SMART ATTENDANCE SYSTEM USING QR CODE

YONG WEI LUN

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Science (Hons.) Software Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

April 2019
DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : ____________________________________

Name : Yong Wei Lun

ID No. : 1605763

Date : 26 April 2019
I certify that this project report entitled **SMART ATTENDANCE SYSTEM USING QR CODE** was prepared by **YONG WEI LUN** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Hons.) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : ________________________________

Supervisor : Dr. Hoo Meei Hao

Date : ________________________________
The copyright of this report belongs to the author under the terms of the copyright Act 1987 as qualified by Intellectual Property Policy of Universiti Tunku Abdul Rahman. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2019, Yong Wei Lun. All right reserved.
ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Dr. Hoo Meei Hao for her invaluable advice, guidance and her enormous patience throughout the development of the research.

In addition, I would also like to express my gratitude to my loving parents and friends who had helped and given me encouragement.
ABSTRACT

Recording attendance is something repetitive and time consuming. The process of attendance taking is the same and repeat for every day. However, the attendance system today is not automated. It requires a lot of manual workforce to accomplish it. An automated attendance system can save human labours, and increase efficient of attendance taking. This will directly help lecturers to save time, and spend more time on academic, rather than attendance records. This project intended to automate the attendance recording. The adopted development methodology is evolutionary prototyping to cater for constantly user feedbacks and improvements. The project had conducted testing on 2 classes, for one lecture class and one practical class. The results were accurate and eliminated the needs of signing attendance on attendance sheets, the manual efforts to transfer data on attendance sheets to computer system. Some improvements have to be made before it is fully functional, for instance, the camera is not zoomable at the moment. This caused inconvenience on students who sit at behind or with poor smartphone camera quality. In conclusion, the project had achieved the objectives, which ultimately save lecturers’ time in managing attendance, bring convenience to students on attendance registration, and reduce the likelihood of fake attendance records.
TABLE OF CONTENTS

DECLARATION ii
APPROVAL FOR SUBMISSION iii
ACKNOWLEDGEMENTS v
ABSTRACT vi
TABLE OF CONTENTS vii
LIST OF TABLES xi
LIST OF FIGURES xii
LIST OF ABBREVIATIONS xv
LIST OF APPENDICES xvi

CHAPTER

1 INTRODUCTION 16
  1.1 Introduction 16
  1.2 Background 16
  1.3 Problem Statement 16
  1.4 Goal and Objectives 18
    1.4.1 Goal 18
    1.4.2 Objectives 18
  1.5 Proposed Solution 18
  1.6 Proposed Approach 20
  1.7 Project Scope 20
    1.7.1 Target Users 20
    1.7.2 Platform 20
    1.7.3 Modules 21
      1.7.3.1 Mobile Module 21
      1.7.3.2 Web Module 21
      1.7.3.3 Backend Service Module 21
    1.7.4 Modules Not Covered 21
## LITERATURE REVIEW

2.1 Introduction  
2.2 Review on Existing System  
    2.2.1 Related Approach  
2.3 Review on Attendance Tracking Technologies  
2.4 Technologies Review  
    2.4.1 Single Page Application  
    2.4.2 Public Cloud Services  
2.5 Development Methodology Review

## PROJECT METHODOLOGY AND PLANNING

3.1 Introduction  
3.2 Software Development Methodology  
    3.2.1 Evolutionary Prototype  
    3.2.2 Prototype Iteration  
3.3 Research Method  
    3.3.1 Respondent’s Regions  
    3.3.2 Summary of Survey  
3.4 Project Plan  
    3.4.1 Work Breakdown Structure  
    3.4.2 Gantt Chart  
3.5 Development Tools  
    3.5.1 Tools and IDEs  
    3.5.2 Services  
    3.5.3 Programming Languages  
    3.5.4 Frameworks  
    3.5.5 Database

## PROJECT SPECIFICATION

4.1 Introduction  
4.2 System Flow  
4.3 Use Case  
    4.3.1 Use Case Diagram
5  SYSTEM DESIGN
   5.1  Introduction  41
   5.2  Architecture Design  41
   5.3  Database Design  43
   5.4  Software Design  43
   5.4.1  Reactive  43
   5.4.2  Functional  44
   5.5  User Interface  44

6  PROJECT IMPLEMENTATION
   6.1  Introduction  53
   6.2  Routes Hierarchy  53
   6.3  Web API Endpoints  54
   6.4  Web Socket  56
   6.5  Code Generator  57
   6.6  Amazon Web Service  59
   6.6.1  Cognito  59
   6.6.2  Elastic Beanstalk  60
   6.6.3  API Gateway  62
   6.6.4  Simple Storage Service  63
   6.7  Firebase  63
   6.7.1  Firebase Hosting  63
   6.8  MongoDB Atlas  64
   6.9  Process Flow Diagram  65
   6.9.1  User Authentication  65
   6.9.2  API Calls  66
   6.9.3  QR Code Generation  66
# PROJECT TESTING

- **7.1 Introduction**
- **7.2 Unit Testing**
- **7.3 User Interface Testing**
- **7.4 Static Code Analysis**
- **7.5 Code Quality Review**
- **7.6 Application Audit**
- **7.7 User Acceptance Testing**

# CONCLUSION AND DISCUSSION

- **8.1 Introduction**
- **8.2 Limitation**
- **8.3 Improvement and Recommendation**
- **8.4 Challenges**
- **8.5 Conclusion**

# REFERENCES

# APPENDICES
# LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Comparison of Attendance Tracking Technologies</td>
<td>24</td>
</tr>
<tr>
<td>2.2</td>
<td>Comparison between SPA and Traditional Website</td>
<td>25</td>
</tr>
<tr>
<td>2.3</td>
<td>Comparison of Software Development Methodologies</td>
<td>27</td>
</tr>
<tr>
<td>3.1</td>
<td>Description of Phases for Prototype Model</td>
<td>30</td>
</tr>
<tr>
<td>4.1</td>
<td>Use case description of lecturer manage class</td>
<td>37</td>
</tr>
<tr>
<td>4.2</td>
<td>Use case description of lecturer manage attendance</td>
<td>38</td>
</tr>
<tr>
<td>4.3</td>
<td>Use case description of lecturer generate QR code</td>
<td>38</td>
</tr>
<tr>
<td>4.4</td>
<td>Use case description of student scan QR code</td>
<td>38</td>
</tr>
<tr>
<td>4.5</td>
<td>Use case description of student check attendance records</td>
<td>39</td>
</tr>
<tr>
<td>6.1</td>
<td>List of web API endpoints</td>
<td>54</td>
</tr>
<tr>
<td>7.1</td>
<td>UI test case</td>
<td>69</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The Proposed Solution Overview</td>
<td>19</td>
</tr>
<tr>
<td>1.2</td>
<td>Prototype Model</td>
<td>20</td>
</tr>
<tr>
<td>2.1</td>
<td>Attendance System at University Technology Petronas (UTP)</td>
<td>22</td>
</tr>
<tr>
<td>2.2</td>
<td>Public Cloud Adoption</td>
<td>26</td>
</tr>
<tr>
<td>3.1</td>
<td>Prototyping Model</td>
<td>29</td>
</tr>
<tr>
<td>3.2</td>
<td>Proposed Solution Architecture Design</td>
<td>35</td>
</tr>
<tr>
<td>4.1</td>
<td>IPO Diagram of Generating QR Code</td>
<td>36</td>
</tr>
<tr>
<td>4.2</td>
<td>IPO Diagram of Scanning QR Code</td>
<td>36</td>
</tr>
<tr>
<td>4.3</td>
<td>Use Case Diagram</td>
<td>37</td>
</tr>
<tr>
<td>5.1</td>
<td>System architecture design</td>
<td>42</td>
</tr>
<tr>
<td>5.2</td>
<td>MongoDB schema design</td>
<td>43</td>
</tr>
<tr>
<td>5.3</td>
<td>Login</td>
<td>44</td>
</tr>
<tr>
<td>5.4</td>
<td>Dashboard</td>
<td>44</td>
</tr>
<tr>
<td>5.5</td>
<td>Current class</td>
<td>45</td>
</tr>
<tr>
<td>5.6</td>
<td>QR code generation</td>
<td>45</td>
</tr>
<tr>
<td>5.7</td>
<td>Attendance list for current class</td>
<td>45</td>
</tr>
<tr>
<td>5.8</td>
<td>Timetable page to manage class slot</td>
<td>46</td>
</tr>
<tr>
<td>5.9</td>
<td>Add slot to a subject</td>
<td>46</td>
</tr>
<tr>
<td>5.10</td>
<td>Subject’s overview</td>
<td>46</td>
</tr>
<tr>
<td>5.11</td>
<td>Past attendance records based on week</td>
<td>47</td>
</tr>
<tr>
<td>5.12</td>
<td>Bar list</td>
<td>47</td>
</tr>
<tr>
<td>5.13</td>
<td>Login</td>
<td>48</td>
</tr>
<tr>
<td>5.14</td>
<td>Dashboard of a student</td>
<td>49</td>
</tr>
<tr>
<td>5.15</td>
<td>QR code scanning for taking attendance</td>
<td>50</td>
</tr>
<tr>
<td>5.16</td>
<td>Timetable</td>
<td>51</td>
</tr>
<tr>
<td>5.17</td>
<td>Attendance records</td>
<td>52</td>
</tr>
</tbody>
</table>
6.1 Routes hierarchy for lecturer view web app 53
6.2 Routes hierarchy for student view web app 54
6.3 Simple diagram of a Web Socket connection over client and server 57
6.4 Code generator design 58
6.5 Cognito configuration for lecturer view authentication 59
6.6 Cognito configuration for student view authentication 60
6.7 The activity of Elastic Beanstalk that serves for web API 60
6.8 Elastic Beanstalk instance configuration with load balancer and auto scaling capacity 61
6.9 The activity of Elastic Beanstalk that shows target response time of web socket connection 61
6.10 The activity of Elastic Beanstalk that serves for web socket connection 61
6.11 Elastic Load Balancing configuration 62
6.12 Dashboard for API Gateway that shows activities 63
6.13 The storage used in S3 63
6.14 Usage of the Firebase Hosting 64
6.15 Overview of MongoDB running on 3 nodes, which are primary node and 2 secondary nodes 64
6.16 Overview of activities on primary node 65
6.17 User authentication process flow 65
6.18 Web API calls process flow 66
6.19 QR code generation process flow 66
7.1 Test results on browser 67
7.2 Test results on console 68
7.3 Test results on browser 68
7.4 Test results on console 68
7.5 Unit test result for backend repository 69
7.6 Code review scoring for lecturer view repository 70
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.7</td>
<td>Code review scoring for student view repository</td>
<td>71</td>
</tr>
<tr>
<td>7.8</td>
<td>Code review scoring for backend repository</td>
<td>71</td>
</tr>
<tr>
<td>7.9</td>
<td>Lighthouse audit report for lecturer view web app</td>
<td>71</td>
</tr>
<tr>
<td>7.10</td>
<td>Lighthouse audit report for student view web app</td>
<td>72</td>
</tr>
<tr>
<td>7.11</td>
<td>Conducting UAT on a practical class</td>
<td>73</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------</td>
<td></td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
<td></td>
</tr>
<tr>
<td>IPO</td>
<td>Input Output Diagram</td>
<td></td>
</tr>
<tr>
<td>SPA</td>
<td>Single Page Application</td>
<td></td>
</tr>
<tr>
<td>AWS</td>
<td>Amazon Web Service</td>
<td></td>
</tr>
<tr>
<td>UAT</td>
<td>User Acceptance Testing</td>
<td></td>
</tr>
</tbody>
</table>
## LIST OF APPENDICES

<table>
<thead>
<tr>
<th>APPENDIX</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Survey Question</td>
<td>78</td>
</tr>
<tr>
<td>B</td>
<td>Results of Survey</td>
<td>79</td>
</tr>
<tr>
<td>C</td>
<td>Work Breakdown Structure</td>
<td>80</td>
</tr>
<tr>
<td>D</td>
<td>Gantt Chart</td>
<td>81</td>
</tr>
<tr>
<td>E</td>
<td>Lighthouse Audit</td>
<td>82</td>
</tr>
<tr>
<td>F</td>
<td>User Satisfaction Survey</td>
<td>83</td>
</tr>
<tr>
<td>G</td>
<td>Supervisor’s comments</td>
<td>84</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 Introduction
This chapter discusses about the background, problem statement, goal and objective, proposed solution, proposed approach and project scope.

1.2 Background
The popularity of smartphone has increases over the year. This could change and speed up the attendance taking process in universities. The current attendance system is time consuming and required manual workload. Lecturers will let the students to sign on attendance list. Then, he or she will key in to the university’s portal to record the attendance, each for every class. This will consume lecturers valuable time. Also, it will introduce human errors during the transferring from paper attendance to digitalised attendance records. Besides, students can easily cheat on attendance by asking flavours from their friends, to sign on the attendance sheet on their behalf. This is hard to avoid when the attendance sheets were given to students. The proposed solution is based on QR code to record students' attendance. The system will able to verify students' identity and prevent false registration. All of the attendance records will be recorded to the system and available to students and lecturers instantly. This will eventually reduce the human efforts on attendance registration.

1.3 Problem Statement
The process of taking students' attendance in higher education in time consuming and inefficient. The process become harder to manage when the class size is big. Prevention cheating on attendance is even harder to control. After each class, lecturers have to record the attendance to related web-based system according to the attendance sheets. The whole attendance taking process is consuming the valuable time of lecturers. With all the problem mentioned above, universities are not
changing the way they record attendance. Refer to Appendix A, 85% of the universities still recording attendance by signing on attendance sheets.

Mobile cellular penetration in Malaysia has reached 131.8% while smartphone penetration stood at 70% in the third quarter of last year (TheStar). With the widespread of smartphones among students in university, and the problem of wasting lecturers time in attendance taking process, proposes a solution that offer to simplify the attendance taking process. The proposed solution proposes QR code for students to scan with their smartphone. The attendance will be confirmed with the identity from the smartphone. This will save time and effort to record attendance at the same time, reduce unwanted paper usage.

The proposed solution identifies unauthorised attendance registration using multi-factor authentication. The details of authentication factors will be explained in following chapters.

1.4 Goal and Objectives
This section discusses about goal and objectives.

1.4.1 Goal
Reduce administrative and lecturers labour manually and minimise paper usage.

1.4.2 Objectives
a. To simplified current attendance taking process by automating the process using QR Code scanning.
b. Increase the transparency of attendance records by enabling students and lecturers to view attendance records anytime.
c. Reduce cheating on attendance by enforcing cheat detection.
d. Minimise paper usage by eliminating the needs of attendance sheets.

1.5 Proposed Solution
The proposed solution is by taking attendance using QR Code. 37.5% of respondents think that QR Code solution is the most cost effective.
The reason of using QR Code is because it provides low initial cost for implementing the system. By using the tools that everyone student has, smartphone to take attendance.

The solution has 3 modules, mobile module, web module and backend service module. Web module displays QR code that generated by backend service module. The QR code updates every 10 seconds to prevent cheating. This can prevent students who are not in class and wanted to scan the QR code. The mobile module enables students to scan QR code to confirm their attendance. The request is then sent to backend service module for verification. Once the attendance is verified, the backend service module will update database on attendance records. The whole process of attendance taking should be less than 5 minutes, which is much faster compared with traditional process.

![Diagram](image)

Figure 1.1: The Proposed Solution Overview

Students can view their attendance records for enrolled subjects, this can increase the transparency of attendance system. The system also issue notification if the attendance percentage is below than certain level. Moreover, lecturers also can manage students records with web module. Lecturers can update students’ attendance records in case they have valid reason to absent class. Lecturers can manage class as well. New class can be created in case there is any class replacement. Existing classes can be modified or deleted by lecturers.

Administrators can manage users with web module. They can add new lecturers or students, update their details and delete existing users.
1.6 Proposed Approach

The proposed methodology is Evolutionary Prototyping Methodology.

![Prototype Model Diagram]

Prototype model involves users in the development stage. This can have clearer understanding of requirements based on users' feedback. Errors can be detected in the early stage; missing module can be identified easily. An attendance system should be reliable and easy to use. Thus, it is good to involve users in the development stage.

1.7 Project Scope

The section below explains the target users, platform and modules of the project.

1.7.1 Target Users

i. UTAR students
   - To record their attendance for every class.

ii. UTAR Lecturers
   - To show the QR Code to students during class.
   - View and manage student's attendance records.

iii. UTAR administrator staffs
   - Manage timetable, classes and users (students and lecturers).

1.7.2 Platform

The system consists of 2 platforms, web and mobile platform. Web is for lecturers. Students can be accessing from mobile platform.
1.7.3 Modules
The section below explains modules in mobile, web and backend service.

1.7.3.1 Mobile Module
i. Record attendance by scanning QR Code.
ii. View attendance records.
iii. Issue notification if students’ attendance lower than certain limit.

1.7.3.2 Web Module
i. View, add, modify classes.
ii. View, modify attendance records.
iii. View overall attendance records in charts.
iv. Display QR Code for each class.

1.7.3.3 Backend service Module
i. Prevent unauthorised registration of attendance with multi-factor authentication.
ii. Provides login authentication on both mobile and web.
iii. Generate code for QR code.
iv. Run data analytics operation to generate charts for attendance records.

1.7.4 Modules that are not covered
a. Tutorials on how-to-use on both mobile and web modules.
b. Web module for administrator to manage user pools and timetable data.
CHAPTER 2

LITERATURE REVIEW

2.1 Introduction
This chapter reviews on existing system, attendance tracking technologies, technologies and development methodologies.

2.2 Review on Existing System
According to appendix A, most of the universities and colleges still record attendance on attendance sheets. Attendance system is clearly missing out the evolution.

According to Hsarena (2017), University Malaysia Sarawak (UNIMAS) was the first in Malaysia to use QR Code to record attendance in October 2017 followed by University Technology Petronas (UTP) in January 2018.

Figure 2.1: Attendance System at University Technology Petronas (UTP)
Similar system has implemented on University Technology Petronas and Multimedia University as well. Lecturers will sometimes give a pop quiz to prevent cheating. No further prevention is done.

2.2.1 Related Approach
Masalha (2014) proposes that using QR code to record attendance with additional security to prevent unauthorized attendance registration using multi-factor authentication, which are username and password with facial identification check. To ensure that the students is present in the class while scanning the QR code, the location will be check as well.

Saraswat (2010) proposes that using fingerprint scanner to records attendance. Fingerprint is something we all have uniquely and accessible all the time. By using fingerprint scanner, it will prevent cheating on attendance and provide convenient yet secure way to record attendance.

Balcoh, et al. (2012) proposes an algorithm for efficient attendance management. With face recognition, the effort of recording attendance is reduced to minimum. The process is done behind the scene. The algorithm able to identify multiple faces from a single image. This will ultimately prevent cheating on attendance.

We noticed that proposal of automated attendance systems is not something new. Different regions have different preferences on attendance taking system. Ultimately, the system will reduce labour work and prevent attendance cheating.

2.3 Review on Attendance Tracking Technologies
The table below shows some of the examples of attendance tracking technologies (The Hans India, 2018). Some of the low-cost setup are using polling, geolocation, and QR code.

Polling required students to enter the answer that asked by lecturers during the class or enter a special unique code. This however is easy to cheat because students can send out the code or answer to their friends who are not in the class. Next, tracking attendance by using geolocation. When students enter into class, the system will detect and record attendance. However, GPS often suffer with accuracy in indoor condition. This will make the attendance not accurate. Thirdly, QR code. Students required to scan the QR code that displayed during class to confirm
attendance. This method also can be cheat easily, by sending the QR code to friends who are not in the class. We could enhance the security by adding few mechanisms. QR code should be refreshed at fast as 10 seconds, so that when the QR code is sent to students who are not in the class, the code will be invalid. The system also needs to allow one device to login at student account at a time. This will prevent cheating on attendance as well.

Table 2.1: Comparison of Attendance Tracking Technologies

<table>
<thead>
<tr>
<th>Tracking technologies</th>
<th>Description</th>
<th>Required equipment</th>
<th>Easy to fake attendance</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polling</td>
<td>Required students to answer a question or enter code to devices</td>
<td>Laptop or smartphone</td>
<td>Easy</td>
<td>low</td>
</tr>
<tr>
<td>Geolocation</td>
<td>Automatically records attendance when the location of students matches with a particular class</td>
<td>Smartphone</td>
<td>Easy</td>
<td>Low</td>
</tr>
<tr>
<td>Card Reader</td>
<td>Required students to scan their student ID on card readers</td>
<td>Student ID, card reader</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Fingerprint Reader</td>
<td>Required students to scan their fingerprint on fingerprint reader to register attendance</td>
<td>Fingerprint reader</td>
<td>Hard</td>
<td>High</td>
</tr>
<tr>
<td>QR Code</td>
<td>Required students to scan QR code that displayed in the class</td>
<td>Smartphone</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Bluetooth</td>
<td>Automatically register attendance when within the Bluetooth coverage area</td>
<td>Smartphone</td>
<td>Medium</td>
<td>Medium</td>
</tr>
</tbody>
</table>
2.4 Technologies Review
This section discusses about technologies that are going to be used.

2.4.1 Single Page Application
Mikowski and Pollen (2013) said that Single Page Application (SPA) has been around for long time. SPA is an application that run on the browser and does not reload the page when using. SPA can render like a desktop application as it redraws the part of the user interface that needs to be update when needed. Unlike traditional website, it redraws on every request as well as user action. The website fully depending on web server. In the case of the server is overloaded or busy, or the internet connection is unable, the page will not response to user action as the user might think that the page is not responding. A page refresh will cause the request to be resent. This deliver horrible experience to the user.

SPA can show its state just like native application. When a SPA is loading, it can render progress bar or indicator dynamically. This will create better user experience, user does not have to keep on refresh the page to check its state, compared to traditional website.

SPA is universal and cross-platform. It works with any platform as long as have access to modern web browsers. It also can be update instantly and distribute.

Moreover, SPA can easily port to Progressive Web App, mobile application and Desktop application with the aids of current technologies.

Table 2.2: Comparison between SPA and Traditional Website

<table>
<thead>
<tr>
<th></th>
<th>Single Page Application</th>
<th>Traditional website</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reload</strong></td>
<td>Does not reload on user action</td>
<td>Reload on every user action</td>
</tr>
<tr>
<td><strong>Response</strong></td>
<td>Nearly immediate response</td>
<td>Solely relay on servers and internet connection</td>
</tr>
<tr>
<td><strong>Show its state</strong></td>
<td>Able to show notify users by showing progress bar or indicator dynamically</td>
<td>Not able to notify users of its state</td>
</tr>
</tbody>
</table>
2.4.2 Public Cloud Services

![Enterprise Public Cloud Adoption](image)

Figure 2.2 Public Cloud Adoption

**Source taken from (Cynthia & Andy, 2018)**

According to Cynthia and Andy (2019), Amazon Web Service (AWS) is dominating in public cloud market shares. AWS has been in the market share leader in cloud Iaas for over 10 years. Although Microsoft Azure and Google Cloud Platform (GCP) are catching up in recent years. AWS offers the most services compared with Azure and GCP. Therefore, in the project AWs is chosen for the cloud services.

2.5 Development Methodology Review

In software project, adopting a rigid or flexible approach can influence on the project outcome. With the face pace in software industry, a suitable methodology will definitely help. Code reduction, reusing and security should be neglect too.

Clear requirements mean the software requirements are clear. Clear solutions mean the proposed solutions are clear. Feedback on work done means that the feedback will be given during development. Frequent enhancements are the enhancements that need to be made after feedbacks. Cost of delay means that the cost when the project is delayed. Experience on projects is based on the software engineer’s experience on similar solution.
Waterfall methodology is suitable when the requirements and solutions are clear, less feedback and enhancement during development, low cost of delay and software engineer have experience on similar project.

Kanban methodology is suitable when the requirements are clear, solution are unclear, required constant feedback but few changes during development, low cost of delay and software engineer may have experience on similar project.

Scrum methodology is suitable when the requirements are clear, solutions are unclear, required constant feedback but few changes during development, high cost of delay and software engineer may have experience on similar project. Similar with Kanban methodology, just the difference in cost of delay.

Extreme programming is suitable when the requirements and solutions are unclear, required constant feedback and frequent enhancements, high cost of delay, and software engineer has less to no experience.

Evolutionary prototyping methodology is suitable when requirements and solutions are clear, required constant feedback and frequent enhancements, low cost of delay and software engineer may have experience on similar projects.

Table 2.3: Comparison of Software Development Methodologies

<table>
<thead>
<tr>
<th>Methodologies</th>
<th>Waterfall</th>
<th>Kanban</th>
<th>Scrum</th>
<th>Extreme Programming</th>
<th>Evolutionary Prototyping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear requirements</td>
<td>Clear</td>
<td>Clear</td>
<td>Clear</td>
<td>Unclear</td>
<td>Clear</td>
</tr>
<tr>
<td>Clear Solutions</td>
<td>Clear</td>
<td>Unclear</td>
<td>Unclear</td>
<td>Unclear</td>
<td>Clear</td>
</tr>
<tr>
<td>Feedback on work done</td>
<td>Minimum</td>
<td>Constant feedback</td>
<td>Constant feedback</td>
<td>Constant feedback</td>
<td>Constant feedback</td>
</tr>
<tr>
<td>Frequent enhancements</td>
<td>No changes</td>
<td>Few changes</td>
<td>Frequent changes</td>
<td>Frequent changes</td>
<td>Frequent changes</td>
</tr>
<tr>
<td>Cost of delay</td>
<td>Minimum</td>
<td>Minimum</td>
<td>Huge</td>
<td>Huge</td>
<td>Minimum</td>
</tr>
<tr>
<td>Experience on projects</td>
<td>Have experience</td>
<td>May have experience</td>
<td>May have experience</td>
<td>Less to no experience</td>
<td>May have experience</td>
</tr>
</tbody>
</table>
The proposed solution has clear requirements and solution, need constant feedback and frequent changes and low cost of delay. Thus, Evolutionary Prototype methodology will be the best fit.
CHAPTER 3

PROJECT METHODOLOGY AND PLANNING

3.1 Introduction
This chapter explains about software development methodology and research method that are chosen and project planning.

3.2 Software Development Methodology
This section discusses about development methodology.

3.2.1 Evolutionary Prototype

Figure 3.1: Prototyping Model
Evolutionary prototyping involves the building of prototype software which incrementally evolves into the final system (Carter, 2001). It has 5 phases of development.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gather requirements</td>
<td>Gather all the requirements. The requirements were gathered from the lecturer and produce requirement specifications.</td>
</tr>
<tr>
<td>Build prototype</td>
<td>Build the prototype accordingly. The prototype was built incrementally based on user feedbacks.</td>
</tr>
<tr>
<td>Evaluate prototype</td>
<td>Evaluate prototype accordingly. The prototype was evaluated by users, and feedbacks have been collected for further evaluations.</td>
</tr>
<tr>
<td>Refine prototype</td>
<td>Refine prototype based on feedback. The prototype was refined based on the feedbacks and prepared for the next iteration of prototyping.</td>
</tr>
<tr>
<td>Deliver</td>
<td>Deliver the system to end user</td>
</tr>
</tbody>
</table>

3.2.2 Prototype Iteration
For the solution have 2 iterations of prototype. During each iteration, evaluation had conducted and refine at next iteration.

3.3 Research Method
This section discusses the research method for the project including respondents’ selection and summary of the research. The research is conducted through online survey.
3.3.1 Respondents’ Regions

There are total 103 higher education centre in Malaysia, 20 public universities, 43 private universities, 31 university colleges and 9 foreign university branches (StudyMalaysia). A survey was conducted with 20 universities as list below with one representative from each university.

Overseas

i. Loma Linda University
ii. London School of Economics
iii. Newcastle University
iv. Nanyang Technological University

Local

i. University Science Malaysia
ii. Health Campus University Science Malaysia
iii. Multimedia University
iv. Sunway College University
v. The One Academy
vi. University Malaysia Terengganu
vii. University Utara Malaysia
viii. UCSI University
ix. Nilai University
x. University of Malaya
xi. National University of Malaysia
xii. Infrastructure University Kuala Lumpur
xiii. Tunku Abdul Rahman University College
xiv. Asia Pacific University of Technology and Innovation
xv. HELP University
xvi. YPC University

3.3.2 Summary of Survey

Based on Appendix B, 90% of the respondents’ universities still recording attendance by signing on attendance sheet, 5% using QR code attendance system and
5% using Barcode scanner on Student ID. This shows that universities are not aware of the attendance systems are outdated, with only a few exceptions.

75% of the respondents have no issue with current attendance system and 75% of the respondents think that cheating on attendance are easy. This shows that current attendance systems don’t have much impact on them, from students’ perspective. However, automated attendance systems will save lecturers’ time and reduce human error when recording attendance to particular web system. 75% of respondents agreed with that and 60% of respondents think attendance system should be automated. Automation needs to be done in order to save lecturers’ time from repetitive tasks.

40% of the respondents think that using biometrics scanner is the best to prevent cheating on attendance. While 20% of them think using geolocation and Student ID reader and 15% of them think using QR code. However, while biometrics scanner might be the most secure to prevent cheating on attendance, it might have some drawbacks such as high cost. 35% of the respondents think that QR code is the most cost-effective attendance tracking technology, while 25% of them think polling, 20% of them think Student ID reader and 10% of them think geolocation and biometrics scanner.

3.4 Project Plan
This section discusses about Work Breakdown Structure and Gantt chart.

3.4.1 Work Breakdown Structure
Work Breakdown structure is attached as appendix C.

3.4.2 Gantt Chart
Gantt chart is attached as appendix D.

3.5 Development Tools
This section discusses about tools and integrated development environment (IDE), services, programming languages, frameworks and database.

3.5.1 Tools and IDEs
i. Git
Git is be used as version control. It is widely used by software development. It is a distributed version control system. It is used to track the software version to ensure that the current changes would not break the software. If it does, we can roll back to the previous version.

ii. MongoDB Compass
MongoDB Compass is a visualisation tool for MongoDB. It eased the development with MongoDB with the handy visualisation feature.

iii. JetBrains IDEs
JetBrains IDEs that using are be Webstorm and GoLand. Webstorm is popular with frontend web development and GoLand in designed for Go programming language. These IDEs were chosen in order to maximise productivity due to familiarity and the features offered in each IDEs.

3.5.2 Services
i. Amazon Web Service (AWS)
Amazon Web Service is a public cloud service that offer variety of services. It supports worldwide with different regions. AWS is used for cloud-native architecture design.

i) Amazon Elastic Beanstalk
Amazon Elastic Beanstalk is a Platform-as-a-Service that provides managed environment for servers to execute. This service is suitable for all kind of server deployments

ii) Amazon Cognito
Amazon Cognito is service that provides user authentication and access on clients. This eased the processing of development user authentication.

iii) Amazon API Gateway
Amazon API Gateway is service for API deployment. It well integrated with AWS services and provide secure APIs at any scale. This service provides a HTTPS connection.

iv) Amazon S3
Amazon S3 is a managed cloud object storage service. It can used to store variety of files, as well as web hosting. It is highly available and accessible anytime.

ii. MongoDB Atlas
MongoDB Atlas is a database as-a-service offered by MongoDB. It runs on 3 public clouds, Amazon Web Service, Google Cloud Platform and Microsoft Azure. By using this service, we do not need to manage our database instances.

iii. Firebase
Firebase is a mobile and web development platform. It offers variety of services that suit for mobile and web development
   i) Firebase Hosting
      Firebase hosting is a static web hosting service that provides HTTPS. It is used for secure web hosting due to only secure connection can access to mobile’s camera.

iv. GitHub
GitHub is a development platform that offer web hosting for version control system. It supports Git out of the box. GitHub is widely used by commercial and open source software development.

3.5.3 Programming Languages
i. TypeScript
TypeScript is superset of JavaScript that developed and maintained by Microsoft. It adds optional types and transpiles to high-performance JavaScript which is suitable for large-based application.

ii. Go
Go is a programming language that developed and maintained by Google. It makes building simple, reliable and effective software easy. It is good in performance and multi-threading task with light weight thread called Goroutine.

3.5.4 Frameworks
i. Angular
Angular is an open source framework for front-end web application and based on TypeScript which is designed and maintained by Angular Team at Google.

ii. Ionic

Ionic is an open source framework for building hybrid mobile application and progressive web app. Ionic is heavily influenced by Angular, and the knowledge can be transferred from Angular to Ionic.

### 3.5.5 Database

i. MongoDB

MongoDB is an open source document-stored NoSQL database. It is well integrated with popular programming languages. It is used due to the performance and flexibility of the database structure.

![Proposed Solution Architecture Design](image)

*Figure 3.2: Proposed Solution Architecture Design*
CHAPTER 4

PROJECT SPECIFICATION

4.1 Introduction
This chapter discusses about system flow, use case and project requirements.

4.2 System Flow

<table>
<thead>
<tr>
<th>Input</th>
<th>Process</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click on &quot;Generate QR code&quot; button</td>
<td>Generate QR code</td>
<td>Display QR code</td>
</tr>
<tr>
<td>Lecturer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.1: IPO Diagram of Generating QR Code

<table>
<thead>
<tr>
<th>Input</th>
<th>Process</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan QR code</td>
<td>Validate QR code</td>
<td>Display success/failure message</td>
</tr>
<tr>
<td>Student</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.2: IPO Diagram of Scanning QR Code
4.3 Use Case
This section discusses about use cases.

4.3.1 Use Case Diagram

![Use Case Diagram](image)

Figure 4.3: Use Case Diagram

4.3.2 Use Case Description

Table 4.1: Use case description of lecturer manage class

<table>
<thead>
<tr>
<th>Use case name</th>
<th>Manage class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Lecturer</td>
</tr>
</tbody>
</table>

Flow of Events

1. **Create class**
   i. Lecturer create class.
   ii. Lecturer fill in class name, class type and duration.
   iii. System sends the request to server and saves to database.

2. **View class**
   i. Lecturer selects class.
   ii. System displays class name, class type, duration.

3. **Update class**
   i. Lecturer selects class.
   ii. Lecturer update the field that desired to change.
   iii. System sends the updated records to server and saves to database.
4. **Delete class**
   i. Lecturer selects class.
   ii. Lecturer deletes class.
   iii. System sends the request to server and deletes from database.

Table 4.2: Use case description of lecturer manage attendance

<table>
<thead>
<tr>
<th>Use case name</th>
<th>Manage attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actor</strong></td>
<td>Lecturers</td>
</tr>
</tbody>
</table>

**Flow of Events**

1. **View attendance**
   i. Lecturer selects student from a class.
   ii. System displays the attendance records for the student of the class.

2. **Update attendance**
   i. Lecturer selects student from a class.
   ii. Lecturer update the attendance status.
   iii. System sends the updated records to server and saves to database.

Table 4.3: Use case description of lecturer generate QR code

<table>
<thead>
<tr>
<th>Use case name</th>
<th>Generate QR code</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actor</strong></td>
<td>Lecturers</td>
</tr>
</tbody>
</table>

**Flow of Events**

i. Lecturer clicks on the “Generate QR code” button to generate QR code.
ii. System sends request to server and retrieve code for the class.
iii. System displays the QR code.

Table 4.4: Use case description of student scan QR code

<table>
<thead>
<tr>
<th>Use case name</th>
<th>Scan QR code</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actor</strong></td>
<td>Students</td>
</tr>
</tbody>
</table>

**Flow of Events**

i. Student clicks on the “Scan QR code” button to scan QR code
ii. System sends the request to server for verification
iii. System displays success or failure message
Table 4.5: Use case description of student check attendance records

<table>
<thead>
<tr>
<th>Use case name</th>
<th>Check attendance records</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Students</td>
</tr>
</tbody>
</table>

**Flow of Events**

1. **Check for attendance status for a class**
   i. Student clicks on a class.
   ii. System displays the attendance status of the class.

2. **Check for overall attendance records for a subject**
   i. System shows the percentage of the attendance records for a subject on home page.

**4.4 Software Requirements**

This section discusses about functional and non-functional requirements.

**4.4.1 Functional Requirements**

**Lecturers**

1) The system shall be able to generate QR code
2) The system shall be able to manage attendance records
   i. The system shall be able to view attendance records
   ii. The system shall be able to update attendance records
   iii. The system shall be able to view bar list
3) The system shall be able to manage classes
   i. The system shall be able to create classes
   ii. The system shall be able to view classes
   iii. The system shall be able to update classes
   iv. The system shall be able to delete classes

**Students**

1) The system shall be able to scan QR code to record attendance
2) The system shall be able to check attendance records
   i. The system shall be able to check attendance records for all classes
   ii. The system shall be able to view students’ attendance percentage
   iii. The system shall be able to view class details with time and venue
4.4.2 Non-Functional Requirements

i. The system shall prevent cheating attendance by verify the request.

ii. The system shall prevent unauthorised login on users' account.

iii. The system shall be error-tolerant.

iv. The system shall be fast and responsive.
CHAPTER 5

SYSTEM DESIGN

5.1 Introduction
This chapter discusses software modelling and design, database design and process modelling.

5.2 Architecture Design
The system architecture design is practicing cloud-native architecture and microservices. Cloud-native architecture is the design practice that is specially designed to run on cloud, to take the advantages of cloud computing. It is taking account of the characteristic and nature of the cloud services, and resulting the application fully take advantage of the cloud. This enabled the application can be scale horizontally effectively when needed. Also, practicing cloud-native architecture, we do not have to manage the infrastructure ourselves most of the time. The time can be reduced for setting up servers on-premise.

Besides, the architecture design also has microservices in mind. Microservices is a design practice that split the application into multiple pieces, which are self-container and standalone services. Microservices increases the scalability and maintainability of the application. Compared with Monolith application, microservices required extra steps to set up and get connected to each service. However, microservices is the preferred architecture style in modern design.

By combining the two, cloud-native and microservices architecture design, the application can fully take the advantages of the cloud services. Several cloud services are leverage due to the application are splitting into separate services. Due to the nature of the cloud, each service can scale horizontally and independently effectively when needed. This offers better pricing compared with on-premise servers’ setup. With cloud, scale up and down can be done easily, create and terminate instance at any time.
Figure 5.1: System architecture design

The diagram above shown the overall system architecture design for the application. Most of the services run on Amazon Web Service some of them on Google Firebase. We have two instances of Elastic Beanstalk. One for the REST API web service, the other for code generating service. These services have different workloads during throughout the day, it is better to split them into separate services. All the web services can be accessed from a same gateway, which is the API Gateway. The API Gateway mapped to the related services based on the requests and response accordingly. It as a unified gateway for all the web services.

The database also has been split into different service. This is to ensure that, when one of the web services is faulty, it would not affect other services. Database is important for data persistence; it is good to separate it from other services.

Moreover, the user authentication service, Cognito has been separated from the web services that we mentioned earlier. User authentication workload can be unpredictable. Users may have logged in to the system, but does not use it. By separate the authentication service with other services to ensure that it does not break other services when there is a lot of users logging in. Also, only authenticated users can access the other web services. It also acts as a secure door for all the services.
Lastly, the static web hosting. Web hosting also has been separated as a standalone service. This is to ensure that anything goes wrong during the web hosting process would not break other services.

Cloud-native architecture and microservices can be applied on any types of application, to improve the scalability and maintainability.

5.3 Database Design

Figure 5.2: MongoDB scheme design

5.4 Software Design

The application is designed with service-oriented, reactive and functional.

5.4.1 Reactive

Reactive programming is an asynchronous programming paradigm that concerned with propagation change and data stream. RxJS (Reactive Extension for JavaScript) is a library for reactive programming. It uses observables to make it easier to compose asynchronous, or callback-based code. RxJS is heavily used in the frontend application in Angular and Ionic.
5.4.2 Functional

Functional programming is a programming paradigm that treats computation as the evaluation of pure mathematics functions, and at the same time, avoids mutable data and changing-state. Functional programming promotes clean code and referential transparency. Functional programming principle are heavily used in both TypeScript and Go programming languages.

5.5 User Interface

Two web application were developed, one is for lecturer, the other is for student.

Lecturer view web application

![Login](image1)

Figure 5.3: Login

![Dashboard](image2)

Figure 5.4: Dashboard
Figure 5.5: Current class

Figure 5.6: QR code generation

Figure 5.7: Attendance list for current class
Figure 5.8: Timetable page to manage class slot

Figure 5.9: Add slot to a subject

Figure 5.10: Subject’s overview
Figure 5.11: Past attendance records based on week

Figure 5.12: Bar list
**Student view web application**

![Login](image)

*Figure 5.13: Login*

*Note that only Android phones with Google Chrome can access this camera.*
Figure 5.14: Dashboard of a student
Figure 5.15: QR code scanning for taking attendance
Figure 5.16: Timetable
Figure 5.17: Attendance records

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>3/25/16</td>
<td>11:00 AM</td>
<td>Present</td>
</tr>
<tr>
<td>3/25/16</td>
<td>08:22 PM</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER 6

PROJECT IMPLEMENTATION

6.1 Introduction

This chapter discusses project implementation on routes hierarchy, web API endpoints, web socket, code generator, Amazon Web Service, Firebase, MongoDB Atlas, and process diagram.

6.2 Routes Hierarchy

For both web app has different routes to reach the certain page. The routes that marked with lazy are lazy-loaded. In another words, a lazy-loaded page would only be loaded when the user route to it. This increases the initial load time as not all pages are downloaded.

Figure 6.1: Routes hierarchy for lecturer view web app
Figure 6.2: Routes hierarchy for student view web app

6.3 Web API Endpoints

The total count of API endpoints are 24 endpoints. Each endpoint serves different purposes.

Table 6.1: List of web API endpoints
### Type: Student, Method: GET

<table>
<thead>
<tr>
<th>Route</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/students</td>
<td>Get all students</td>
</tr>
<tr>
<td>/students/{id}</td>
<td>Get a student with ID</td>
</tr>
<tr>
<td>/students/{id}/subjects</td>
<td>Get all the subjects of the student</td>
</tr>
<tr>
<td>/students/{id}/today</td>
<td>Get the student’s today classes</td>
</tr>
<tr>
<td>/students/{id}/current</td>
<td>Get the student’s current class</td>
</tr>
<tr>
<td>/students/{id}/subjects/week/{weekno}</td>
<td>Get the student’s subjects by week number</td>
</tr>
</tbody>
</table>

### Type: Lecturer, Method: GET

<table>
<thead>
<tr>
<th>Route</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/lecturers</td>
<td>Get all lecturers</td>
</tr>
<tr>
<td>/lecturers/{id}</td>
<td>Get a lecturer with ID</td>
</tr>
<tr>
<td>/lecturers/{id}/subjects</td>
<td>Get all the subjects of the lecturer</td>
</tr>
<tr>
<td>/lecturers/{id}/today</td>
<td>Get the lecturer’s today classes</td>
</tr>
<tr>
<td>/lecturers/{id}/current</td>
<td>Get the lecturer’s current class</td>
</tr>
<tr>
<td>/lecturers/{id}/subjects/week/{weekno}</td>
<td>Get the lecturer’s subjects by week number</td>
</tr>
<tr>
<td>/lecturers/{id}/timeslot?slotDate={value} &amp;endDate={value}</td>
<td>Get the lecturer’s timeslot between two dates</td>
</tr>
</tbody>
</table>

### Type: Subjects, Method: GET, POST, PUT, DELETE

<table>
<thead>
<tr>
<th>Route</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/subjects</td>
<td>Get all subjects</td>
</tr>
<tr>
<td>/subjects/{code}</td>
<td>Get a specific subject with subject code</td>
</tr>
<tr>
<td>/subjects/{code}/weeks</td>
<td>Get subject info like start and end date, total week, and current week</td>
</tr>
<tr>
<td>/subjects/{code}/list?startTime={value} &amp;day={value}&amp;type={value}&amp;date={value}</td>
<td>Get the student list of the subjects between two dates</td>
</tr>
<tr>
<td>/subjects/{code}/overview</td>
<td>Get the overview student list of the subjects which includes students’</td>
</tr>
</tbody>
</table>
Web Socket
Web Socket is computer communication protocol, which provides full-duplex connection channel over a TCP connection. Web Socket often used for maintaining stateful connection between client and server. One the connection between both parties are established, the server can send data to the client and vice versa. This eliminates the needs of unnecessary of calling the web service to get updated data. Web Socket connection is used for the application for sending random code from the server and render as a QR code on the client. Due to the code is generated from the server in short interval, it is better to keep a stateful connection between client and server. When the client issues a close request, the server closes the Web Socket connection.

<table>
<thead>
<tr>
<th>Route</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/subjects/{code}/barlist</td>
<td>Get the student list who get barred or attendance percentage less than 80% so far</td>
</tr>
<tr>
<td>POST /subjects/{code}/slot</td>
<td>Add a new timeslot for the subject</td>
</tr>
<tr>
<td>PUT /subjects/{code}/slot</td>
<td>Update a timeslot for the subject</td>
</tr>
<tr>
<td>DELETE /subjects/{code}/slot</td>
<td>Delete a timeslot for the subject</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Route</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/attendance</td>
<td>Get all attendance records</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Route</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/registration?mode=auto</td>
<td>Automatic attendance registration by scanning QR code</td>
</tr>
<tr>
<td>/registration?mode=manual</td>
<td>Manual attendance alter/registration</td>
</tr>
</tbody>
</table>

6.4 Web Socket
Web Socket is computer communication protocol, which provides full-duplex connection channel over a TCP connection. Web Socket often used for maintaining stateful connection between client and server. One the connection between both parties are established, the server can send data to the client and vice versa. This eliminates the needs of unnecessary of calling the web service to get updated data. Web Socket connection is used for the application for sending random code from the server and render as a QR code on the client. Due to the code is generated from the server in short interval, it is better to keep a stateful connection between client and server. When the client issues a close request, the server closes the Web Socket connection.
6.5 Code Generator

Code generator server is used for random and unique code generation and send to client to displayed as QR code.

Figure 6.3: Simple diagram of a Web Socket connection over client and server
Scenario

1. In idle state, only one Goroutine, Manager is on ready state.
2. Client issues an open request to generate QR code
3. The open request is sent to Manager
4. Manager spawns additional 3 Goroutines concurrently, which are Socket Reader, Socket Writer and Code Generator. Each Goroutine is in ready state. Socket Writer waiting input from Code Generator meanwhile Socket Reader waiting input from the client.
5. Code Generator keeps generate code based on the time interval set. The generated code will store in a Channel.
6. Socket Writer reads the input from Channel, and send the generated code to client, and at the same time, it spawns an additional self-closing Goroutine, which is Database Manager, to save the generated code to database. When this operation is completed, the Data Manager will close itself.
7. The steps 5 and 6 keeps on repeating based on the time interval set from client.

8. When the client issues a close request, the Socket Reader will receive the request and inform Manager.

9. Once the Manager has received the close request, it closes all Goroutines, which are Socket Writer, Socket Reader and Code Generator.

10. It backs to only one Goroutine left, which is the Manager, same as the idle state.

6.6 Amazon Web Service

Four services are used on Amazon Web Service. These services serve different purposes.

6.6.1 Cognito

Cognito is used for user authentication on both lecturer and student. Both lecturer and student are in a separate user pool on Cognito.

Currently have 3 active users who are holding lecturer role.

<table>
<thead>
<tr>
<th>Estimated number of users</th>
<th>3</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Required attributes</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias attributes</td>
<td>none</td>
</tr>
<tr>
<td>Username attributes</td>
<td>none</td>
</tr>
<tr>
<td>Custom attributes</td>
<td>Choose custom attributes...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Minimum password length</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password policy</td>
<td>no requirements</td>
</tr>
<tr>
<td>User sign ups allowed?</td>
<td>Only administrators can create users</td>
</tr>
</tbody>
</table>

Figure 6.5: Cognito configuration for lecturer view authentication
Currently have 83 active users who are holding student role.

**Estimated number of users** 83

- **Required attributes** none
- **Alias attributes** none
- **Username attributes** none
- **Custom attributes** Choose custom attributes...

**Minimum password length** 7

- **Password policy** no requirements
- **User sign ups allowed?** Only administrators can create users

Figure 6.6: Cognito configuration for student view authentication

### 6.6.2 Elastic Beanstalk

Two Elastic Beanstalk instance are running, one for the web API, the other for the code generator. The code generator requires TCP connection; therefore, it needs an application load balancer. The load balancer opens up a TCP connection and enables auto scaling when the network is max out.

Web API

Figure 6.7: The activity of Elastic Beanstalk that serve for web API
Web Socket

Figure 6.8: Elastic Beanstalk instance configuration with load balancer and auto scaling capacity

Figure 6.9: The activity of Elastic Beanstalk that show target response time of web socket connection

Figure 6.10: The activity of Elastic Beanstalk that serve for web socket connection
6.6.3 API Gateway

API Gateway provides secure connection to clients and acts as a centralised gateway for all the web services.
6.6.4 Simple Storage Service

Simple Storage Service (S3) provides reliable static web hosting.

6.7 Firebase

Firebase provides various of services, but only Hosting service is used.

6.7.1 Firebase Hosting

Firebase Hosting provides static web hosting over HTTPS connection. Secure connection is a need for mobile device to access camera through web browser.
6.8 MongoDB Atlas

MongoDB Atlas provides Database-as-a-Service, we can use MongoDB without setting up a database server, and run on public clouds.

Figure 6.14: Usage of the Firebase Hosting

Figure 6.15: Overview of MongoDB running on 3 nodes, which are 1 primary node and 2 secondary notes.
Figure 6.16: Overview of activities on primary node

6.9 Process Flow Diagram

Process flow diagram shows the flow of services involved during a process.

6.9.1 User authentication

Figure 6.17: User authentication process flow
6.9.2 API Calls

Figure 6.18: Web API calls process flow

6.9.3 QR Code Generation

Figure 6.19: QR code generation process flow
CHAPTER 7

PROJECT TESTING

7.1 Introduction
This chapter discusses unit testing, static code analysis, code quality review, application audit and user acceptance testing.

7.2 Unit Testing
Unit testing has been conducted with all repositories. Jasmine is used as a testing framework for frontend application. It is a JavaScript testing framework that does not require a DOM. Karma is a testing tool which spawns a web server that executes source code against test code for each browser connected. Jasmine and Karma are used together for testing for Angular and Ionic.

WebUI repository

![Karma v3.0.0 - connected](image)

Figure 7.1: Test results on browser
Mobile repository

Figure 7.2: Test results on console

![Karma v3.1.4 - connected](image)

Mobile repository

Figure 7.3: Test results on browser

![Test results on console](image)

Backend repository

Figure 7.4: Test results on console
7.3 User Interface Testing

Table 7.1: UI test case

<table>
<thead>
<tr>
<th>Behaviour</th>
<th>Expected</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URL to enter a page without login</td>
<td>System should redirect to login page</td>
<td>System redirects user to login page</td>
</tr>
<tr>
<td>Invalid credentials to login</td>
<td>System should show error message</td>
<td>System shows error message</td>
</tr>
<tr>
<td>Valid credentials to login</td>
<td>System should redirect to authenticated page</td>
<td>System directs user authenticated page</td>
</tr>
<tr>
<td>Current class page only shows the current class</td>
<td>Only at exact time, the current class page should</td>
<td>The current class page shows the current class</td>
</tr>
</tbody>
</table>
information on the exact time of the class slot | only show the current class information | information during the exact time of the class
---|---|---
QR code configuration with auto closing dialog and refresh rate | System should close the dialog after 1 minutes with 5 seconds of QR code refresh rate | System closed the dialog after 1 minutes with and shows QR code with 5 seconds of refresh rate

7.4 **Static Code Analysis**

Static code analysis had been conducted throughout the whole development process. TSLint is used for TypeScript project for both lecturer and student view repositories.

TSLint is constantly and automatically analyse the code and prompt errors if the code written does not comply to the rules or style. This ensures that the code follows the best practice of the rules or style.

7.5 **Code Quality Review**

Code quality review had done automatically for all the codebase with Codebeat during the development. Full GPA is 4.0. The figures below shown the results for each repository.

![3.69 GPA on github.com/WLun001/webUI](image)

<table>
<thead>
<tr>
<th>complexity issues</th>
<th>duplications</th>
<th>style issues</th>
<th>lines of code</th>
<th>code coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>7</td>
<td>1</td>
<td>1137</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 7.6: Code review scoring for lecturer view repository
7.6 Application Audit

Google Lighthouse is an open-source tool for audit web pages. It is used for audit web app for both lecturer and student view.

Figure 7.7: Code review scoring for student view repository

Figure 7.8: Code review scoring for backend repository

Figure 7.9: Lighthouse audit report for lecturer view web app
User Acceptance Testing

User Acceptance Testing had been conducted for two classes, practical and lecture class. Total number students involved in the testing are 62 students, however only 26 of them response to the survey. Students are giving positive feedback on the system user interface, flow, meanwhile negative feedback on the completeness of the application. Due to the camera is not zoomable, some of the students found it difficult to scan from far from the projected screen.
Besides, 1 lecturer was involved and received 1 response to the survey. The respondent pointed out the convenience and accuracy of the system, with some unnecessary selection of inputs on UI. Further improvement can be made to the system.

The survey overall statistic is attached at appendix F.
CHAPTER 8

CONCLUSION AND DISCUSSION

8.1 Introduction
This chapter discusses about limitation of the project, future improvement and recommendation and conclusion.

8.2 Limitation
The project comes with few limitations as stated in the following:

• No secure connection on code generator server as secure connection requires a verified domain to get an SSL license.
• No HTTPS connection on lecturer web app as web service that the web app consume are not secure.

8.3 Improvement and Recommendation
There are several improvements and recommendations to be considered in the future development.

• Administrator view for manage timetable, users, and view resources consumed.
• Zoomable camera on student view web app.
• Develop as full-fledged progressive web app on all web apps.
• Develop in-deep data analysis feature to further analyse students’ attendance records.

8.4 Challenges
The challenges when developing the projects are listed as follows:

• Required a lot of self-learning to pick up libraries, frameworks and cloud services.
• Less similar implementations or solutions to refer to when designing the application.
8.5 Conclusion

In conclusion, taking attendance with QR code is the cheapest and adaptable options among all of the solutions. It does not require infrastructure changes to adapt it. With the popularity of the smartphone and internet accessibility, it can widely use in universities. With the automatically refresh QR code, cheating on attendance is become even difficult. Besides, it eliminates a lot of lecturers’ effort on managing students’ attendance records. The project objectives were achieved.
References


APPENDICES

APPENDIX A: Survey Question
Student Attendance System Survey

Thank you for agreeing to take part in this important survey. I am Yong Wei Lun, Software Engineering student from University Tunku Abdul Rahman (Sungai Long campus). I am currently working on my Final Year Project which is developing smart attendance system using QR Code. I would like to conduct this survey with you to obtain more information about the existing and similar system in your University or College. This survey will only take less than 3 minutes. For your information, the answer you provided will be taken into consideration for my project and the information you provided might be included in my project report. Your help is much appreciated and try your best to complete this survey form. Thank you!

* Required

1. What is the name of your University/College (full name) *

2. Which attendance system does your University/College using? *
   * Mark only one oval.
   - Automated attendance system
   - Sign attendance on attendance sheets
   - Other:

3. What do you think about current attendance system? *
   * Mark only one oval.
   - Cumbersome
   - Is fine
   - Other:

4. Do you think it's easy to cheat on attendance? *
   * Mark only one oval.
   - Yes
   - No
   - Maybe

5. Do you think attendance system should be automated? *
   * Mark only one oval.
   - Yes
   - No
   - Maybe
6. Do you think an automated attendance system will increase the productivity of lecturers and reduce human error? *

Mark only one oval.

☐ Yes
☐ No
☐ Maybe
☐ Other:

7. Do you think which tracking technology is the best to prevent fake attendance? *

Mark only one oval.

☐ Using QR Code in class, will be reset after short interval
☐ Polling registers students as present in class by requiring them to answer a polling question or enter a special code on their laptops or phones.
☐ Using geolocation
☐ Using Biometric
☐ Using student ID reader
☐ Other:

8. Do you think which attendance tracking technology is the most cost effective? *

Mark only one oval.

☐ Using QR Code
☐ Polling
☐ Geolocation
☐ Biometric reader
☐ Student ID reader
☐ Other:
APPENDIX B: Results of Survey
What is the name of your University/College (full name)

Universiti Sains Malaysia
London School of Economics
Kampus Kesihatan Universiti Sains Malaysia
Multimedia University
Sunway College University
The One Academy
University Malaysia Terengganu
University Utara Malaysia
UCSI UNIVERISTY
Nilai University
Nanyang Technological University
Universiti Malaya
Loma Linda University
University Kebangsaan Malaysia
Infrastructure University Kuala Lumpur
Tunku Abdul Rahman University College
Asia Pacific University
Newcastle University
HELP University
YPC International College
Which attendance system does your University/College using?
20 responses

What do you think about current attendance system?
20 responses

Do you think it's easy to cheat on attendance?
20 responses
Do you think attendance system should be automated?
20 responses

Do you think an automated attendance system will increase the productivity of lecturers and reduce human error?
20 responses

Do you think which tracking technology is the best to prevent fake attendance?
20 responses
Do you think which attendance tracking technology is the most cost effective?

20 responses

- Using QR Code in class, will be reset after short interval: 35%
- Polling registers students as present in class by requiring them to answer a polling question or enter a special code on their laptops or ph...: 25%
- Using geolocation: 20%
- Using Biometric: 15%
- Using student ID reader: 10%
APPENDIX C: Work Breakdown Structure
# Smart Attendance System

<table>
<thead>
<tr>
<th>Planning</th>
<th>0%</th>
<th>Start</th>
<th>Due</th>
<th>Assigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gather requirements</td>
<td>0%</td>
<td>May 28, 2018</td>
<td>Jun 3, 2018</td>
<td></td>
</tr>
<tr>
<td>Define scope</td>
<td>0%</td>
<td>Jun 5, 2018</td>
<td>Jul 9, 2018</td>
<td></td>
</tr>
<tr>
<td>Prepare preliminary report</td>
<td>0%</td>
<td>Jun 11, 2018</td>
<td>Jul 8, 2018</td>
<td></td>
</tr>
<tr>
<td>Develop proposal</td>
<td>0%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System design</td>
<td>0%</td>
<td>Jul 10, 2018</td>
<td>Sep 20, 2018</td>
<td></td>
</tr>
<tr>
<td>Completion on System architecture design</td>
<td></td>
<td>Sep 21, 2018</td>
<td>Sep 21, 2018</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Iteration 1</th>
<th>0%</th>
<th>Start</th>
<th>Due</th>
<th>Assigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prototyping</td>
<td>0%</td>
<td>Sep 24, 2018</td>
<td>Dec 7, 2018</td>
<td></td>
</tr>
<tr>
<td>Evaluation</td>
<td>0%</td>
<td>Dec 13, 2018</td>
<td>Dec 19, 2018</td>
<td></td>
</tr>
<tr>
<td>Improvement</td>
<td>0%</td>
<td>Dec 19, 2018</td>
<td>Jan 1, 2019</td>
<td></td>
</tr>
<tr>
<td>Completion on Iteration 1</td>
<td></td>
<td>Jan 2, 2019</td>
<td>Jan 2, 2019</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Iteration 2</th>
<th>0%</th>
<th>Start</th>
<th>Due</th>
<th>Assigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prototyping</td>
<td>0%</td>
<td>Jan 3, 2019</td>
<td>Feb 27, 2019</td>
<td></td>
</tr>
<tr>
<td>Evaluation</td>
<td>0%</td>
<td>Feb 28, 2019</td>
<td>Mar 6, 2019</td>
<td></td>
</tr>
<tr>
<td>Improvement</td>
<td>0%</td>
<td>Mar 7, 2019</td>
<td>Mar 12, 2019</td>
<td></td>
</tr>
<tr>
<td>Completion on Iteration 2</td>
<td></td>
<td>Mar 13, 2019</td>
<td>Mar 13, 2019</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prepare to deliver</th>
<th>0%</th>
<th>Start</th>
<th>Due</th>
<th>Assigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment</td>
<td>0%</td>
<td>Mar 14, 2019</td>
<td>Mar 19, 2019</td>
<td></td>
</tr>
<tr>
<td>Testing after deployment</td>
<td>0%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration testing</td>
<td>0%</td>
<td>Mar 20, 2019</td>
<td>Mar 21, 2019</td>
<td></td>
</tr>
<tr>
<td>System testing</td>
<td>0%</td>
<td>Mar 19, 2019</td>
<td>Mar 25, 2019</td>
<td></td>
</tr>
<tr>
<td>Load testing</td>
<td>0%</td>
<td>Mar 27, 2019</td>
<td>Apr 3, 2019</td>
<td></td>
</tr>
<tr>
<td>Completion of project</td>
<td></td>
<td>Apr 5, 2019</td>
<td>Apr 5, 2019</td>
<td></td>
</tr>
</tbody>
</table>

[View in Gantt](#)
APPENDIX D: Gantt Chart
# Smart Attendance System

<table>
<thead>
<tr>
<th>Task</th>
<th>Start</th>
<th>End</th>
<th>0h</th>
<th>0%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning</td>
<td>05/28</td>
<td>09/21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gather requirements</td>
<td>05/28</td>
<td>06/03</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Define scope</td>
<td>06/05</td>
<td>07/09</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Prepare preliminary report</td>
<td>06/11</td>
<td>07/08</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Develop proposal</td>
<td>07/08</td>
<td>08/20</td>
<td>0h</td>
<td>0%</td>
</tr>
<tr>
<td>System design</td>
<td>07/10</td>
<td>09/20</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Completion on System design</td>
<td>09/21</td>
<td>09/21</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Iteration 1</td>
<td>09/24</td>
<td>01/02</td>
<td>0h</td>
<td>0%</td>
</tr>
<tr>
<td>Prototyping</td>
<td>09/24</td>
<td>12/07</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Evaluation</td>
<td>12/13</td>
<td>12/19</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Improvement</td>
<td>12/19</td>
<td>01/01</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Completion on Iteration 1</td>
<td>01/02</td>
<td>01/02</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Iteration 2</td>
<td>01/03</td>
<td>03/13</td>
<td>0h</td>
<td>0%</td>
</tr>
<tr>
<td>Prototyping</td>
<td>01/03</td>
<td>02/27</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Evaluation</td>
<td>02/28</td>
<td>03/06</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Improvement</td>
<td>03/07</td>
<td>03/12</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Completion on Iteration 2</td>
<td>03/13</td>
<td>03/13</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Prepare to deliver</td>
<td>03/14</td>
<td>04/05</td>
<td>0h</td>
<td>0%</td>
</tr>
<tr>
<td>Deployment</td>
<td>03/14</td>
<td>03/19</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Testing after deployment</td>
<td>03/19</td>
<td>04/02</td>
<td>0h</td>
<td>0%</td>
</tr>
<tr>
<td>Integration testing</td>
<td>03/20</td>
<td>03/21</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>System testing</td>
<td>03/19</td>
<td>03/25</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Load testing</td>
<td>03/27</td>
<td>04/03</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Completion of project</td>
<td>04/05</td>
<td>04/05</td>
<td>0</td>
<td>0%</td>
</tr>
</tbody>
</table>
APPENDIX E: Lighthouse Audit
https://smart-attendance-system-a242d.firebaseapp.com/today
Mar 31, 2019, 3:07 PM GMT+8
No emulation, Simulated Slow 4G network

Performance

Metrics

- First Contentful Paint: 1,920 ms
- Speed Index: 2,770 ms
- Time to Interactive: 3,760 ms

- First Meaningful Paint: 1,960 ms
- First CPU Idle: 3,520 ms
- Estimated Input Latency: 262 ms

Diagnostics

More information about the performance of your application.

1. Has significant main thread work: 3,740 ms
2. JavaScript boot-up time is too high: 2,280 ms
3. Critical Request Chains: 5 chains found
4. User Timing marks and measures: 30 user timings

Passed audits

18 audits

Progressive Web App
These checks validate the aspects of a Progressive Web App, as specified by the baseline PWA Checklist.

1. Page load is fast enough on 3G
2. Does not respond with a 200 when offline
3. User will not be prompted to Install the Web App
   - Failures: No manifest was fetched, Site does not register a service worker.
4. Uses HTTPS
5. Redirects HTTP traffic to HTTPS
6. Has a `<meta name="viewport">` tag with width or `initial-scale`
7. Does not register a service worker
8. Does not provide fallback content when JavaScript is not available
   - The page body should render some content if its scripts are not available.
9. Is not configured for a custom splash screen
   - Failures: No manifest was fetched.
10. Address bar does not match brand colors
    - Failures: No manifest was fetched, No `<meta name="theme-color">` tag found.
11. Content is sized correctly for the viewport
12. The `short_name` won’t be truncated on the homescreen

**Accessibility**

These checks highlight opportunities to improve the accessibility of your web app. Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

**Meta Tags Used Properly**

These are opportunities to improve the user experience of your site.

1. `[user-scalable="no"]` is used in the `<meta name="viewport">` element or the `[maximum-scale]` attribute is less than 5.

**Passed audits**

**Not applicable**
Best Practices

Passed audits

SEO

These checks ensure that your page is optimized for search engine results ranking. There are additional factors Lighthouse does not check that may affect your search ranking. Learn more.

Content Best Practices

Format your HTML in a way that enables crawlers to better understand your app's content.

1. Document does not have a meta description

Crawling and Indexing

To appear in search results, crawlers need access to your app.

2. robots.txt is not valid

29 errors found

Additional items to manually check

Runtime settings

- URL: https://smart-attendance-system-a242d.firebaseapp.com/today
- Fetch time: Mar 31, 2019, 3:07 PM GMT+8
- Device: No emulation
- Network throttling: 150 ms TCP RTT, 1,638.4 Kbps throughput (Simulated)
- CPU throttling: 4x slowdown (Simulated)
- User agent (host): Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.102 Safari/537.36

Generated by Lighthouse 3.0.3 | File an issue
Performance Metrics are estimated and may vary.

Diagnostics

More information about the performance of your application.

1. Has significant main thread work
   - 3,700 ms

2. JavaScript boot-up time is too high
   - 2,600 ms

3. Critical Request Chains
   - 2 chains found

4. User Timing marks and measures
   - 26 user timings

Passed audits

18 audits

Progressive Web App

These checks validate the aspects of a Progressive Web App, as specified by the baseline PWA Checklist.
<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Page load is fast enough on 3G</td>
</tr>
<tr>
<td>2</td>
<td>Does not respond with a 200 when offline</td>
</tr>
<tr>
<td>3</td>
<td>User will not be prompted to Install the Web App</td>
</tr>
<tr>
<td></td>
<td>Failures: No manifest was fetched, Site does not register a service worker.</td>
</tr>
<tr>
<td>4</td>
<td>Does not use HTTPS</td>
</tr>
<tr>
<td></td>
<td>9 insecure requests found</td>
</tr>
<tr>
<td>5</td>
<td>Does not redirect HTTP traffic to HTTPS</td>
</tr>
<tr>
<td>6</td>
<td>Has a <code>&lt;meta name=&quot;viewport&quot;&gt;</code> tag with width or initial-scale</td>
</tr>
<tr>
<td>7</td>
<td>Does not register a service worker</td>
</tr>
<tr>
<td>8</td>
<td>Does not provide fallback content when JavaScript is not available</td>
</tr>
<tr>
<td></td>
<td>The page body should render some content if its scripts are not available.</td>
</tr>
<tr>
<td>9</td>
<td>Is not configured for a custom splash screen</td>
</tr>
<tr>
<td></td>
<td>Failures: No manifest was fetched.</td>
</tr>
<tr>
<td>10</td>
<td>Address bar does not match brand colors</td>
</tr>
<tr>
<td></td>
<td>Failures: No manifest was fetched, No <code>&lt;meta name=&quot;theme-color&quot;&gt;</code> tag found.</td>
</tr>
<tr>
<td>11</td>
<td>Content is not sized correctly for the viewport</td>
</tr>
<tr>
<td></td>
<td>The viewport size is 958px, whereas the window size is 1477px.</td>
</tr>
<tr>
<td>12</td>
<td>The short_name won't be truncated on the homescreen</td>
</tr>
</tbody>
</table>

### Additional items to manually check

3 audits

### Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

#### Elements Have Discernible Names

These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Buttons do not have an accessible name</td>
</tr>
<tr>
<td>2</td>
<td>Links do not have a discernible name</td>
</tr>
</tbody>
</table>

#### Color Contrast Is Satisfactory

These are opportunities to improve the legibility of your content.

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Background and foreground colors do not have a sufficient contrast ratio.</td>
</tr>
</tbody>
</table>
### Best Practices

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Does not use HTTPS</td>
<td>9 insecure requests found</td>
</tr>
<tr>
<td>2</td>
<td>Does not use HTTP/2 for all of its resources</td>
<td>9 requests not served via HTTP/2</td>
</tr>
<tr>
<td>3</td>
<td>Links to cross-origin destinations are unsafe</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Browser errors were logged to the console</td>
<td></td>
</tr>
</tbody>
</table>

### Passed audits

11 audits

### SEO

These checks ensure that your page is optimized for search engine results ranking. There are additional factors Lighthouse does not check that may affect your search ranking. [Learn more.]

### Content Best Practices

Format your HTML in a way that enables crawlers to better understand your app's content.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Document does not have a meta description</td>
</tr>
</tbody>
</table>

### Crawling and Indexing

To appear in search results, crawlers need access to your app.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Page has unsuccessful HTTP status code 404</td>
</tr>
</tbody>
</table>

### Runtime settings

- **URL**: http://sas-qrcode.s3-website-ap-southeast-1.amazonaws.com/pages/timetable
Fetch time: Mar 31, 2019, 3:06 PM GMT+8
Device: No emulation
Network throttling: 150 ms TCP RTT, 1,638.4 Kbps throughput (Simulated)
CPU throttling: 4x slowdown (Simulated)
User agent (host): Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.102 Safari/537.36

Generated by Lighthouse 3.0.3 | File an issue
APPENDIX F: User Satisfaction Survey
User Satisfaction Survey for Attendance System
(Student)

My name is Yong Wei Lun, final year student in Software Engineering. Thanks for testing out my final year project, Smart Attendance System using QR code. Kindly fill in the survey form. Your cooperation is high appreciated.

* Required

1. I think that I would like to use this web app for attendance registration *
   Mark only one oval.
   - Yes
   - No
   - Maybe

2. I found the web app unnecessarily complex. *
   Mark only one oval.
   1 2 3 4 5
   Strongly Disagree Strongly Agree

3. I thought the web app was easy to use *
   Mark only one oval.
   1 2 3 4 5
   Strongly Diagree Strongly Agree

4. I think that I would need the support of a technical person to be able to use this web app.
   Mark only one oval.
   - Yes
   - No
   - Maybe

5. I found this web app was easily moved through without a lot of backtracking or data re-entry. *
   Mark only one oval.
   1 2 3 4 5
   Strongly Disagree Strongly Agree

6. I thought there was too much inconsistency in this web app *
   Mark only one oval.
   1 2 3 4 5
   Strongly Disagree Strongly Agree
7. I would imagine that most people would learn to use this web app very quickly. *  
   Mark only one oval.

   1 2 3 4 5

   Strongly Disagree □ □ □ □ □ Strongly Agree

8. I found the web app very awkward to use. *  
   Mark only one oval.

   1 2 3 4 5

   Strongly Disagree □ □ □ □ □ Strongly Agree

9. I felt very confident using the web app *  
   Mark only one oval.

   1 2 3 4 5

   Strongly Disagree □ □ □ □ □ Strongly Agree

10. I needed to learn a lot of things before I could get going with this web app *  
    Mark only one oval.

    1 2 3 4 5

    Strongly Disagree □ □ □ □ □ Strongly Agree

11. What did you like best about the web app?

    ____________________________
    ____________________________
    ____________________________
    ____________________________

12. What did you like least about the web app?

    ____________________________
    ____________________________
    ____________________________
    ____________________________
13. If you were to describe this web app to a colleague in a sentence or two, what would you say?

14. Do you have any other final comments or questions?
User Satisfaction Survey for Attendance System (Student)
26 responses

I think that I would like to use this web app for attendance registration
26 responses

I found the web app unnecessarily complex.
26 responses
I thought the web app was easy to use
26 responses

I think that I would need the support of a technical person to be able to use this web app.
26 responses

I found this web app was easily moved through without a lot of backtracking or data re-entry.
26 responses
I thought there was too much inconsistency in this web app

26 responses

I would imagine that most people would learn to use this web app very quickly.

26 responses
I found the web app very awkward to use.

26 responses

I felt very confident using the web app

26 responses

I needed to learn a lot of things before I could get going with this web app
What did you like best about the web app?

19 responses

- Simple
- Nothing at all
- Nothing
- Convenience on signing attendance
- Easy to use
- Environment friendly as students need not sign on paper, reducing paper wastage
- Easy
- Nice user interface.
- The fact that we do not have to sign for attendance
- High tech
- Ease of usage.
- User friendly and easy to use.
- Using camera to scan QR code for attendance
- QR code
- Replace manual attendance
- Scan barcode for attendance
What did you like least about the web app?

21 responses

No
Everything there is to it
Weird position of the header
Need internet connection for signing
Default setting of front camera
Cannot help friends sign
Camera cannot zoom, students sit far from the screen will have to move nearer to the screen just to get the qr code. Also preferable if can fix a qr code since the qr changes once its being capture, causing the increase of time consumption

Camera
Refresh rate of qr code
The default camera is the front camera. The circle should include on the middle bottom should include an icon to indicate camera. If invalid, the message should close automatically after a period of time.
The QR code expires after one person scans it

Not so convenience
Troublesome to scan qr code.

Camare
Try to allow users for scanning QRcode to take attendance, instead of asking user to enter URL
Attendance page need to be refreshed after scanning, address too long
User interface can be improved (after scan ckode, the done cannot be seen)

Simple interface

Need Internet, need to type in the page

User interface
If you were to describe this web app to a colleague in a sentence or two, what would you say?

15 responses

- Easy to use
- Nothing good
- Well thought but implementation can be done better
- Nice website
- Convenient
- Still need improvement
- Easy, efficient and effective to record attendees
- Good.
- Just try it
- Attendance system
- Good for record attendance.
- It's fine
- Nice app

Do you have any other final comments or questions?

14 responses

- No
- No thanks ^^
- Instead of retrying manually, maybe retry it automatically and after few fail attempts only require retry manually
- Nope
- Well done
- Camera cannot zoom cannot focus and a bit hard to scan
- Can be used with same ip adress, thus still able to help others
- I do not think this can solve the problem where students are 'signing' for their friends as one person can easily log in to multiple accounts using one device.
The problem of scanning the qr code must be solved before proper use.

Maybe cannot help people to sign attendance if this system is success

Hope that the camera can have enlarge function.

Got few pages back button missing got function but no icon

Qr code refreshing interval is too short. Imagining it keeps on refreshing and not every phone has high specification. 1.Lecturer tend to spend time on giving students to scan. 2. What if people came late or have to leave early, what would happened? It wouldn't be fair to everyone. Barred chance will be increasing than usual.
User Satisfaction Survey for Attendance System (Lecturer)

1 response

I think that I would like to use this web app for attendance registration

1 response

I found the web app unnecessarily complex.
I thought the web app was easy to use
1 response

I think that I would need the support of a technical person to be able to use this web app.
1 response

I found this web app was easily moved through without a lot of backtracking or data re-entry.
1 response
I thought there was too much inconsistency in this web app
1 response

I would imagine that most people would learn to use this web app very quickly.
1 response
I found the web app very awkward to use.

1 response

I felt very confident using the web app

1 response

I needed to learn a lot of things before I could get going with this web app
What did you like best about the web app?

1 response

convenient, accurate

What did you like least about the web app?

1 response

some unnecessary selection of input widget (eg checkbox and scroll value in the configuration)

If you were to describe this web app to a colleague in a sentence or two, what would you say?

1 response

good to have, cheapest option
Do you have any other final comments or questions?

1 response

Further analysis of student attendance, eg list of students, achieved < 80% attendance to be alerted and able to see the attendance pattern in other subjects.
Appendix G: Supervisor’s comments
<table>
<thead>
<tr>
<th>Key Assessment for Project Proposal</th>
<th>Supervisor Comments/Remarks</th>
<th>Moderator Comments/Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Description</td>
<td>Include the short survey (observation of current practices) and its result in the problem statement. Proposed solution: show the process in which current practices is automated by the implemented solution. Proposed approach: cite/relate some similar works/projects that use evolution prototyping approach in development.</td>
<td></td>
</tr>
<tr>
<td>Project Scope and Objectives</td>
<td>ok</td>
<td>improve on the review of technologies used.</td>
</tr>
<tr>
<td>Literature Review / Fact Finding for Benchmarking / Verification of Project</td>
<td>Technologies review are too short. Combined with the tools reviewed. Simplify the writing about the development methodologies review.</td>
<td></td>
</tr>
<tr>
<td>Research/Development Methodology and Development Tools</td>
<td>too simple. The diagram is not complete. Give some explanations of planned activities in each phase prototyping. Identify the outcomes/deliverables in each iteration. Tools used: have explained each tool used; lacking of analysis to help in deciding the tools used.</td>
<td></td>
</tr>
<tr>
<td>Project Plan</td>
<td>Should show work breakdown structures in hierarchical decomposition of the work. To understand the overall plan, show the important milestones/phases with its start date, completion date and resources to be used in a table.</td>
<td></td>
</tr>
<tr>
<td>Initial Deliverables</td>
<td>- show the current process (based on the observation and document reviews made) and the proposed process workflow. This is to meet the objective #1.</td>
<td></td>
</tr>
<tr>
<td>Report Structure and References</td>
<td>All diagrams, tables and appendices must refer in the report content. Inconsistency of font type and size. Content about survey and its result should move to Chapter 1 as it is mainly to help to formulate problems.</td>
<td></td>
</tr>
<tr>
<td>Language and Clarity of Writing</td>
<td>contain some incomplete sentences. Need to use different variety of conjunction to join the explanation, argument etc. Contains a lot of similar structure of languages.</td>
<td></td>
</tr>
</tbody>
</table>