to setup and build the database schema. On the other hand, the system management layer contains the web server to host the web application. Apache v. 2.4.18 was installed to act as the web server. The system management layer was developed using PHP language v. 5.5.37 and JavaScript. System administrator can login to the system via web application and access the database using the interface provided.

In the mobile app layer, the mobile app that was developed run on an android tablet with the Android operating system (Lollipop, 5.3.1). The mobile app was developed and deployed by using Android Studio v. 1.5.1. Additional packages (Java SE Development Kit 7, Android 5.1.1 API Level 21, Android SDK Tools 24.4.1, Android SDK Build Tools) were downloaded in order to support for Android app development. These dependencies are important to ensure that the app can be developed based on the intended hardware and software requirements. Android Virtual Device (AVD) was created based on the characteristics of the targeted device to simulate the app running during testing and debugging process.

Figure 5-2 below shows the Android Studio and AVD development environment.

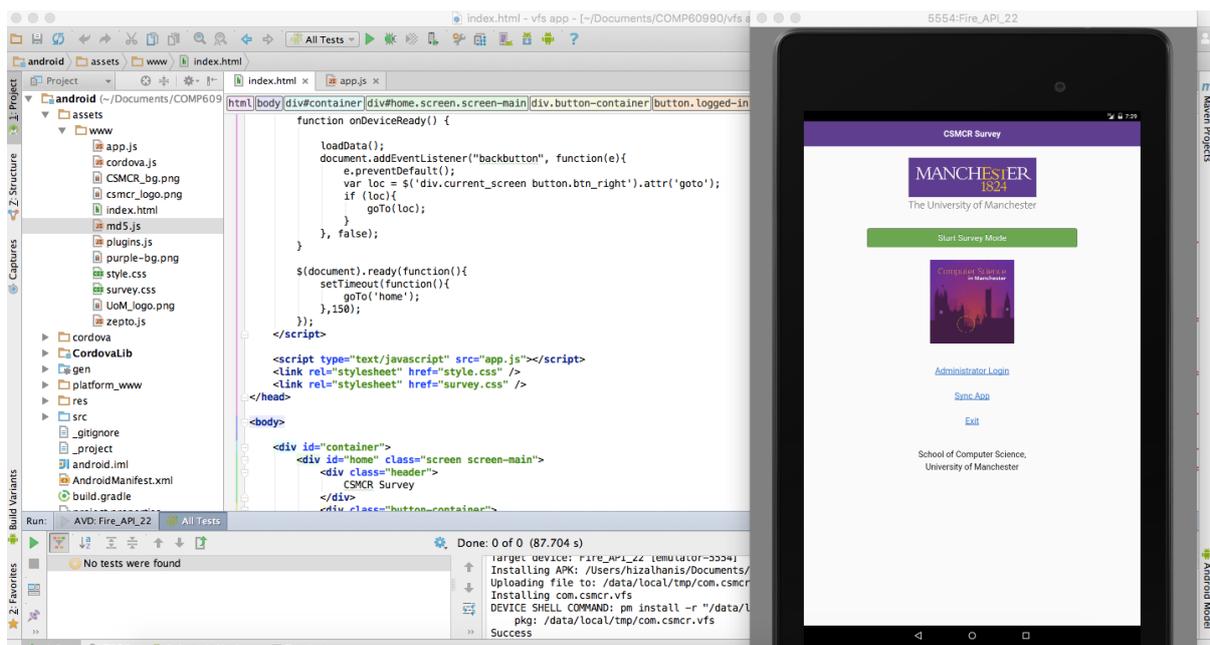


Figure 5-2: Android Studio and AVD

Connection between mobile application layer and system management layer can be made either by Wi-Fi or mobile network. To ensure for a successful connection, the system management layer should reside within a server that is accessible by a public IP address. Alternatively, both the mobile app and system management layers can reside in the same local area network with their ports open and visible to the client.

5.2. Database

MariaDB v. 10.1.13 was chosen as the DBMS for this project. It is an open source relational database established by the original founders of MySQL after it was acquired by Oracle. MariaDB maintains high compatibility with MySQL as it has the “drop-in” replacement capability and contains library binary equivalent and exactly matching MySQL APIs and syntax (58). This DBMS was selected for this project as it offers all the functionalities needed.

There are three tables created for this project. Table questionnaire was created for the purpose of storing the survey questions and its related attribute. Table survey_gen serves as the main table to store the survey data collected from the respondents. To manage users' information in the system, table users were created.

Table 5-1 summarizes the overall database and table structures.

Table 5-1: Data Structure for CSMCR Survey Database

Table name	Field name	Description	Example value
Questionnaire	id	Specify the question ID auto-generated by the database	2069
	question	Specify the question and options related to the question in JSON representation.	{ "description":"Do you have any other comments you would like to make?", "questions":[{"no":1,"question":"","labels":null,"comments":0,"comments_description":"","state_reason_if_min":false,"state_reason_if_max":false}
	answers	Provide coding value and its corresponding answer option for close-ended questions. Value is NULL for open-ended questions. Use JSON representation.	[{"no":1,"value":"Excellent"},{"no":2,"value":"Good"},{"no":3,"value":"Okay"},{"no":4,"value":"Poor"},{"no":5,"value":"Bad"}]
	type	Specify the question type. Single-answer: Use for close-ended questions. Matrix-answer: Use for open-ended questions. Multiple-answer: Use in close-ended questions and users are allowed to choose more than one answer.	Single-answer

	Matrix-choice: Typically use with questions requiring 'Likert-scale' answer options.	
map_to	Specify which field the answer value will be stored in survey_gen table.	other_comment
no	Specify the order numbering for the question	1
display	Specify if the question will be displayed in the survey questionnaire. Value: 1 – Question will appear in the survey questionnaire 0 – Question will not appear in the survey questionnaire	1
Data_type	Specify the special data type (e.g. range, date, time) for analysis purposes. Value: NULL – Normal data type date – use for date data type time – typically when answer is provided in time/hour format age – typically when answer is provided with age information	age
script	When creating a question, administrator may specify the use of JavaScript in displaying the answer option. Hence, the JavaScript will be stored in this field.	<pre><script language="javascript" type="text/javascript" > Function Sample {} </script></pre>
modifier	When creating a question, administrator may specify the use of HTML in displaying the answer option. Hence, the HTML code will be stored in this field.	<pre></pre>

	other	In providing answer options in close-ended questions, administrator may request for specific answer when answer option 'other' is selected. This field will store if the specific answer is requested. Use JSON representation.	{"state_reason_if_checked": "0"}
Survey_gen	Id	Answer id auto-generated by the database	1
	Date	Date when the survey was taken	01/07/2016
	*question's specific field names	The field name will be specified when administrator add the question. Field name should be specified in the database first, before adding the question through the management system. Therefore, the administrator will be able to see the field name in the database from the dropdown list when adding the questions.	mode_of_transport
users	Id	User id auto-generated by the database	1
	Username	The username chose by the user	admin
	password	Password chose by the user. Value saved in MD5 hash.	21232f297a57a5a743894a0e4a801fc3
	Type	Type of user. Value: Superadmin – user have all the administrative rights to the database Admin – user may add, edit, delete and upload survey data User – user may only add survey data	user
	Name	Name of the user	John Appleseed

ip	The IP address when the user last logged in to the system	130.88.195.77
Session	Last session ID from the user	8da94fb7b2ed17b720bfb4bb34e82849
Timeout	Timeout for the user	1469614661
Hash	The hash function key for the user	bae2142c8c672bf6083fe391f019c921
Added_on	Date and time of when the user is added to the system	2016-07-25 20:39:06
Added_by	Specify the username of who added the user to the system	john_appleseed
Modified_on	Specify the date and time of when someone last modify the user information in the system	2016-07-04 10:54:21
Modified_by	Specify the username of who last modify the user information in the system	admin

The database can be accessed using the phpMyAdmin system from any web browser with the address <https://dbhost.cs.man.ac.uk/phpMyAdmin/>. The 'database.php' configuration file in the web application directory was updated with the following information:

```

1 <?php
2 $db[ 'default' ] [ 'hostname' ] = "dbhost.cs.man.ac.uk";
3 $db[ 'default' ] [ 'username' ] = "mbaxrhh2";
4 $db[ 'default' ] [ 'password' ] = "admin123";
5 $db[ 'default' ] [ 'database' ] = "mbaxrhh2";
6 ?>

```

Figure 5-3 below shows the phpMyAdmin interface to create and manage the database.

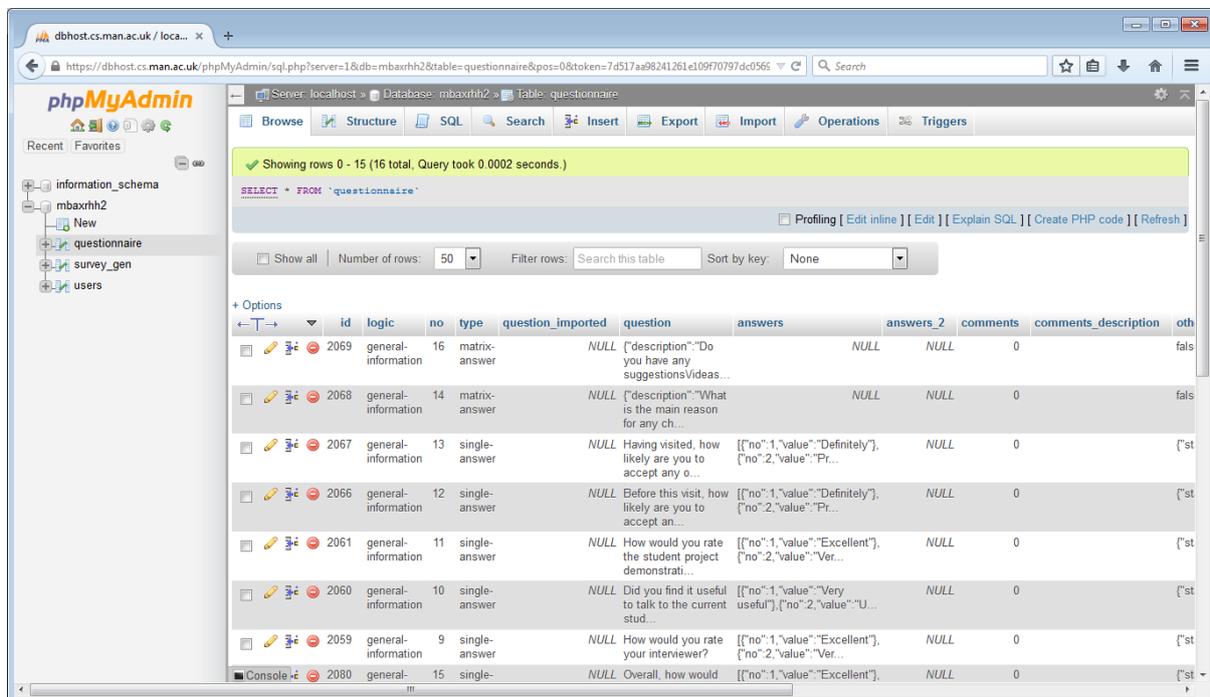


Figure 5-3: phpMyAdmin Interface for Database Management

In creating the tables, indexes were also created for several key attributes (columns representing identifiers, dates). This will help to speed up the data query process when data analysis is performed by the system. By completing this process, the main database implementation is completed and the Data layer is now available to be used by the system management layer and mobile app layer.

5.3. System Implementation

The system developed in this project consists of two major parts, which are the mobile app and web application. The web application in this project is mainly used for managing the survey questions and records, data query and display relevant data analysis to the user in tables and graphical representation. On the other hand, the purpose of the mobile app is for collecting respondent feedback and transmitting it back to the database server.

Most web environment in the market today uses HTML and CSS to provide interface and present data to users and JavaScript is used for user interactions. This is known as “front-end” or “client-side” and the same method is used for implementing this project.

In client-side implementation, LocalStorageProxy stores the proxies that are used in order to load and save the model data. Data objects are transferred by using the JavaScript Object Notation (JSON) using the standardized data format (`{"attribute": value}`). Readers will play the functions as a data interpreter before they are loaded into a model or store. LocalStorageProxy is particularly useful to store user-specific information without having the need for server-side infrastructure.

5.3.1. Mobile App

The main function of the mobile app is for conducting surveys. The app can work offline as survey questionnaire, survey responses and user credentials information are stored locally. Most of the coding were done in JavaScript, jQuery and HTML. The use of wrapper in Apache Cordova framework (as detailed in Chapter 4) will do the translation and execute the code.

Features that were created for the mobile app are described in following sub-chapters.

Homepage

Provide options to start survey mode, administrator login and sync app. No login is required to start the survey mode. However, for settings and upload data option, administrator login is required. Upon administrator login, additional buttons will appear for list of saved responses and logout (Figure 5-4).

When the app is first initialized, the loadData() function will be called. Survey questionnaire and saved responses were stored in JSON format. If the data is available, the app will fetch the data from the local device storage and display it. Snippet of the code is shown as below:

```
function loadData(){  
    var data = window.localStorage.getItem('records');  
    if (!data) records = [];  
    records = JSON.parse(data);  
}
```

Logout

Provide the option to logout from the administrator account. Upon successful logout, 'List of Saved Responses' and the logout buttons will be hidden (Figure 5-5).

Survey Mode

The survey process starts once the user press on the survey mode. Respondents can answer the survey questionnaire by using the app and submit their response. The response is saved in the devices' local storage and a new set of questionnaire will be generated ready for the next respondent to answer. Exit from survey mode can be done by 'double-tap' the exit button. This is to prevent the survey respondents from exiting the survey mode on their own (Figure 5-6).

Survey questionnaire is loaded from the device storage. Variable logic is used to allow flexibility in the future should there is more than one form will be used. The overall code used is shown in the snippet below:

```

function loadForms(){
  var cf = window.localStorage.getItem('caseForm');
  var ud = window.localStorage.getItem('userData');

  if (cf){
    $('span.case-form-status').html(' &nbsp; &#x2713;');
  }

  if (ud){
    $('span.user-data-status').html(' &nbsp; &#x2713;');
  }

  $('#surveyform div.content').html(cf);

  $('form.question').submit(function(e){
    e.preventDefault();

    var no = $(this).attr('no');
    var logic = $(this).attr('logic');
    var data = $(this).serializeObject();

    switch (logic){
      case 'general-information':
        saveCaseRecord(currentRecordID, no, data);
        break;
    }
  });
}

```

To save the data, the `.serializeArray()` method is used. The method will create a Javascript array of objects. Then, `JSON.stringify` method will parse it into JSON text and store it as a string. Snippet of the code use is shown as below:

```

$.fn.serializeObject = function(){
  var o = {};
  var a = this.serializeArray();
  $.each(a, function() {
    if (o[this.name] !== undefined) {
      if (!o[this.name].push) {
        o[this.name] = [o[this.name]];
      }
      o[this.name].push(this.value || '');
    } else {
      o[this.name] = this.value || '';
    }
  });
  return o;
};

function saveData(){
  var data = JSON.stringify(records);
  window.localStorage.setItem('records',data);
}

```

List of Saved Responses

This option is only available upon administrator login. List of all survey responses saved in the device will be shown. Administrator will have the option to view, edit or delete each of the responses. The administrator can also upload all responses at once or by individual response (Figure 5-7). The following code snippet shows the upload all function used.

```

$('button.upload-all').click(function(){
  if (confirm('Upload all survey records?')){
    for (var id in records){
      var case_record = records[id];
      var case_data = JSON.stringify(case_record);
      api('submit_case', 'data='+encodeURIComponent(case_data)+'&user_id='+currentUser.id, function(res){
        if (res.status == 'OK'){
          for (var id in records){
            deleteRecord(id);
            goTo('home');
          }
        }
      })
    }
  }
}
}

```

Administrator Login

Login to the administrator account and access additional features such as view list of saved responses, upload data and sync app (Figure 5-8). Password is saved with MD5 hash function and comparison will be made with the stored credential on the device. Code snippet below shows the overall login process.

```

$('button.login').click(function(){
  var username = $('input.user-username').val();
  var password = hex_md5($('input.user-password').val());
  var ok = false;
  for (var i in users){
    var user = users[i];
    if (user.username == username && user.password == password){
      goTo('home');
      window.localStorage.setItem('logged_in',1);
      window.localStorage.setItem('current_user',JSON.stringify(user));
      loggedIn = 1;
      $('.logged-in').show();
      $('.unlogged-in').hide();
      ok = true;

      currentUser = user;
    } else {
    }
  }

  if (ok){
  } else {
    alert('Incorrect username or password')
  }
})

```

Sync App

After installing the app to the device, administrator can download the survey form and the login information for the app administrators. A successful download confirmation box and right tick will appear next to the button to indicate successful download. Once the login information is downloaded, users will need to login in order to access this feature again. The login was disabled for the first time access as there is no login information saved in the device yet. Figure 5-9 shows the snapshot of Sync App.

The server URL to sync the app used the following format <http://{domain}/{root directory}/api/>. In addition, the sync app is also used to update the app survey questionnaire. Modification to the questionnaire can be done by using the web application and the sync app will pull the update information to the mobile device.

The snippet of code below shows the functions that are used to download the survey form and user data. Function `api()` is used to send the request to the server specifying the type of information needed and confirmation of success. The JSON data received will be stored in the device local storage as string.

```
function downloadCaseForm(){
  api('Get_General_Form',null,function(res){
    if (res.status == 'ok'){
      window.localStorage.setItem('caseForm',res.html);
      alert('Survey form updated!');
      loadForms();
    }
  });
}

function downloadUserData(){
  api('Get_Users_List',null,function(res){
    if (res.status == 'ok'){
      console.log(res.data);
      window.localStorage.setItem('userData',JSON.stringify(res.data));

      alert('User lists updated. Please restart the app to login. ');
      loadForms();
    }
  });
}

function api(request, data, success){
  $.ajax({
    url: api_url+request,
    type: 'post',
    data: data,
    dataType: 'json',
    success: function (res){
      success(res);
    }
  });
}
```

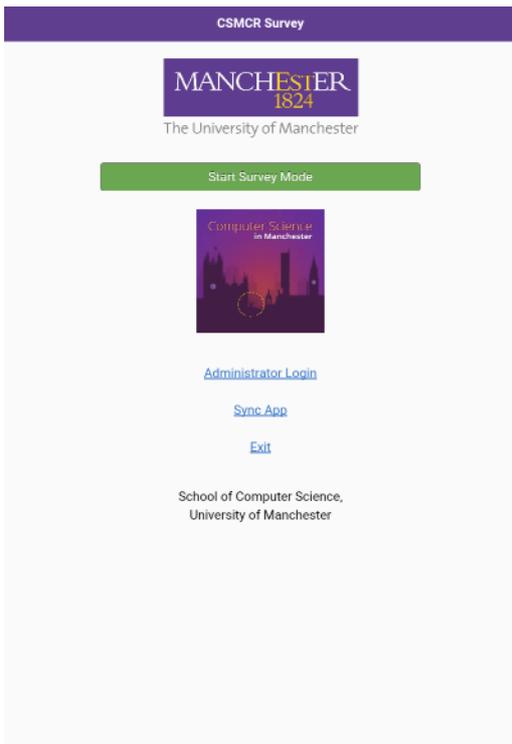


Figure 5-4: Survey App Homepage

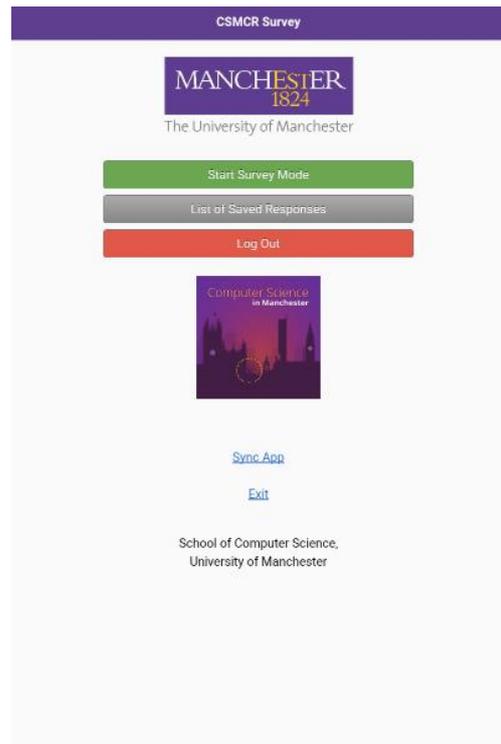


Figure 5-5: Additional features in Homepage after Administrator Login

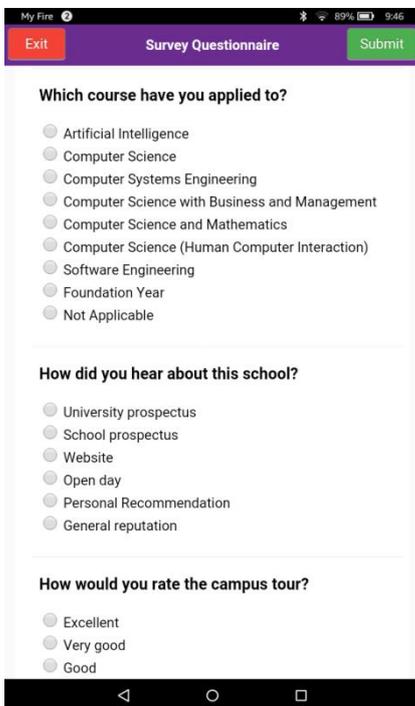


Figure 5-6: Survey Mode



Figure 5-7: List of Saved Responses

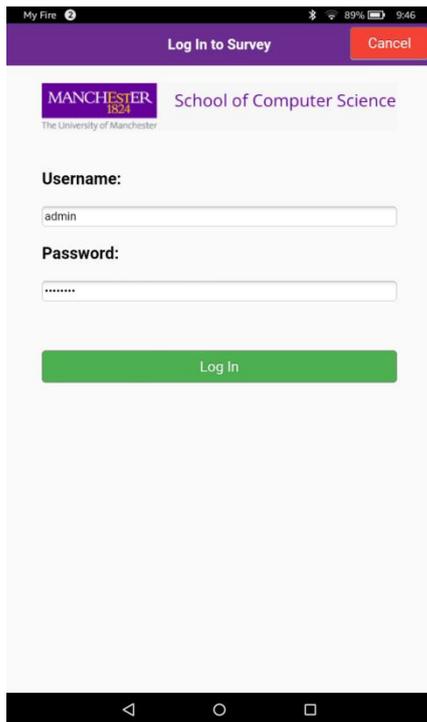


Figure 5-8: Administrator Login to the App

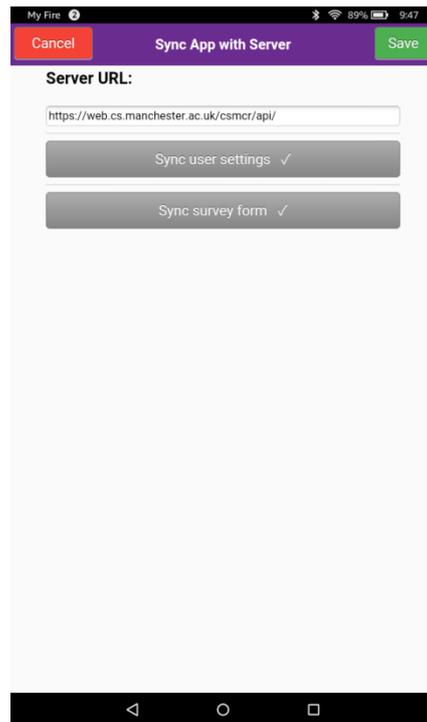


Figure 5-9: Sync App

5.3.2. Web Application

The web application and database server are hosted on the School of Computer Science server (<https://web.cs.manchester.ac.uk/mbaxrh2/csmcr>). Apache is used as the web server and MySQL as the database server. The school server publishes CSMCR Survey web application from the remote Github repository (<https://github.com/hizalhanis/csmcr.git>).

Most part of the web application was programmed in PHP, HTML is used for displaying the content, JavaScript (includes jQuery) for dynamic update of web content and CSS for styling the content.

Access Control

Access to the web management system is only allowed for the system administrators. Each administrator is assigned with username, password and their user type when their account is created. Upon successful system login (Figure 5-10), they can access to the system features. Authentication is made using the login details and the credentials stored in the

database. Upon successful log on, administrators will be directed to the homepage. The user session is saved as PHP session until they log out from the system.

The login information is as follows:

Username:	admin
Password:	admin123

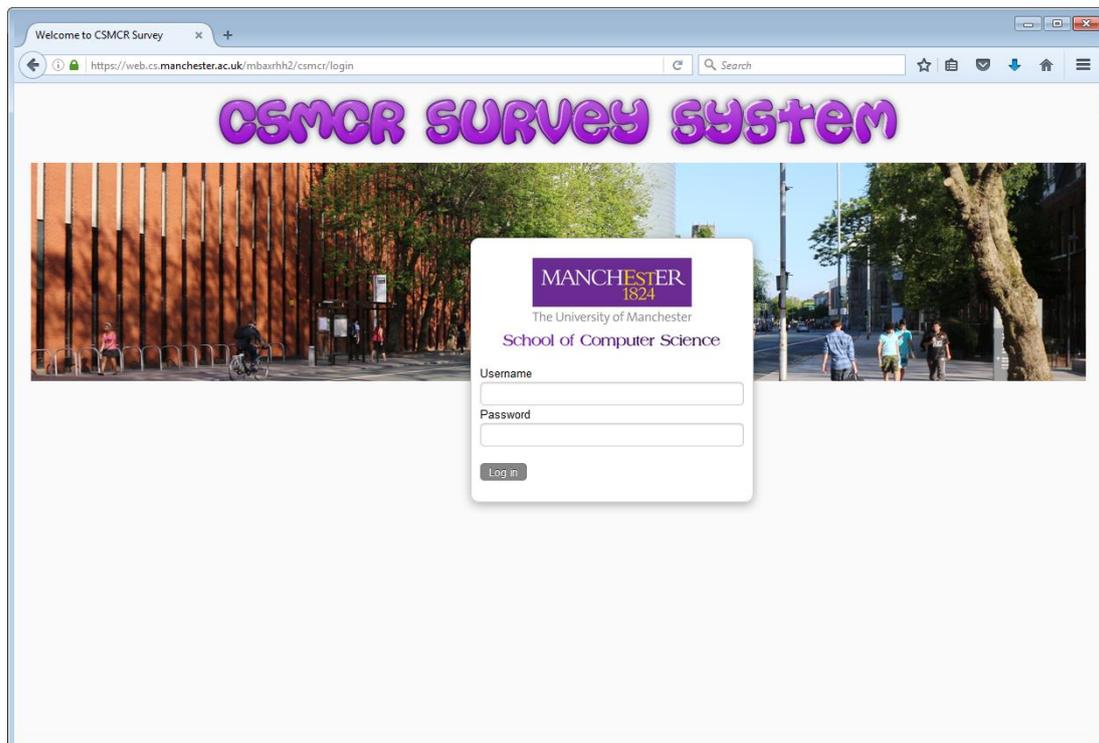


Figure 5-10: Login Page

Main page

Figure 5-11 below shows the system homepage. The user login information is displayed on the right side of the page. Navigation bar provides access to the system main features. Meanwhile, the sidebar menu provides action specific for the features.

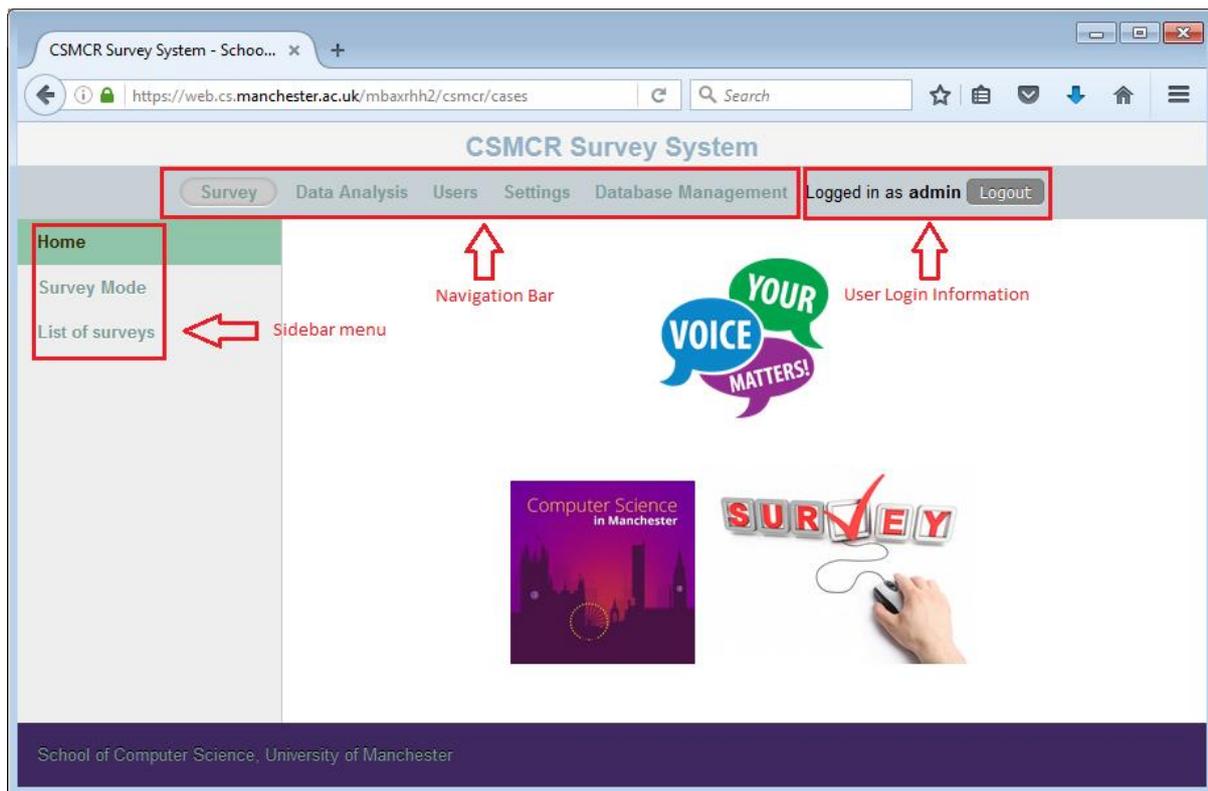


Figure 5-11: Homepage

Survey

The survey features provide options for survey mode and list of surveys. Survey mode provides the capability of conducting the survey using web interface. Answered questions will be highlighted with green colour in the question navigation area. Users may also skip to the questions they wanted to answer by simply clicking on the questions. In addition, the survey mode also acts as a backup solution for conducting the surveys should the mobile app becomes unavailable. Figure 5-12 below shows the survey mode.

On the other hand, list of surveys perform querying of survey responses. This can be done by using the survey ID or the date of survey. The dropdown list will provide all available survey dates. From the list of responses found, administrator can view, update or delete the response. This is particularly useful should there errors occurred in the database such as duplicate responses. Figure 5-13 below shows the list of surveys page.

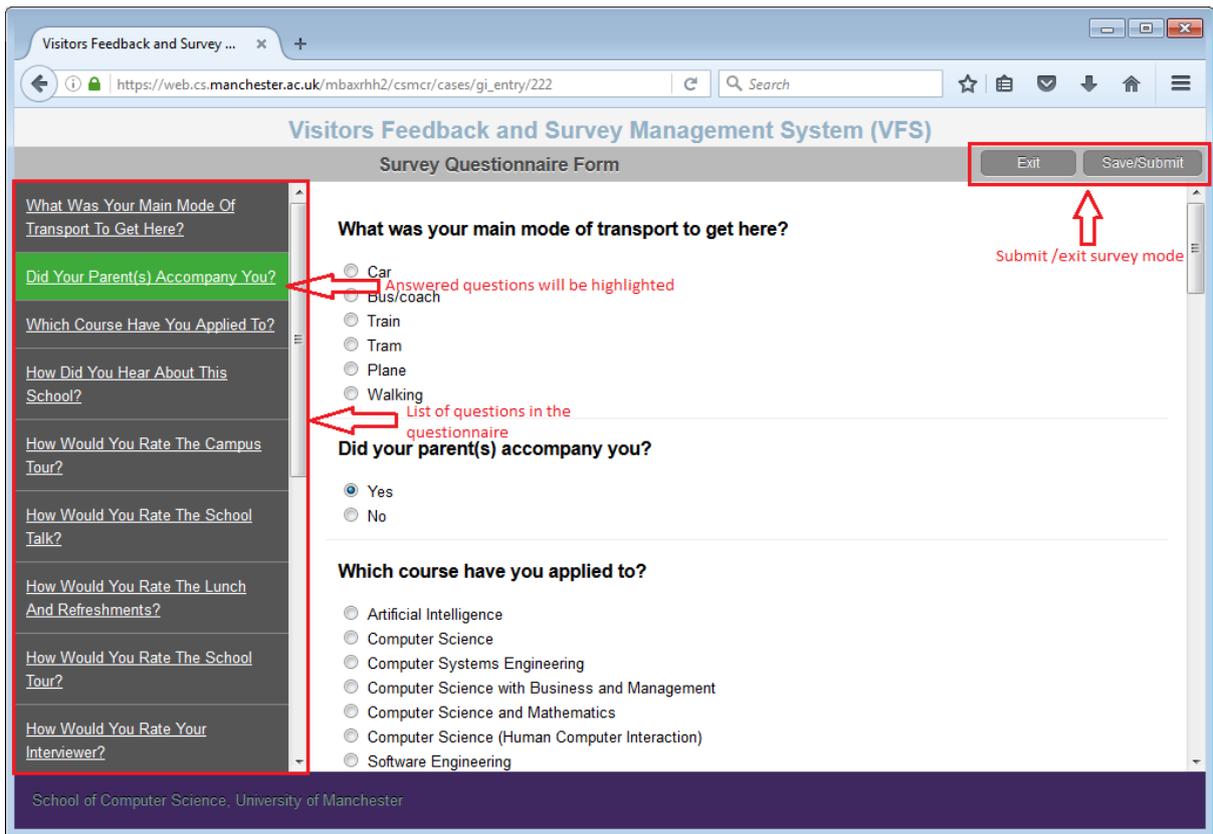


Figure 5-12: Survey Mode

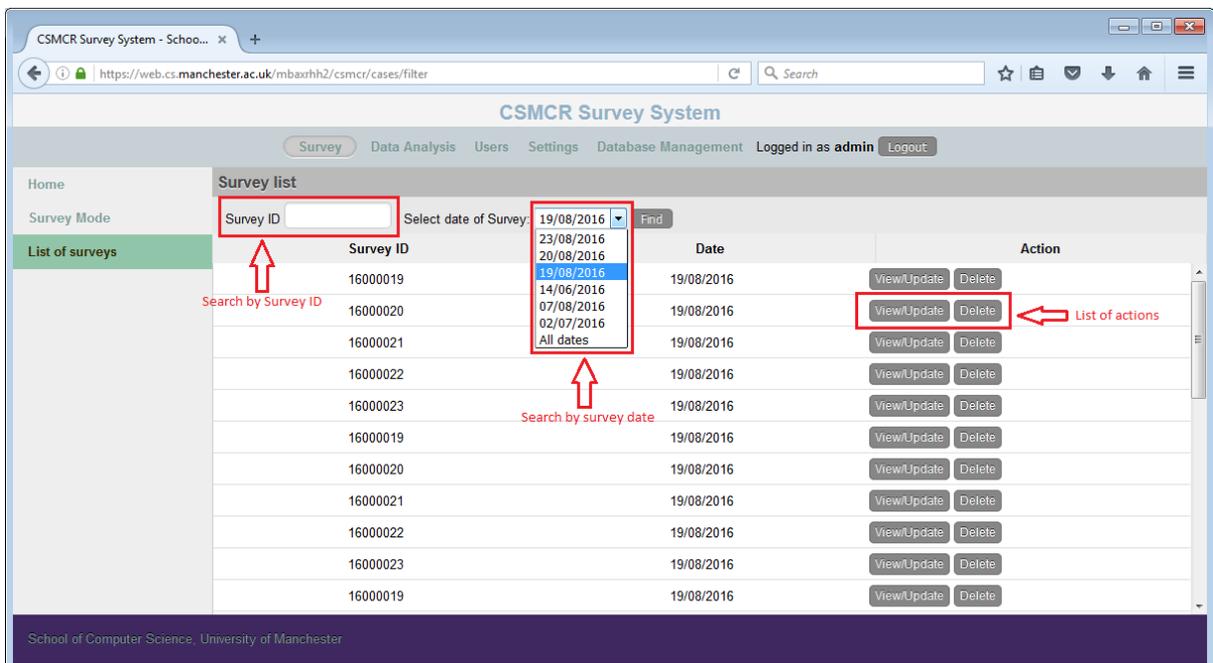


Figure 5-13: List of surveys

Data Analysis

The data analysis feature consists of 3 types of data representation, which are dashboard, frequency and contingency. The details are as follows:

- **Dashboard** – Gives percentage of responses for all questions on a given survey date (or all dates) in a graph and table form. This is particularly useful for a quick glance on answer responses. Figure 5-14 shows the dashboard.
- **Frequency** – Provide the count of responses received for a particular question and its answer options. It is commonly used to see the distribution of answers.
- **Contingency** – Display frequency distribution of questions and answers in the form of a table. This type of analysis shows the interrelation between the chosen questions and help the user to find interactions between them. In addition, data filtering capabilities were included which is useful for narrowing down the dataset. For example, the user can find the distribution of answers for overall event ratings and likeliness to accept the offer, with dataset specific to only those who applied for Artificial Intelligence course. Figure 5-15 shows the snapshot of contingency table options.

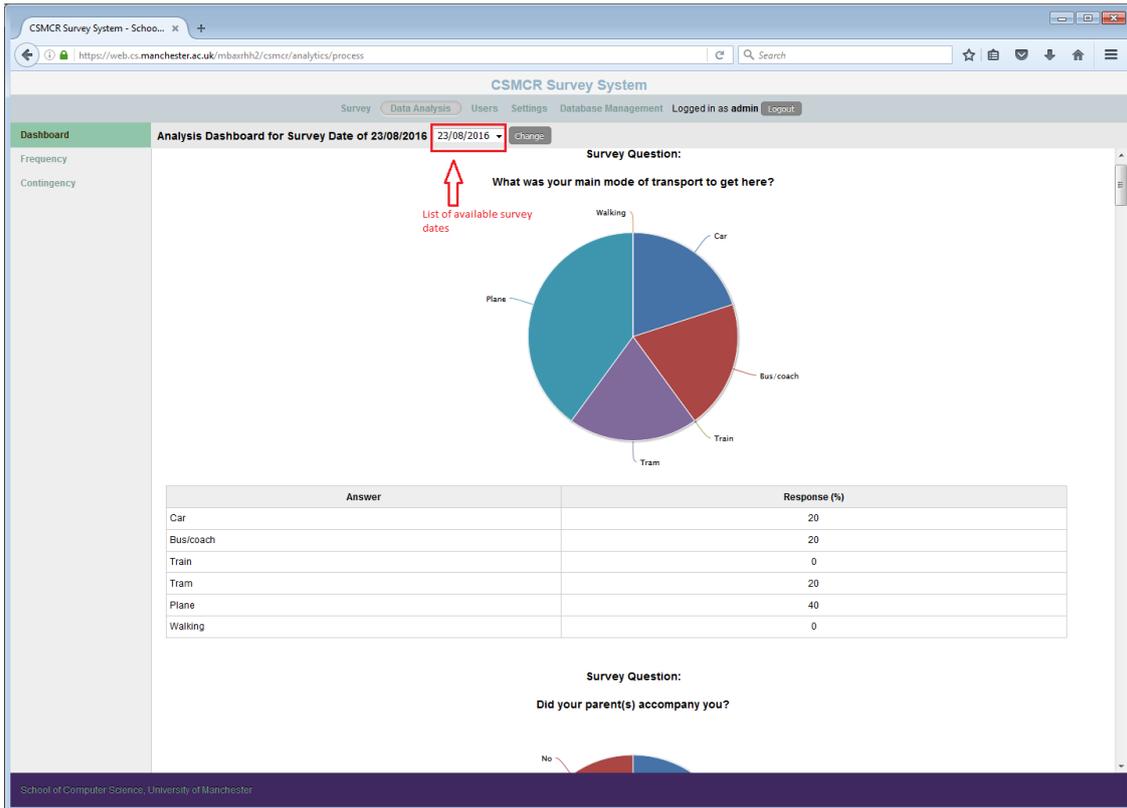


Figure 5-14: Analysis Dashboard

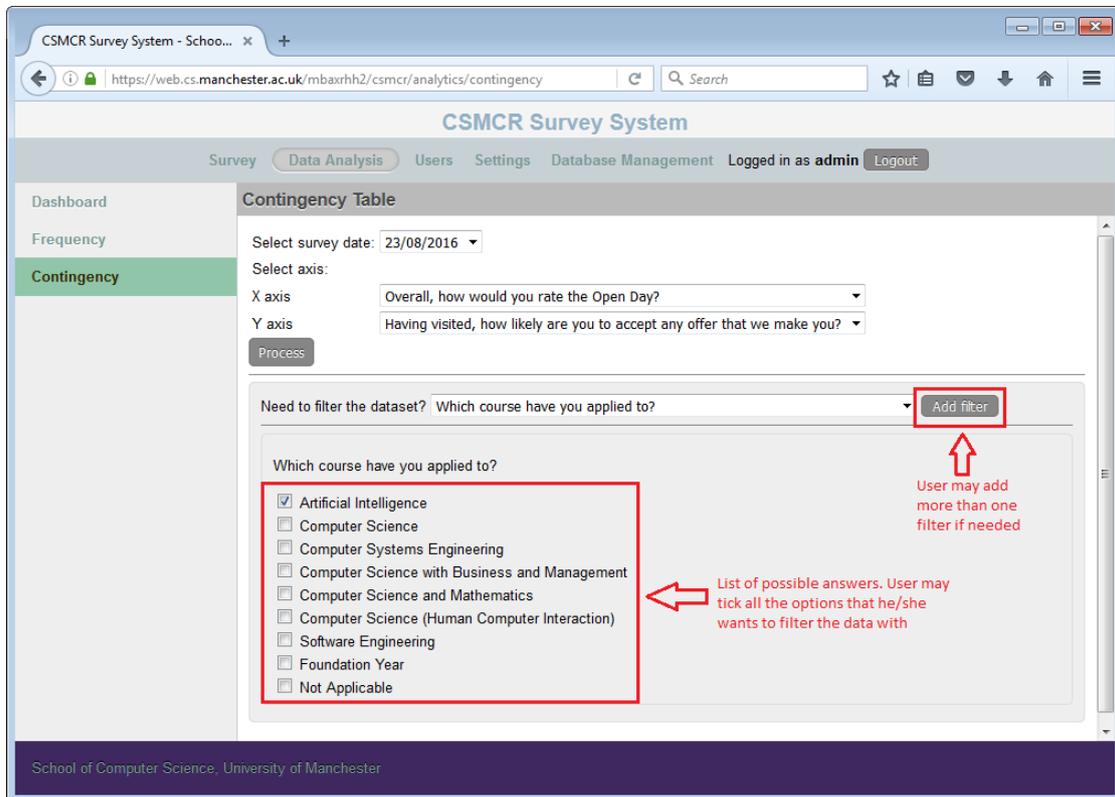


Figure 5-15: Contingency Table with Data Filtering

In conducting the data analysis, the possible answer options were obtained from map_to field in the questionnaire table of the database. The filter for data querying (WHERE clause of SQL) will be based on the question type; close-ended question will have answer options while open-ended will have all answers stored in the filter array. The following shows the code snippet for setting the filter in conducting data analysis.

```

foreach ($q as $mapto => $answer){
    $filter = null;
    foreach ($answer as $subans){
        list($val, $no, $qnc, $qtype) = explode('|',$subans);
        $filter_desc[$qnc][] = $val;
        if ($qtype == 'multiple-answer'){
            $filter[] = " FIND_IN_SET('$no', `{$mapto}`) ";
        } elseif ($qtype == 'matrix-answer'){
            $filter[] = "`{$mapto}` = '$no'";
        } else {
            $filter[] = "`{$mapto}` = '$no'";
        }
    }
    $or_filters[] = ' ( ' . implode(' OR ', $filter) . ' ) ';
    $i++;
}

```

Users

Administrators may add, edit, delete or assign user types by using this feature. User list option provide a list of all users created in the system while add new option allows the administrator to create new user accounts. Figure 5-16 shows the snapshot of this feature.

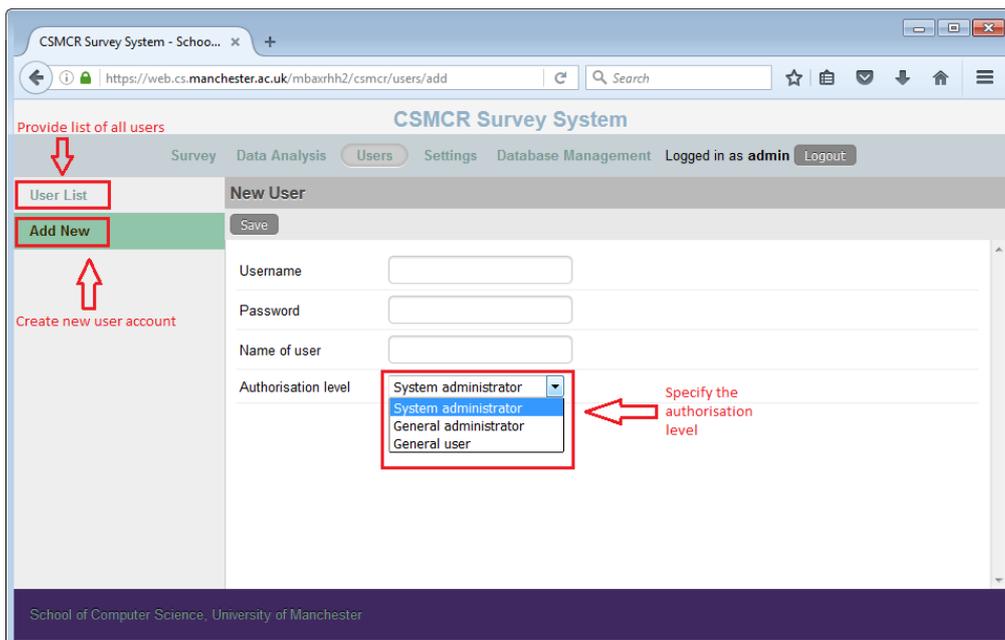


Figure 5-16: Add New User

Settings

Questionnaires management is primarily done in settings. Administrators may view, add, edit or delete questions here. In addition, the type of question, database field where the data will be saved to, custom answer options and ordering of the questions can also be performed here. Figure 5-17 shows the snapshot of this feature.

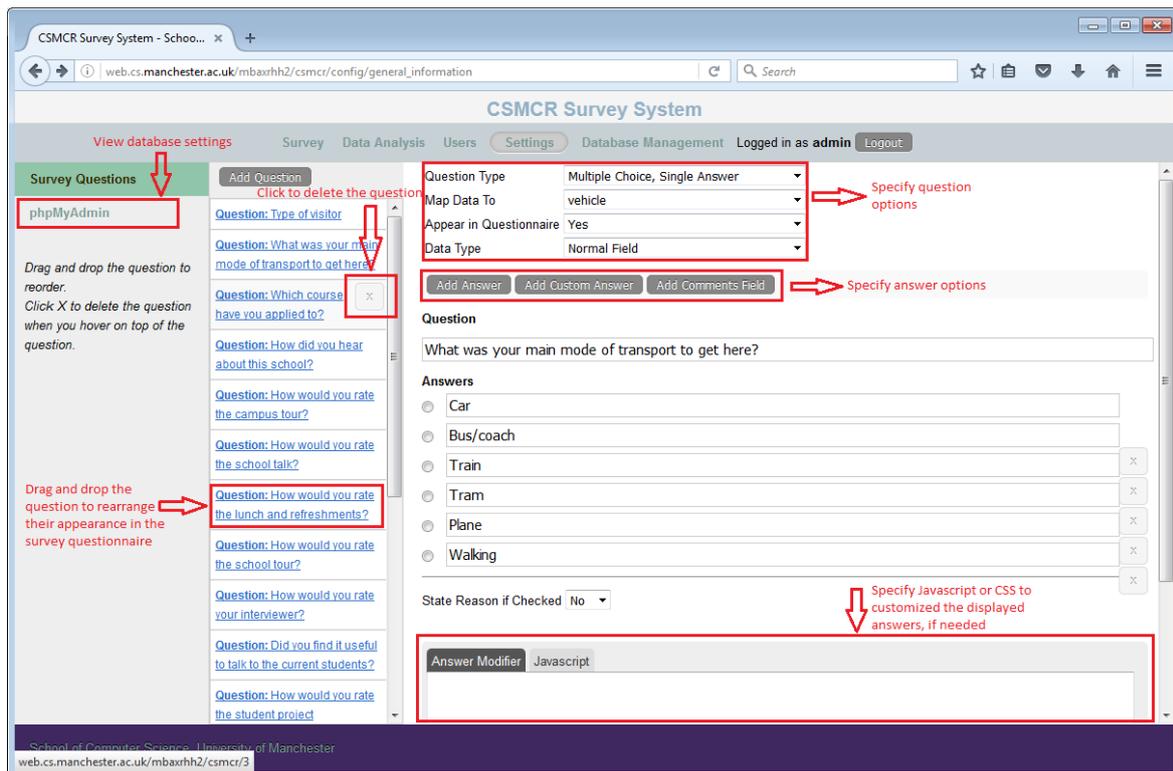


Figure 5-17: Survey Question Settings

Database management

Administrators can backup or restore the survey data using this option. In addition, historic survey data from past surveys that are formatted in CSV can be uploaded here to utilize the system analysis functions. Figure 5-18 shows the snapshot of database management page.

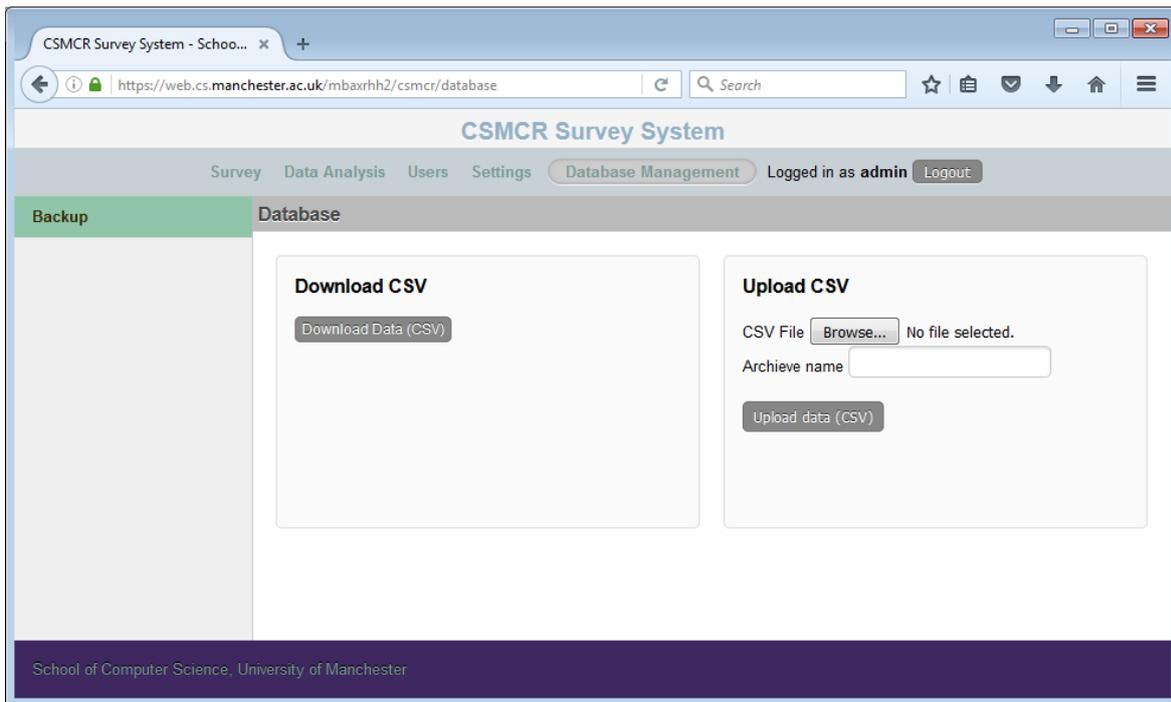


Figure 5-18: Data Backup and Restore

The following code snippet shows how the data was organized into csv format. CSV was chosen as the backup format as it is commonly used and flexible to be converted into other format if necessary.

```
function Download_csv(){
    $query = $this->db->query("SELECT * FROM `survey_gen` ");
    $list = $query->result();

    header('Content-Type: text/csv; charset=utf-8');
    header('Content-Disposition: attachment; filename=data.csv');

    // create a file pointer connected to the output stream
    $output = fopen('php://output', 'w');

    // output the column headings
    foreach ($list[0] as $col => $val){
        $colnames[] = $col;
    }
    fputcsv($output, $colnames);

    $list = $query->result_array();

    foreach ($list as $row){
        fputcsv($output, $row);
    }
}
```

Chapter 6: Evaluation

This chapter will focus on the evaluation of both mobile and web applications. Two types of evaluation were conducted; one for the technical and another for the functional aspect of the system. Technical aspects include features during and after the process of development and integration of components. On the other hand, the functional evaluation covers the project requirements as specified in chapters 3 and 4 of this report.

6.1. Technical Evaluation

Technical aspects that were employed in this application were evaluated for their performance. This is to ensure that both mobile and web application are optimized for operations.

6.1.1. Database

A dataset consisting of sample survey data was created consisting of 200 records (the average turnout for an open day event is around 100 visitors). Queries from the two main tables, `survey_gen` and `questionnaire` were executed using joins and filters commonly created by the web and mobile applications. All queries were performed promptly, mainly due to indexes and optimization when querying (e.g. choice of `IN` and `BETWEEN` in building `WHERE` conditions).

On average, the results show execution time of less than 0.001 seconds for all queries tested with a standard deviation of 0.0002 seconds. Fast query time will help to ensure that the data layer will not be a bottleneck to the system and response times can be expected to be in uniform without much of a variation (i.e. small standard deviation value).

6.1.2. Hardware

Three mobile devices were used in the test, ranging in specification types from the low-range (Amazon Fire 7), medium-range (Tesco Hudl 2) and high-range (Samsung Galaxy Note 4) devices (Figure 6-1). Factors that were considered in this evaluation are the touch screen performance, UI display, battery and processing power.

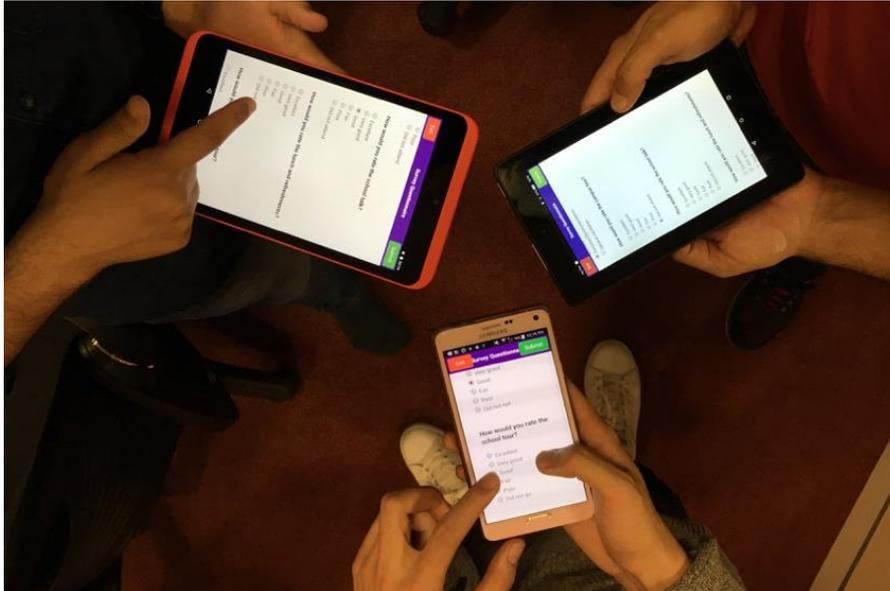


Figure 6-1: CSMCR Survey Mobile App in Various Devices

Table 6-1 below shows the results from hardware evaluation. Based on the result, it is suggested that devices with similar specification as Tesco Hudl 2 to be used as the survey device. The screen size and resolution is suitable for conducting surveys and have more than sufficient processing power and battery for the app to run smoothly for a one-day event.

Table 6-1: Results from Hardware Evaluation for Mobile App

Hardware aspect/Device name	Amazon Fire 7 (59)	Tesco Hudl 2 (60)	Samsung Galaxy Note 4 (61)
Specification range	Low	Medium	High
CPU	Quad-core ARM Cortex A7, 1.3 GHz	Quad-core Intel Atom, 1.83 GHz	Quad-core 2.7 GHz Krait 450 - Snapdragon 805
Screen resolution	1024px x 600px	1920px x 1200px	1440px x 2560px
Screen size	7"	8.3"	5.7"
RAM	1 GB	2 GB	3 GB
Battery size	2980mAh	Not disclosed	3220mAh
Results	At 170ppi, the display is not sharp and sometimes affects when making survey answer selections using the touch screen, especially for radio buttons. Battery is estimated to last for about 8 hours with constant use. Processing time is as expected. The screen size is suitable for survey purposes.	At 273ppi, the display looks sharp and no sensitivity issues in making survey answer selections by using the touch screen. Battery is estimated to last for about 8 hours with constant use. Processing time is as expected. The screen size is suitable for survey purposes.	At 515ppi, the display looks sharp and no sensitivity issues in making survey answer selections by using the touch screen. Battery is estimated to last for about 11 hours with constant app use. Processing time is as expected. The screen size is quite small for survey purposes.

6.1.3. Performance

The quality and performance of the system are largely defined by the quality of its architecture. Poorly planned elements of application architecture can limit or result in a bottleneck to the expected levels of performance. Enhancing performance requires the art of balancing between three main factors that are the response time, CPU usage and throughput. The response time and CPU usage are often regarded as the limiting factors while throughput is the factor to be maximized. Figure 6-2 below shows the relations between these three factors and the targeted area.

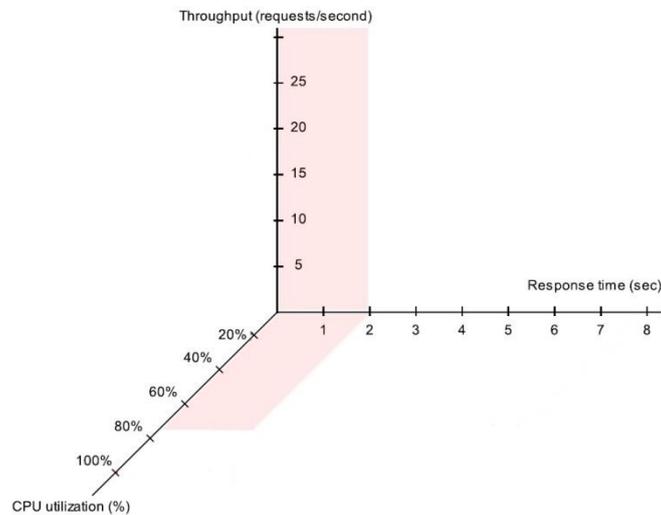


Figure 6-2: Performance trade-offs

The shaded area in the graph represents the targeted limit for each area. The X-axis shows the response time in seconds with the maximum time of 8 seconds. Previous study shows that 8 seconds is the maximum tolerable wait time for average users (62). However, for this project, it is targeted that 90% of the requests to be at the maximum wait time of 5 seconds. On the other hand, the Z-axis shows the CPU utilization. Targeted usage limit is set to be at a maximum of 70%, as anything above it with constant usage will raise the probability of the CPU to malfunction in the long run. In addition, it allows for the additional 30% to be used during unusual peak period. Having both response time and CPU utilization target set, the throughput which is represented by Y-axis was measured.

In order to measure the throughput and overall performance of the system, an artificial workload was created to mirror the real workload that the system would perform when it goes live. The workload was created to resemble real traffic for user observations, system interactions and data processing. Workload model was created to simulate average user interaction with the system (Figure 6-3). Percentage for each workload is estimated based on the priority in the user stories during the project requirements gathering phase.

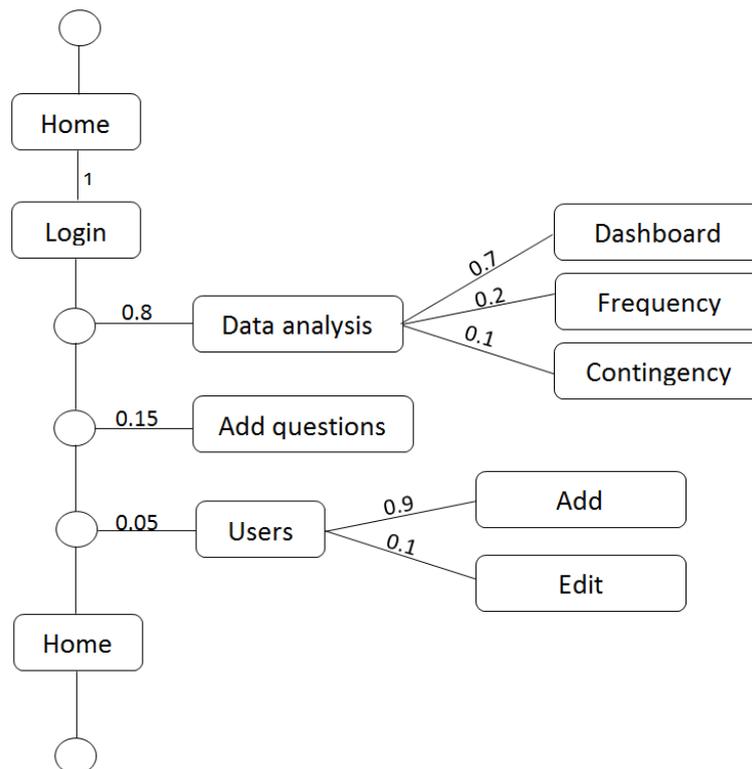


Figure 6-3: Workload Model for Performance Testing

Activities in the form of HTTP requests were executed for each time the test is performed, and the probability of users choosing the action is indicated by the connecting lines. The workload test starts with all users arriving at the home page, and all will have to log on to the system in order to access its service. It is expected that 80% of the users will use data analysis option, in which 70% going for the dashboard, 20% for the frequency and the remaining 10% will go for contingency table analysis. Meanwhile, 15% of the users will use the add questions function and the remaining 5% will use the manage users option. In users option, 90% will use the add function while the remaining 10% will use the edit function. Finally, all users will go back to the home page and leave the application page.

Data for each performance test, such as the average response time and throughput were generated from JMeter. Figure 6-5 below shows the average response times for HTTP requests based on the workload model defined earlier. The set target of at least 90% of all requests took no more than 5 seconds were met as none of the requests took longer than that.

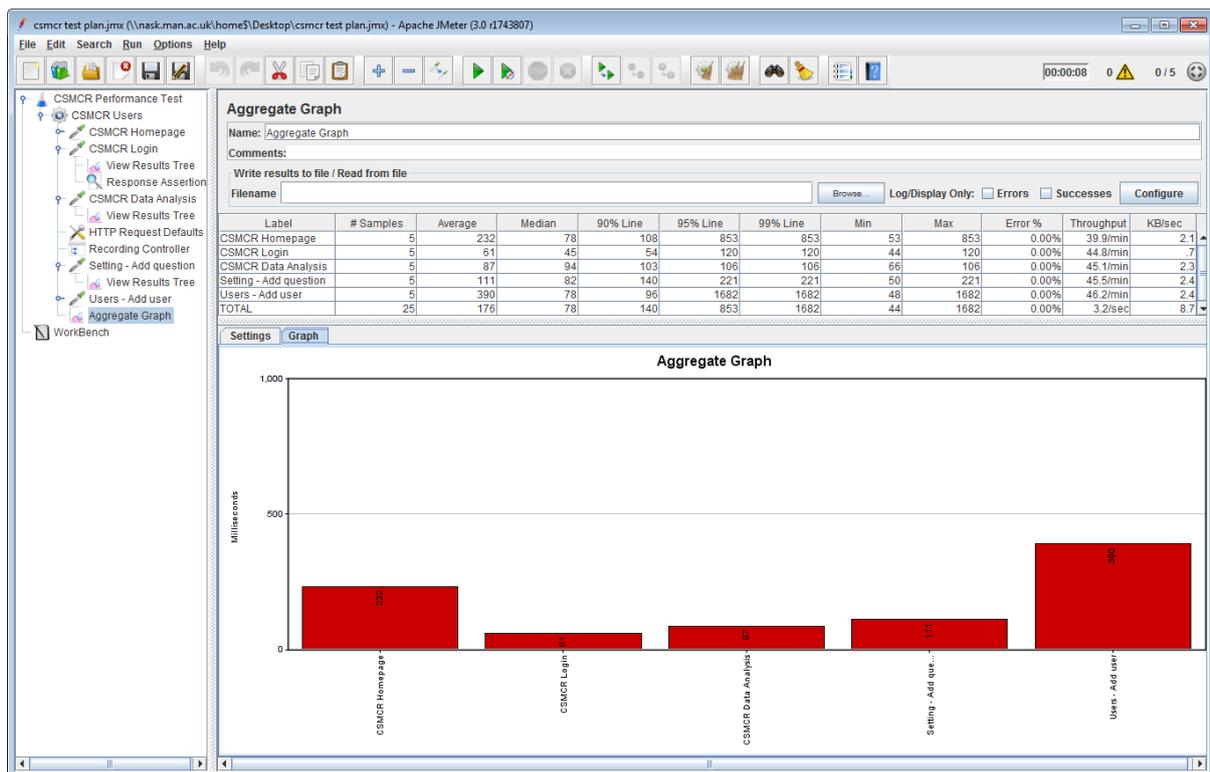


Figure 6-5: JMeter Results in Aggregate Graph

All of the tests were conducted in an isolated WLAN involving a PC to act as the web and database server, 1 Gigabit WLAN switch and a MacBook acting as the load generator. The PC acting as the server has the following specification: CPU Intel® Core™ i5-2467M Processor, 3MB L3 Cache, RAM 4GB DDR3, HDD 500GB SATA, OS Windows 8.1. Although the system will be deployed in the school infrastructure, the testing was not done using the school server as user agents need to be installed at the server and this will breach the school security policy. However, it is expected that the test will produce better result using the school server as the hardware specification is much better than the test server PC used.

6.2. Functional Evaluation

Functional evaluation with system owners and potential users was conducted to ensure that the system meets its intended objectives. Types of system users include system administrators, survey analysts and survey respondents. Therefore, the evaluation must take into account these various types of users and the tasks that they will perform using the system. The project requirements and system usability were evaluated by verifying the system functionalities with use cases and conducting usability test with general users.

6.2.1. Requirement

Evaluation of project requirements was based on use cases created for the project. These use cases provide guidelines and act as an indicator for project requirement's fulfilment. Verification was done with system owners for each of these use cases as to how the system is able to perform the functions.

Table 6-2 below shows the results.

Table 6-2: Requirement Evaluation by Use Case

No.	Use case	Result
1.	Manage users	The system provides interface to view, add, edit and remove users from the system. Each user can be assigned to 3 different user groups with each group having its own set of privileges.
2.	Manage survey questionnaires	The system provides interface to view, add, edit and remove questions from the system. Question responses can be assigned with specific types according to administrator needs.
3.	Manage database	The system provides interface for mapping questions to database fields and administrator may also use the phpMyAdmin interface for advanced operations.
4.	Backup data	Administrator may backup or upload data in bulk using the system

		interface provided.
5.	Upload data	Administrator may upload the offline-stored data in mobile app to the database server in bulk or one by one.
6.	Query data	The system provides interface for generating data analysis dashboard, frequency table and contingency table with additional option of data filtering. Users may also view data by querying for specific survey ID or date.
7.	Generate report	Dashboard, frequency and contingency table generated in data analysis can be exported to Microsoft Excel® format.
8.	Answer survey	Respondents may use either the mobile app or web app to view and answer the survey.

The system developed has successfully fulfilled all project requirements. Additional features such as backup solution for survey questionnaires using web interface were added as the need emerged during the development phase.

6.2.2. Usability

At present, it is no longer sufficient for a system to be just functional. Other factors that typical users would look at include ease of use, visual appeal, and the ability to engage (64). Therefore, usability evaluation was conducted to measure these factors. In addition, usability evaluation supplements the requirement evaluation by taking into account users' experience in using the system's functions. Systems with great usability will attract users to actually use the system and benefit from it.

Evaluations were conducted by constructing prominent tasks that will help to investigate the usability of the system's main functionalities. These tasks were created in the form of user stories and acceptance tests. Moderating techniques such as Concurrent Think Aloud and Retrospective Probing were employed while the evaluation was conducted and the results were recorded (65).

Fifteen students were recruited to take part in the process. Eleven of them have a background in computer science while the other four have a background in law and business administration. System and user guidelines were explained before they started the tasks.

Table 6-3 below shows the summary of usability evaluation with system users.

Table 6-3: Summary of Usability Evaluation with Users

No.	Tasks	Issues/concerns	Improvement made
1.	You are a prospective student visiting the school during an Open Day. Answer the survey questionnaire using the mobile device provided.	Hard to touch and make selection at the radio button. Small font size made it hard to read.	Touch sensitivity is part of hardware limitation. However, the buttons were made bigger and users can now touch the answer option to make the selection. Font size was also made bigger.
2.	As an administrator, you want to upload all the survey data stored in the mobile device.	None	None
3.	You want to get information on how many visitors have applied for Artificial Intelligence course. Use the web application to do this.	Need to refer to the questionnaire in order to know which questions have the answer option for Artificial Intelligence.	When selecting for any question in the frequency table, a list of all answer options for that particular question will appear automatically thus making it easier to choose the right question.
4.	You want to get the number of visitors who	Some users got confused on whether	There are two ways to get the count and it is probably the

	initially wanted to reject their offer but changed their mind after visiting the Open Day. Use the web application to do this.	they need to use the filter option or not in order to get the count.	reason for the confusion. If frequency table is used, user need to set filter but if contingency table is used then filtering is not needed since both answers will be displayed. The issue is more towards understanding the difference between those two tables. Improvements were made in user guidelines to clearly explain this.
5.	Generate a report to show the overall visitors' satisfaction.	No generate report options were found in the system.	Users need to first perform data analysis, before they can export the data to Microsoft Excel© format.
6.	A new survey administrator joins the team. Create a new profile so that he could access the system.	The task does not specify which user group level the new survey administrator should be. Nonetheless, the task can be conducted easily once the information was provided.	None. The issue/concern was due to the lack of information in the task instruction.
7.	Find all survey records for 1 st September 2016.	None.	None.
8.	Add a new survey question to the system.	Most users did not update the mobile app to reflect the new added question.	A pop-up message box is added as a reminder to update the mobile app.

9.	Change the order of appearance of question in the survey.	No one knew how to do it.	An info tip was added to inform users how to do it.
10.	The database became corrupted. Upload the survey data by using the backup provided.	None	None

After the improvements were made to the system, the same tasks were repeated to ensure that there are no new issues arising due to the improvement. All students managed to complete them successfully.

Usability evaluation is an important stage in functional evaluation as it helps to determine if the system is user-friendly enough and can attract potential users to utilize it. A system can only be successful in achieving its objectives if the users are comfortable and not reluctant to use it.

6.3. Security Evaluation

One of the main concerns in any system development, especially for systems that are accessible through the network, is security. Systems with weak security can be the point of attack or illegal access to the private network infrastructure and harm other critical systems. In order to avoid any security issues, security evaluation was conducted to identify any security vulnerabilities and patch the system accordingly. There are three types of tests that were performed to evaluate the security performance of the survey system and they include manual and automated source code analysis.

Table 6-4 below shows the type of security test conducted and their results.

Table 6-4: Types of Security Test and Result

Type	Description	Result
Manual technical review and testing	Review and test was conducted based on security checklist for web application from SANS Institute (66) and Certified Secure (67).	Based on scopes that are applicable to the project, the only lacking aspect is the use of SSL/TLS. This can be fixed when the system is deployed in server with certified public key certificate.
Automated source code analysis	SWAAT tool was utilized for this purpose. SWAAT helps to identify potential dangerous functions and strings in the source code, explain how they could be dangerous and how to mitigate the potential risks (68).	One of the potential threat found is the source code contains the default database password that is used for the system to access the database. To resolve this issue, the password is moved from the source code to configuration file. Other threats that were found turned out to be false positive.

For testing mobile app security, an automated mobile app vulnerability tool named Appvigil was used (69). Appvigil scans the Android apk file to identify any security and privacy vulnerabilities. At the end of scanning, a report was generated and outlined the exact location of those vulnerabilities and provided recommendation on how to fix them. Based on the report generated (Figure 6-6), no vulnerabilities were found and the threat level was rated as 0%.

Results of both web application and mobile app security tests show the system is secured and does not have any security issues or high-risk vulnerability. Therefore, the system is ready and can be safely deployed in any network environment.

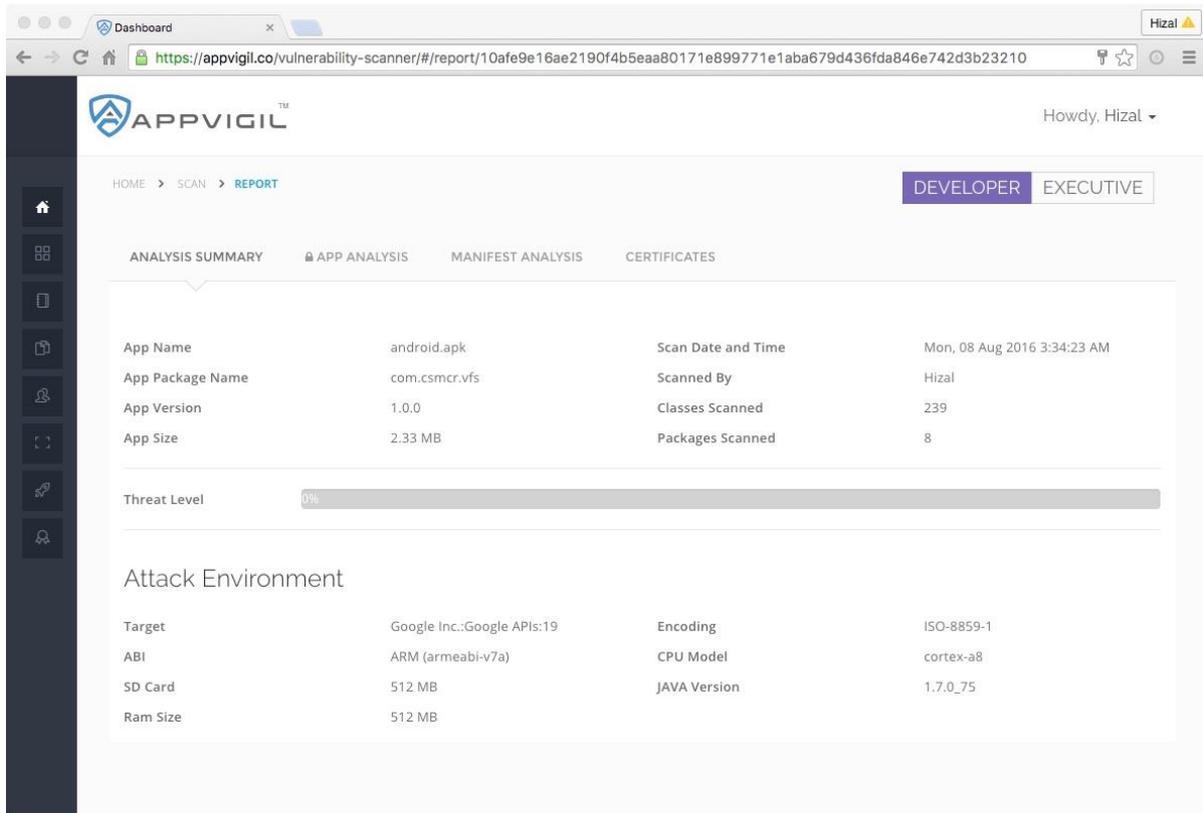


Figure 6-6: Appvigil Vulnerability Scanner Report

6.4. Measure of Effectiveness

A quantitative measurement method was developed in order to evaluate the effectiveness of the survey system in fulfilling its objectives. The extent of how much the survey system meets its intended objectives and goals can be effectively measured by comparing values for a set established parameters for existing survey practice and the developed system. As stated in the introduction chapter of this report, among the main concerns when conducting the surveys are resource issues (i.e. time, manpower) and analysis capabilities. Therefore, the measurement of effectiveness should use these criteria in its evaluation.

Table 6-5 below provides a list of parameters that were used for evaluating the system effectiveness.

Table 6-5: Parameters for Measurement of Effectiveness

Measurement of Effectiveness	
Resource	Analysis of data
Time taken to answer survey	Analysis generation rate
Time taken for data entry	Number of analysis types
Cost	Response accuracy rate

A mock survey session consisting of 10 students was conducted at the Kilburn’s Tootill Lab to measure these parameters (Figure 6-7). Group 1 consisting of 5 students was asked to complete the survey using paper format while the remaining 5 students in Group 2 were asked to use the mobile app. Both groups were then asked to conduct a series of data analysis. Group 1 used Microsoft Excel® for this task while Group 2 used the developed survey system.



Figure 6-7: Mock Survey Session for Evaluation

Table 6-6 below shows the results from Measurement of Effectiveness.

Table 6-6: Results from the Measure of Effectiveness (Average for 10 Respondents)

	Parameters	Paper format	CSMCR Survey system
Resource	Average time taken to answer survey	57 seconds	84 seconds
	Average time taken for data entry	2.5 minutes/response	2 seconds to upload all response
	Average cost	£0.08/response (paper and photocopying)	£40/tablet for unlimited number of survey.
Analysis	Average time taken to produce each response frequency analysis	23 seconds/closed question 2 minutes/open question	2 seconds for all questions
	Number of analysis types	1 – One way table	4 – One way table, two way table, higher way table, graphs
	Response accuracy rate	97% (1 student mistakenly entered different answer when transferring from paper to Ms Excel)	100%
Total resources used for every 100 respondents		4.2 man-hours for data entry. 12.3 man-hours for data analysis.	2 seconds for data uploads. 2 seconds for data analysis.

Based on the result, for an open day event with 100 survey respondents, the CSMCR Survey System is expected to save at least 16.5 man-hours as compared to paper-based survey questionnaires. Assuming that no breaks were taken during the process of data entry and data analysis for paper questionnaires, this is equivalent to 2 man-days saved per event. The time taken to complete survey using mobile app were slightly higher as the typing speed in virtual keyboard for open-ended questions is taking a longer time as compared to written the answer on paper.

Errors in data entry process that occurred in paper questionnaires can also be avoided. These errors would affect the accuracy hence the information obtained from any data analysis can be misleading. Ultimately, judgements that were made for improvements may result into something in the opposite. The inaccuracies with paper questionnaires as found in this exercise were in line with other similar research findings (15), (70). In addition, more types of analysis can be conducted with the survey system thus providing more perspective into the data as compared to only a single type of analysis when using paper questionnaires.

On the other hand, the cost for running the survey system will be high initially before greatly reduced in the long run. The initial cost to obtain one mobile device will be compensated after approximately 5 surveys ($\text{£}0.08 \times 100 \text{ respondents} \times 5 \text{ events} = \text{£}40$).

Chapter 7: Conclusion

Upon critical assessment on the survey requirements for the School of Computer Science, the project was concluded with the development of CSMCR Survey mobile app and web application (Figure 7-1). The system is intended to automate the manual process currently done when conducting surveys. It is designed with minimal changes to the existing practice and process flows to allow for easy adoption by the potential users and system administrators. By systematically collecting survey data, it will help to save resources and enable new knowledge to be obtained from the data analysis.

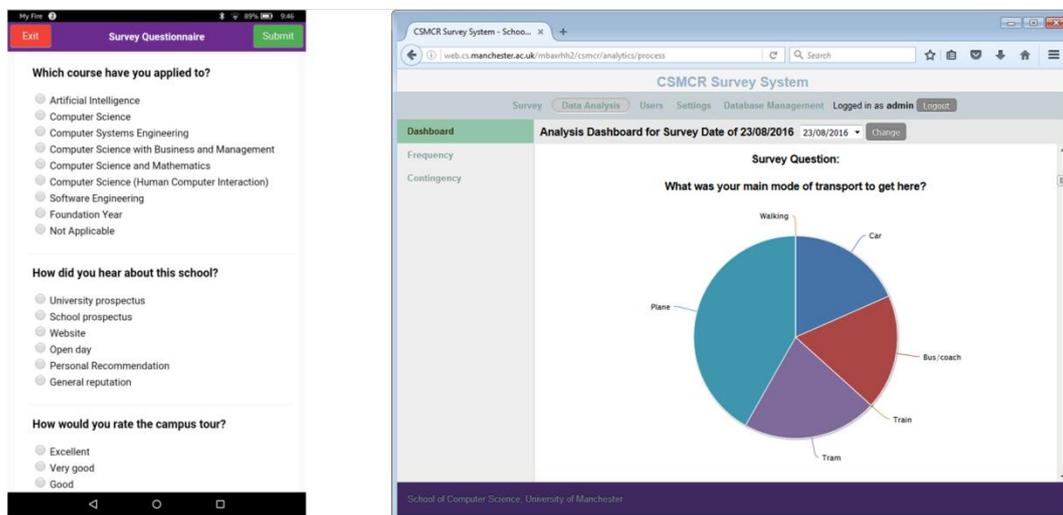


Figure 7-1: CSMCR Survey System

In addition, visitors will be able to conveniently provide their feedback through an easily accessible application. Moreover, feedback data obtained from the event will be more accessible and engaging. As a result, improvement on critical areas can be identified immediately. Based on various aspect of evaluation conducted, the system has successfully met its aim and objectives. It is hoped that the successful implementation of this project will make a good case study for various survey system development in the future.

This chapter will highlight the project summary, project achievement, issues encountered while conducting the project and outline future work that can be conducted to further strengthen the system.

7.1. Project Summary

This dissertation describes a complete system development process specifically for a survey system; right from the initial concept, survey form, mobile app and web application system development. The following describes the summary for each chapter.

- **Background** – various technologies and development methods were discussed. Background research on technologies, methods and techniques helped to determine the best approach for the project. Aspects such as project requirements, technology evolutions and target user needs primarily influenced the decision making process. These selections affect the characteristics of web and mobile environment and features that were used to develop the application.
- **Method** – analysis of targeted user needs, defining requirements and setting the overall project scope. Proper understanding of requirements and needs helped in formulating suitable design and implementation strategy. It also helped in understanding the limitations such as resource constraints. A final set of methodology, technology and technique was then adopted and adapted for the project specification in order to proceed with the design and implementation stage.
- **Design** – involved planning for the overall architecture, features and functionality of the system. This is where project requirements played a major role. Some of the requirements were found to be vague and additional clarifications were needed. It was great to have the project supervisor and stakeholders readily available to further explain the particular requirements and how they fit into the initial design. Use cases and UI were designed to provide the overall system logic and flow. Verification of the design was necessary before proceeding to implementation phase as it was easier for changes to be made at this stage as opposed to the later stages.
- **Implementation** – This phase took the most time and effort. The project design served as the fundamental start to the implementation. Modification and

enhancement were then made where appropriate. The use of Integrated Development Environment (IDE) such as Android Studio helped with the coding and debugging process. Availability of common tools such as IDE plugins, compilers and libraries facilitated the programming process. In addition, when facing difficulties, various documentations and help were available through the use of product knowledgebase and forums. One important milestone in this phase was the establishment of questionnaire programming logic. This helped to expedite the implementation of other features such as data analysis and questionnaire management.

- **Evaluation** – the technical, functionality and security of the system were evaluated. These evaluations helped to test the quality and durability of the system when it was deployed. It was also an effective and efficient way to find any defects or vulnerabilities of the system so that improvement could be made before actual deployment.

7.2. Project Achievement

Among the project achievements are:

- **Systematic and comprehensive survey system** – Based on the initial project description, the project aims for either an Android app or web application in its delivery. However, upon research on suitable tools and technologies for fulfilling the project requirements, it was concluded that both approach is needed, with android app focusing on survey data collection and web application focusing on data analysis, general survey maintenance and overall system management. The project started from scratch and both apps were completed within the stipulated time period.
- **Work offline** – One of the major advantages of this system is that the mobile app can work offline. The survey using the mobile app can be conducted anywhere and without network data plan. Survey responses are stored locally in the device and can

be uploaded to the server when network connection is available. Other similar systems in the market require constant network connection thus limiting the survey to be conducted in places with limited connectivity.

- **'Zero-maintenance'** – The project deliverables state that the system should not rely on any hardware/software that is outside of the school infrastructure. This requirement is fulfilled as the system was compatible with the existing infrastructure and can be deployed from the school web and database server and the mobile app can be used with the existing Amazon Fire® 7 and Tesco Hudl® 2 tablet owned by the school. The system also requires 'zero-maintenance' for both software and hardware. No monetary cost was incurred in developing the system. Open-source software was used to develop and run the system thus no software license was needed.
- **Run independently** – The system can run independently once the initial set-up is done and would be able to continue running without the need of any technical expertise. Common issues in managing survey questionnaires such as adding, removing or editing questions can all be done using the system interface. Updated questions can be pushed to the mobile apps without the need to change the source code.

7.3. Issues Encountered

While conducting the project, several issues were encountered. These issues are mainly due to technical difficulties and problems in project management. The followings highlights issues that have been encountered and how it was dealt with.

- **Dynamic and complex** – As the system involves dynamic content, the system logic needs to be sound and complete. Various types of user input and actions should be anticipated, not only the valid actions but also invalid and wrong input. The processing for each input type would be different in data analysis and when adding a

new question, thus complex data handling method was needed. To confront these issue, similar systems available on the market was referred to and the logic behind the applications were anticipated. Trial-and-error method was used until the correct logic was found.

- **Too many options** – In developing the system, various methods and strategies are available to provide the framework and guide the overall process. Popularity of Agile methodologies and hybrid mobile development is on the rise with various success stories. In conducting this project, aside from the coding, the challenges include choosing the right method and strategy to ensure successful development. They were overcome by thorough research on literatures and best practices. The project strategy and methodology were selected carefully, and scholars and professionals in the field were consulted for their feedback.
- **Skills** – Acquiring proper skills for programming was also a big challenge. Although the system mainly used common web programming languages, the integration between web and mobile application is something new.
- **Memory Management** – Hybrid mobile apps present a challenge in terms of access towards device features that require the use of native code and memory management. Hybrid mobile apps call for the device WebView, which by itself uses substantial amount of memory. Therefore, efficiency is the main factor considered in the development and the system was developed with minimal memory usage so that it will run smoothly on any device. This issue may present one of the most challenging aspects of the project, but it is also the most interesting aspect of the project.
- **Choice of UI** – In hybrid app, choice of UI and functionality are also limited due to compatibility concerns with multiple platforms. Instead of utilizing the generic design commonly used in Android app, the app needs to be able to adapt to design commonly used for the web. This may affect the users as they may anticipate the common design and functionality usually found in the platform and it might take

some time to get them accustomed to the new design. For example, back button is commonly used for Android app but not for the iOS. To tackle this issue, simplistic design is used and colour-coded button was employed to highlight the common course of action for the user.

- **The right balance** – Project planning and time management brought the greatest challenge. For a project that aimed to develop a system as big as Google forms and Survey Monkey, it must be managed carefully. Employing Agile development method was an excellent choice as iterations and small releases ensured that the project workload was manageable and was always moving forward. Conducting the project during academic semester period brought the challenge of determining priority between coursework and project progress. Both aspects were given due attention and although the project progress was slightly behind schedule during the semester period, it gained momentum after the semester ended.
- **Project Planning** – Because the project was the work of a single programmer, the burden of ‘knowing it all’ has to be carried alone. The project requirements need the system to be maintenance-free thus consideration about the system should not only be about the existing environment, but necessitates anticipation of future problems and how the environment would change over time. Therefore, planning for a robust system is top priority and technology selection plays an important part in the project development as it is rapidly changing. Literature review helps to recognize some the potential issues and ways to mitigate them.

7.4. Future Work

Although a considerable effort has been made in developing the system, due to time and resource constraints, some of the functionalities can be further enhanced such as the followings:

- **Survey questionnaire** – Currently, the system is only equipped to handle a single set of questionnaire questions. The school may want to utilize a few sets of questionnaire based on the event type (e.g. Technology fair, Exhibition). The current system requires the administrator to either modify the questions through the system or upload the saved questionnaire into the database to replace the existing ones. Ideally, this should be done by just selecting the saved questionnaires type from the system. In addition, for each questionnaire, answer verifications can also be added to further validate the respondent's feedback.
- **Visualization of data** – Data visualization from analysis can be done in a more dynamic and interactive way. Better representation of data will greatly improve discovery of information, help in decision-making and provide explanation of certain situations. The existing system provides count figures in the form of tables and graphs but additional statistics can be added such as the measures of central tendency (mode, mean or median) and indicators of statistical variance or dispersion (error bars or box plots). The use of a more interactive form of data visualization can promote survey data exploration and lead to greater findings. Automated report generation can also be enhanced in which common data analysis can be generated and pre-formatted for printing and quick dissemination.
- **Automated Text Mining** – For qualitative and open type of questions, the system administrator will need to extract this information manually. Improvement can be made by introducing automated text mining. Frequent keywords found in answer texts can be captured and this can be used as a basis in improving the question or use it as the answer options in close type of questions. Data analysis will be easier

when dealing with quantitative type of data. In addition, answer options will not rely on assumptions but rather be based on real answers provided by previous respondents.

- **User Interface** – The overall UI design can also be improved. Little time was spent on this, as the focus was on developing and creating the essential functions. A better designed UI can help to promote the system and improve its overall usability.

Bibliography

1. **Cooper, S. Barry and Leeuwen, Jan Van.** *Programmer's Handbook for the Manchester Electronic Computer Mark II: Local Programming Methods and Conventions*, In Alan Turing: *His Work and Impact*. Boston : Elsevier, 2013. Pages 465-478.
2. **Fink, A.** Survey Research Methods. [book auth.] Penelope Peterson, Eva Baker and Barry McGaw. *International Encyclopedia of Education (Third Edition)*. Oxford : Elsevier, 2010.
3. Choosing a university course: applicants' top five factors. [Online] [Cited: 16 July 2016.] <http://www.thecompleteuniversityguide.co.uk/courses/choosing-a-course/choosing-a-university-course-applicants%27-top-five-factors/>.
4. **Faulkner, C.** *The Essence of Human-Computer Interaction*. London : Prentice Hall, 1998.
5. **Manski, Charles F.** Questionnaires: Collection of Subjective Expectations Data. [book auth.] James D. Wright. *International Encyclopedia of the Social & Behavioral Sciences (Second Edition)*. Oxford : Elsevier, 2015.
6. *Who's using PDAs? Estimates of PDA use by health care providers: a systematic review of surveys.* **Garrity, C. and Emam, K.** 2, 2006, J Med Internet Res, Vol. 8, p. e7.
7. *Data collection outcomes comparing paper forms with PDA forms in an office-based patient survey.* **Gallagher, J. M., et al., et al.** 2, 2008, Ann Fam Med, Vol. 6, pp. 154–160.
8. *Comparison of a computer-administered and paper-and-pencil-administered questionnaire on health and lifestyle behaviors.* **Vereecken, Carine A. and Maes, Lea.** 4, April 2006, Journal of Adolescent Health, Vol. 38, pp. 426-432.
9. *Using an iPad in research with children and adolescents.* **Linder, L. A., et al., et al.** 2, 2013, J Spec Pediatr Nurs, Vol. 18, pp. 158–164.
10. *Experience with digital entry of national iodine survey data in Senegal.* **Siekman, K., et al., et al.** 7, 2012, Afr J Food Nutr Sci, Vol. 12, pp. 6987–7000.
11. *The information aged: examination of university students' attitudes towards personal digital assistants (PDAs) usage in terms of gender, age and school variables.* **Alzaidiyeen, N., et al., et al.** 3, 2011, TOJET, Vol. 10, pp. 287–295.
12. *Using personal digital assistants in post-conflict health surveys: potential and constraints.* **d'Harcourt, E. and Mulumba, F.** 3, 2008, Glob Public Health, Vol. 3, pp. 326–342.
13. *Elementary school students' attitudes and self-efficacy of using PDAs in a ubiquitous learning context.* **Tsai, P. -S. and Tsia, C. -C.** 3, 2010, Australas J Educ Tech, Vol. 26, pp. 297–308.
14. *A comparison of paper-based questionnaires with PDA for behavioral surveys in Africa: findings from a behavioral monitoring survey in Kenya.* **Tegang, S. P., et al., et al.** 1, 2009, J Health Inform Dev Ctries, Vol. 3, pp. 22–25.
15. *Using computer-assisted survey instruments instead of paper and pencil increased completeness of self-administered sexual behavior questionnaires.* **Spark, Simone, et al., et al.** 1, 2015, Journal of Clinical Epidemiology, Vol. 6, pp. 94-101.
16. *How to improve the validity of sexual behaviour reporting: systematic review of questionnaire delivery modes in developing countries.* **Langhaug, L. F., Sherr, L. and Cowan, F. M.** 3, 2010, Trop Med Int Health, Vol. 15, pp. 362–381.
17. *A systematic review and meta-analysis of quantitative interviewing tools to investigate self-reported HIV and STI associated behaviours in low- and middle-income countries.* **Phillips, A. E., et al., et al.** 2010, Int J Epidemiol, Vol. 39, pp. 1541–1555.

18. **Mendoza, Adrian.** Chapter 8 - Mobile App or Mobile Web: The Big Debate. [book auth.] Morgan Kaufmann. *In Mobile User Experience*. Boston : s.n., 2014, pp. 161-173.
19. *Taxonomy of Cross-Platform Mobile Applications Development Approaches.* **El-Kassas, Wafaa S., et al., et al.** 23 October 2015, Ain Shams Engineering Journal.
20. *Potential advantages and disadvantages of multiplatform development frameworks – a vision on mobile environments.* **Corral, L., Janes, A. and Remencius, T.** 2013, *Procedia Comput Sci*, Vol. 10, pp. 1202–1207.
21. *The Impact of Mobile Platforms in Obstetrics.* **Valente, S., et al., et al.** 2013, *Procedia Technology*, Vol. 9, pp. 1201-1208.
22. **Klima, P. and Selinger, S.** Towards platform independence of mobile applications. [book auth.] R. Moreno-Díaz, F. Pichler and A. Quesada-Arencibia. *Computer aided systems theory – EUROCAST 2013*. Berlin Heidelberg : Springer, 2013, Vol. 8112, pp. 442–449.
23. *Component-based mobile web application of cross-platform.* **Biao, P., Kun, X. and Lei, L.** 2010. 2010 IEEE 10th international Conference on Computer and Information Technology (CIT). pp. 2072–7.
24. **Version One.** State of Agile Development. [Online] 2016. [Cited: 17 April 2016.] <https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf>.
25. *What do we know about agile software development?* **Dybå, T. and Dingsøy, T.** 5, 2009, *IEEE Softw.*, Vol. 26, pp. 6–9.
26. *Agile modeling, agile software development, and extreme programming: the state of research.* **Erickson, J., Lyytinen, K. and Siau, K.** 4, 2005, *J. Database Manag.*, Vol. 16, pp. 88–100.
27. **Hughes, Ralph.** Chapter 2 - Primer on Agile Development Methods. [book auth.] Morgan Kaufmann. *In Agile Data Warehousing for the Enterprise*. Boston : s.n., 2016, pp. 13-29.
28. **Beck, K.** *Extreme Programming Explained: Embrace Change*. s.l. : Addison-Wesley Professional, 2000.
29. **Schwaber, Sutherland, Schwaber, K. and Sutherland, J.** The scrum guide. [Online] 2011. [Cited: 20 April 2016.] <http://scrum.org/>.
30. **Benfield, Deemer, et al., et al.** The scrum primer. [Online] [Cited: 21 April 2016.] http://www.scrumprimer.org/scrumprimer20_small.pdf.
31. **Petersen, K.** Is lean agile and agile lean? a comparison between two software development paradigms. [book auth.] A. H. Dogru and V. Bicer. *Modern Software Engineering Concepts and Practices: Advanced Approaches*. 2010, pp. 19–46.
32. *Exploring the sources of waste in kanban software development projects.* **Ikonen, Kettunen and Abrahamsson, Oza.** s.l. : IEEE, 2010. 36th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA). pp. 376–381.
33. **Poppendieck, M. and Poppendieck, T.** *Lean Software Development: An Agile Toolkit*. Boston : Addison-Wesley Longman Publishing Co., Inc., 2003.
34. **Abrahamsson, Pekka, et al., et al.** *Agile software development*. Vuorimiehentie : VTT Technical Research Centre of Finland, 2002. 478.
35. **Abrahamsson, P., Oza, N. and Siponen, M. T.** Agile software development methods: a comparative review. [book auth.] T. Dingsøy, T. Dybå and N. B. Moe. *Agile Software Development*. Berlin : SpringerVerlag, p. 31V59.
36. *Empirical studies of agile software development: A systematic review.* **Dybå, Tore and Dingsøy, Torgeir.** 9-10, August 2008, *Information and Software Technology*, Vol. 50, pp. 833-859.

37. **International Organization for Standardization (ISO)**. *Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability*. 1998.
38. **Olmsted-Hawala, Erica, Holland, Temika and Quach, Victor**. 3 - Usability Testing. [book auth.] Andrew Schall and Jennifer Romano Bergstrom. [ed.] Jennifer Romano Bergstrom, Andrew Jonathan Schall and Morgan Kaufmann. *Eye Tracking in User Experience Design*. Boston : s.n., 2014, pp. 49-80.
39. **Nielsen, J.** Usability 101: Introduction to Usability. [Online] 2012. [Cited: 10 June 2016.] <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>.
40. **Rosenzweig, Elizabeth**. Chapter 7 - Usability Testing. [book auth.] Morgan Kaufmann. *Successful User Experience: Strategies and Roadmaps*. Boston : s.n., 2015, pp. 131-154.
41. *Subjective rating scales: science or art?* **Annett, J.** 2002, *Ergonomics*, Vol. 45, pp. 79-102.
42. *Current practice in measuring usability*. **Hornbæk, K.** 2006, *Int. J. Hum. Comput. St.*, Vol. 64, pp. 79-102.
43. *A methodology for evaluating the usability of audiovisual consumer electronic products*. **Kwahk, J. and Han, S. H.** 2002, *Appl. Ergon.*, Vol. 33, pp. 419-431.
44. *Ergonomics in consumer product evaluation: an evolving process*. **Butters, L. M. and Dixon, R. T.** 1998, *Appl. Ergon.*, Vol. 29, pp. 55-58.
45. *Towards a practical method of user interface evaluation*. **Johnson, G. I., Clegg, C. W. and Ravden, S. J.** 1989, *Appl. Ergon.*, Vol. 20, pp. 255-260.
46. *The application of ergonomics to consumer product evaluation*. **Rennie, A. M.** 1981, *Appl. Ergon.*, Vol. 12, pp. 163-168.
47. *Embracing change with extreme programming*. **Beck, K.** 10, 1999, *Computer*, Vol. 32, pp. 70-77.
48. **Smith, Jerry A.** Six Types of Analyses Every Data Scientist Should Know. [Online] [Cited: 20 April 2016.] <https://datascientistsinsights.com/six-types-of-analyses-every-data-scientist-should-know/>.
49. *Effects of questionnaire length, respondent-friendly design, and a difficult question on response rates for occupant-addressed census mail surveys*. **Dillman, D. A., Sinclair, M. D. and Clark, J. R.** 3, 1993, *Public Opinion Quarterly*, Vol. 57, pp. 289-304.
50. *The purpose, design and administration of a questionnaire for data collection*. **Marshall, Gill.** 2, 2005, *Radiography*, Vol. 11, pp. 131-136.
51. *Designing and Implementing Internet Questionnaires Using Microsoft Excel*. **Black, Iain R., et al., et al.** 2, 2005, *Australasian Marketing Journal (AMJ)*, Vol. 13, pp. 61-72.
52. *Agile Requirements Engineering Practices: An Empirical Study*. **Cao, Lan and Ramesh, Balasubramaniam Ramesh.** 1, 2008, *IEEE Computer Society*, Vol. 25, pp. 60-67.
53. **Reenska, Trygve.** XEROX PARC 1978-79. *Trygve/MVC*. [Online] [Cited: 30 July 2016.] <https://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>.
54. **Trowbridge, D.** *Enterprise Solution Patterns Using Microsoft .Net: Version 2.0 : Patterns & Practices*. s.l. : Microsoft Corporation, 2003.
55. **The Apache Software Foundation.** Architectural Overview of Cordova Platform. [Online] [Cited: 28 July 2016.] <https://cordova.apache.org/docs/en/latest/guide/overview/>.
56. **Hales, W.** *HTML5 and JavaScript Web Apps*. 1st Edition. s.l. : O'Reilly Media, 2013.
57. **Google Inc.** Starting an Activity. *Android Developers*. [Online] [Cited: 20 July 2016.] <http://developer.android.com/training/basics/activity-lifecycle/starting.html>.
58. **MariaDB.** MariaDB versus MySQL - Compatibility. [Online] 2016. [Cited: 2 August 2016.] <https://mariadb.com/kb/en/mariadb/mariadb-vs-mysql-compatibility/>.

59. **GSM Arena.** Amazon Fire 7 - Full tablet specification. [Online] [Cited: 20 August 2016.] http://www.gsmarena.com/amazon_fire_7-7609.php.
60. **AOL Tech.** Tesco Hudl 2 Spec. *engadget*. [Online] [Cited: 20 August 2016.] <https://www.engadget.com/products/tesco/hudl/2/specs/>.
61. **GSM Arena.** Samsung Galaxy Note 4 - Full Phone Specification. *GSM Arena*. [Online] [Cited: 20 August 2016.] http://www.gsmarena.com/samsung_galaxy_note_4-6434.php.
62. *Analysis and Evaluation of Web Application Performance Enhancement Techniques.* **Jugo, I., Kermek, D. and Mestrovic, A.** Toulouse : ICWE 2014, 2014. 14th International Conference. pp. 40-56.
63. **Apache Software Foundation.** Apache JMeter™. [Online] [Cited: 3 August 2016.] <http://jmeter.apache.org>.
64. *Evolution of the Mobile Enterprise App: A Design Perspective.* **Newhook, Richard, et al., et al.** 2015, Procedia Manufacturing, Vol. 3, pp. 2026-2033.
65. **U.S. Department of Health & Human Services.** Running a Usability Test. [Online] [Cited: 3 August 2016.] <https://www.usability.gov/how-to-and-tools/methods/running-usability-tests.html>.
66. **Bayse, Zemanek Gail.** A Security Checklist for Web Application Design, SANS Institute. [Online] 2004. [Cited: 3 August 2016.] <https://www.sans.org/reading-room/whitepapers/securecode/security-checklist-web-application-design-1389>.
67. **Certified Secure.** Certified Secure Web Application Security Test Checklist. [Online] [Cited: 28 July 2016.] <https://www.certifiedsecure.com/checklists/cs-web-application-security-test.pdf>.
68. **OWASP.** Category: OWASP SWAAT Project. [Online] [Cited: 6 August 2016.] https://www.owasp.org/index.php/Category:OWASP_SWAAT_Project.
69. **Appvigil.** Automated Mobile App Vulnerability Detection. [Online] [Cited: 10 August 2016.] <https://appvigil.co/en/mrps/app-vulnerability-scanner/>.
70. *Open-source electronic data capture system offered increased accuracy and cost-effectiveness compared with paper methods in Africa.* **Dillon, David G., et al., et al.** 12, 2014, Journal of Clinical Epidemiology, Vol. 67, pp. 1358-1363.