

DEVELOPING ANDROID APPLICATION TO GUIDE LEAN SIX SIGMA PDCA PROJECT

By:

ONG SIN JOO

(Matrix no.: 125428)

Supervisor:

Associate Professor Ir. Dr Chin Jeng Feng

May 2018

This dissertation is subjected to

Universiti Sains Malaysia

As partial fulfillment of the requirement to graduate with honors degree in

**BACHELOR OF ENGINEERING (MANUFACTURING ENGINEERING WITH
MANAGEMENT)**



School of Mechanical Engineering

Engineering Campus

Universiti Sains Malaysia

Declaration

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed (ONG SIN JOO)

Date

Statement 1

This thesis is the result of my own investigation, except where otherwise stated. Other sources are acknowledged by giving explicit references. Bibliography/references are appended.

Signed (ONG SIN JOO)

Date

Statement 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for interlibrary loan, and for the title and summary to be made available outside organizations.

Signed (ONG SIN JOO)

Date

Acknowledgment

I would like to take this opportunity to specially thank and appreciate my supervisor A.P. Ir. Dr. Chin Jeng Feng for his patient guidance, enthusiastic encouragement and useful critiques. Without his advice and assistance in providing necessary information regarding this project, progress on schedule could never be made in this thesis. With his guidance and support along this course, I am able to gain new knowledge and skills in this project. I also would like to express my appreciation to all supporting people and friends who have been a constant source of encouragement for successful completion of this project. A special thanks to 10 students from yellow belt and advanced lean practitioner for giving opportunity to conduct my guideline app testing and providing their valuable ideas and feedback. Lastly, I would like to thank to my family that always stay with me.

Table of Contents

Declaration	i
Acknowledgment	ii
List of Figures	vi
List of Tables	ix
List of Abbreviations	x
Abstract (BM)	xi
Abstract (BI)	xii
CHAPTER 1: INTRODUCTION	1
1.1 Research Background.....	1
1.2 Problem Statement	6
1.3 Objective	6
1.4 Scope of Work.....	6
CHAPTER 2: LITERATURE REVIEW	7
2.1 Chapter Overview	7
2.2 Plan -Do-Check-Act Cycle.....	7
2.2 Kaizen Concept	9
2.3 Lean Manufacturing Tool.....	10
2.4 Chapter Summary.....	11
CHAPTER 3: METHODOLOGY	12
3.1 Chapter Overview	12
3.2 Development Approach.....	12
3.2.1 Android Software Development Tool Selection	12
3.2.2 Exploration on MIT App Inventor.....	17
3.2.3 PDCA Guideline Collection	24

3.2.3.1 Project Team Formation	24
3.2.3.2 Plan Phase	25
3.2.3.3 Do Phase	53
3.2.3.4 Check Phase	55
3.2.3.5 Act Phase	56
3.2.4 User Interface Design	57
3.2.4.1 Fusion Table.....	58
3.2.5 Coding and Testing.....	62
3.2.6 Implementation.....	62
3.2.7 Result Analysis	62
3.2.8 Future Development	62
3.3 Chapter Summary.....	63
CHAPTER FOUR: MODEL DEVELOPMENT	64
4.1 Chapter Overview	64
4.2 Application Algorithm	64
4.3 Application Development	69
4.3.1 Start Screen.....	69
4.3.2 Login Screen.....	70
4.3.3 Register Screen	71
4.3.4 PDCA Screen.....	72
4.3.4.1 Create Team Screen	72
4.3.4.2 Step 1 Screen.....	73
4.3.4.3 Step 2 Screen.....	75
4.3.4.4 Step 3 Screen.....	76
4.3.4.5 Step 4 Screen.....	77

4.3.4.6 Step 5 Screen.....	79
4.3.4.7 Step 6 Screen.....	80
4.3.4.8 Step 7 Screen.....	81
4.3.4.9 Step 8 Screen.....	82
4.3.4.10 Step 9 Screen.....	82
4.3.4.11 Step 10 Screen.....	83
4.3.4.12 Step 11 Screen.....	84
4.3.4.13 Step 12 Screen.....	85
4.3.5 Fusion Table Screen	86
4.4 Chapter Summary.....	86
CHAPTER FIVE: RESULT AND ANALYSIS	87
5.1 Chapter Overview	87
5.2 Users Analysis.....	87
5.3 Survey Numerical Analysis.....	89
5.4 Survey Comment Analysis.....	91
5.5 Chapter Summary.....	93
CHAPTER 6: CONCLUSIONS AND RECOMMENDATION	94
6.1 Chapter Overview	94
6.2 Conclusion.....	94
6.3 Recommendations	94
Reference	95
Appendix A: System Usability Scale (SUS) Survey Form.....	100
Appendix B: MIT App Inventor Block Program.....	101

List of Figures

Figure 2. 1: Kaizen Umbrella-Shaped Concept	10
Figure 3. 1: Example of Software Templates Provided in AppsGeyser	14
Figure 3. 2: Graphical Interface Designer Screen View of AppMakr	15
Figure 3. 3: Graphical Interface Designer Screen View of Shoutem	15
Figure 3. 4: Graphical Interface Designer and Block Editor of MIT App Inventor	16
Figure 3. 5: MIT App Inventor Website	17
Figure 3. 6: Main panel of MIT App Inventor (Graphical interface designer).....	18
Figure 3. 7: Visible and Non-Visible Components.....	19
Figure 3. 8: Main panel of Block Editor MIT App Inventor	20
Figure 3. 9: Event Handler Blocks.....	21
Figure 3. 10: Setter (Left) and Getter (Right) Commands Blocks.....	22
Figure 3. 11: Puzzle Piece Matching	22
Figure 3. 12: Icon of MIT App Inventor Companion	23
Figure 3. 13: Choose AI Companion to Connect.....	23
Figure 3. 14: Connect and Scan with QR Code	23
Figure 3. 15: Example of Process Flowchart	31
Figure 3. 16: Histogram	33
Figure 3. 17: Pareto Chart.....	34
Figure 3. 18: Run Chart	35
Figure 3. 19: Graph of Workload Balancing	37
Figure 3. 20: Relationship between Target Condition and Challenge	38
Figure 3. 21: Cause & Effect Diagram	40
Figure 3. 22: 5 Why Structure.....	41
Figure 3. 23: Scatter Diagram.....	42
Figure 3. 24: Effort and Impact Matrix.....	44
Figure 3. 25: Pull System Kanban Flow	47
Figure 3. 26: Steps of Jidoka.....	50
Figure 3. 27: Process Performance between Before and After Improvement	55
Figure 3. 29: Create Fusion Tables	59
Figure 3. 30: Create an Empty Fusion Table	59

Figure 3. 31: Edit the Columns in the Fusion Table	60
Figure 3. 32: Fusion Table Sharing.....	60
Figure 3. 34: Fusion Table ID.....	61
Figure 3. 35: Iterative Development Methodology.....	63
Figure 4. 1: Algorithm of PDCA Guideline.....	68
Figure 4. 2: Start Screen.....	69
Figure 4. 3: Login Screen.....	70
Figure 4. 4: Register Screen.....	72
Figure 4. 5: Create Team Screen View	73
Figure 4. 6: Step 1 Screen View	74
Figure 4. 7: Step 2 Screen View	76
Figure 4. 8: Step 3 Screen View	77
Figure 4. 9: Step 4 Screen View	79
Figure 4. 10: Step 5 Screen View	80
Figure 4. 11: Step 6 Screen View	81
Figure 4. 12: Step 7 Screen View	81
Figure 4. 13: Step 8 Screen View	82
Figure 4. 14: Step 9 Screen View	83
Figure 4. 15: Step 10 Screen View	83
Figure 4. 16: Step 11 Screen View	84
Figure 4. 17: Step 12 Screen View	85
Figure 4. 18: Fusion Table Screen View	86
Figure 5. 1: Scatter Plot of SUS Score by Students.....	89
Figure 5. 2: Storage Component in MIT App Inventor	92
Figure 5. 3: Instruction of Update and Retrieve Table	92
Figure A. 1: SUS Survey Form.....	100
Figure B. 1: Start Screen Block Program.....	101
Figure B. 2: Login Screen Block Program.....	101
Figure B. 3: Register Screen Block Program.....	102
Figure B. 4: Fusion Table Webviewer Block Program.....	102
Figure B. 5: Fusion Table Delete Row Block Program	103

Figure B. 6: Create Team Screen Initialize Block Program	104
Figure B. 7: Category of Problem Block Program (Repeated for other buttons)	105
Figure B. 8: Insert Row into Fusion Table Block Program	106
Figure B. 9: Update Fusion Table Block Program (Repeated for other steps)	106
Figure B. 10: Fusion Table Got Result after Update Block Program	107
Figure B. 11: Retrieve Data from Fusion Table Block Program (Repeated for other steps)	107
Figure B. 12: Fusion Table Retrieve Data to App Block Program	107

List of Tables

Table 1. 1: Sigma Performance Level.....	2
Table 3. 1: Overview of Built-In Blocks	22
Table 3. 2: Role and Responsibility of Team Member	24
Table 3. 3: Voice of Customer to Critical to Customer Conversion.....	25
Table 3. 4: Category of Problem.....	26
Table 3. 5: Matching of Lean Data Collection Tool with Category of Problem	28
Table 3. 6: Common Symbols Used in Process Flowchart.....	30
Table 3. 7: Continuous Data of Histogram	33
Table 3. 8: Categorize the Histogram Data.....	33
Table 3. 9: Data Collection of Pareto Chart.....	34
Table 3. 10: Data collection of Check Sheet.....	36
Table 3. 11: Rating Scale	43
Table 3. 12: Criteria Weight	43
Table 3. 13: Prioritization Matrix	44
Table 3. 14: Available Countermeasures	45
Table 3. 15: Steps of SMED	48
Table 3. 16: 5S Explanation.....	51
Table 3. 17: Implementation Countermeasure Plan.....	54
Table 3. 18: Comparison between Before and After Improvement.....	55
Table 5. 1: PDCA Project of Yellow Belt Group	87
Table 5. 2: PDCA Project of Advanced Lean.....	88
Table 5. 3: Table of SUS Scores.....	89
Table 5. 4: Average Score of Survey Question.....	90

List of Abbreviations

ALP	Advanced Lean Practitioner
AI	Artificial Intelligence
CTQ	Critical-To-Quality
C-N-X	Control-Noise-Experiment
DPMO	Defects per Million Opportunities
DMAIC	Define-Measure-Analyze-Improve-Control
Etc	Et Cetera
HTC	High Tech Computer
IPO	Input-Process-Output
ID	Identity
MIT	Massachusetts Institute of Technology
NVA	Non-Value Added
NVABR	Non-Value-Added but Required
PDCA	Plan-Do-Check-Act
QR	Quick Response
SUS	System Usability Scale
SMART	Specific + Measurable + Attainable + Relevant + Timely
SOP	Standard Operating Procedure
SMED	Single Minute Exchange of Die
SME	Subject Matter Expertise
URL	Uniform Resource Locato
UI	User Interface
VA	Value Added
WIP	Work-in-Progress

Abstract (BM)

Lean Six Sigma dapat dipelajari melalui latihan atas talian, kursus universiti dan program latihan syarikat. Tiada garis panduan yang disediakan bagi pengamal untuk menjalankan *Plan-Do-Check-Act (PDCA)*. Ini selalu menyebabkan kekeliruan dan menyalahgunakan alat *lean*. Aplikasi mudah alih berkembang pesat dengan banyak pembangunan aplikasi yang sedia ada. Dalam kajian ini, pendekatan yang berasaskan Android telah dibanguankan untuk menyediakan garis panduan *PDCA*. Institut Teknologi *Massachusetts (MIT) App Inventor* telah dipilih sebagai alat pembangunan aplikasi dalam projek ini. Garis panduan *PDCA* termasuk alat *lean* yang berkaitan bagi peringkat yang berbeza telah dikumpulkan daripada kajian literatur. Permukaan grafik telah direka dengan mengikuti algoritma aplikasi. Kajian percubaan aplikasi ini telah dijalankan ke atas tujuh projek daripada enam orang pelajar *yellow belt* dan empat orang pelajar *advanced lean*. Keputusan yang diperolehi melalui borang soal selidik skala system kebolegunaan telah dianalisis dan dibincangkan. Markah skala system kebolegunaan yang telah diperolehi daripada sepuluh orang pelajar adalah dalam lingkungan 52.5 hingga 67.5 telah dianggap sebagai aplikasi yang sederhana dan memerlukan penambahbaikan. Penambahbaikan dibuat dengan sewajarnya. Sebagai sumbangan kepada praktikal, kerja untuk menghasilkan aplikasi mudah alih ini dapat membantu para pengamal pada peringkat awal perjalanan ke *lean* dalam industri. Aplikasi mudah alih ini menyediakan garis panduan bagi mereka untuk memastikan metodologi dalam penyelesaian masalah. Sesetengah cadangan telah disediakan untuk pembangunan masa depan.

Abstract (BI)

The knowledge of Lean Six Sigma can be learned through online training, university course and company training program. There is no existing guideline available for practitioner to carry out the Plan-Do-Check-Act (PDCA). This often leads to confusions and misuse of lean tools. Mobile application grows rapidly with abundant of application development. This research aims to develop an Android-based approach to provide guidelines of PDCA. Massachusetts Institute of Technology (MIT) App Inventor was selected as the application development platform. Guidelines to PDCA, including the associated lean tools for different stages were collected from the literature review. The graphical user interface was designed by follow with application algorithm. Testing of the application was conducted on seven projects from six yellow belt and four advanced lean students. The results obtained through the system usability scale (SUS) survey were analyzed and discussed. The SUS score obtained from the seven projects were in a range of 52.5 to 67.5 which considered as a moderate system and seek for improvement. Improvements were made accordingly. As the practical contribution, the work produces a ready to use mobile application to help practitioners in particular those in the industry at the early stage of lean journey. The mobile application provides guidance for them to ensure methodological problem solving. Some recommendations were suggested for future development.

CHAPTER 1: INTRODUCTION

1.1 Research Background

Lean manufacturing becomes popular for past few years among the manufacturing field. In Japan, after the Second World War, Japanese manufacturers could not afford high investment in company facilities as the United States of America Company. Therefore, the Japanese companies, especially Toyota developed the Lean manufacturing concept. Taiichi Ohno and Shigeo Shingo at Toyota have originated Toyota Production System, another name of Lean manufacturing (Pavnaskar et al., 2003). The Lean manufacturing is a concept of the elimination of waste from the flow to achieve the target of continuous process improvement (Tenera and Pinto, 2014). In practice, the value of the product has been enhanced through the elimination of waste. Lean manufacturing defined as methodology in classify the Value-Added Activity (VA) and Non-Value-Added Activity (NVA). The NVA means as the activity that does not add value to the flow and product (Sundar et al., 2014). The 8 forms of waste are Transport, Inventory, Motion, Waiting, Over-production, Over-processing, Defect and Underutilization. Basically, Lean manufacturing focus on 7 forms of waste without underutilization because underutilization of labor is a type of waste that does not appear on every project. According to Pavnaskar et al. (2003), the advantages of Lean manufacturing are product quality improvement, cycle time minimization, work in progress (WIP) reduction, delivery on-time, higher profits, cost saving, inventory elimination, higher productivity, enhance space flexibility and better labor utilization.

Lean Six Sigma combines two continuous improvement philosophies: Lean and Six Sigma. Six Sigma is the concept of reducing variation of process and product to eliminate the defects by investigating the problem and root cause. It is a project format in problem detection approach of defining, measuring, analyzing, improving and controlling. After identifying the problem in the Define phase, specific tools are available to measure the current condition and root cause analyzing. Then, the root cause will be eliminated with an implementation plan. Lastly, it is to achieve the saving and sigma level of Critical-To-

Quality (CTQ) in the Control phase. From Table 1.1, it's shown that the performance of sigma level from one sigma until six sigma. The comparing performances for each sigma level are quality rate, defect rate and defects per million opportunities (DPMO). Six Sigma was applied due to it has lowest defect rate in 3.4 DPMO and high quality rate at 99.9997% by comparing to another sigma level (Youssof et al., 2014). Finally, Lean Six Sigma integrate the application of eliminating waste tool and problem detection approach from both Lean and Six Sigma to reduce process variability and waste reduction (Cherrafi et al., 2016).

Table 1. 1: Sigma Performance Level

Sigma	Quality Rate (%)	Defect Rate (%)	DPMO
1	30.9	69.1	691462
2	69.1	30.9	308538
3	93.9	6.7	66807
4	99.38	0.62	6210
5	99.977	0.023	233
6	99.9997	0.00034	3.4

Lean Six Sigma has been applied to a lot of business operation from management, manufacturing, and service (Dr. Dharmalingam S, 2015). Steere et al. (2018) applied Lean Six Sigma in the hospital to improve the hospital management, elimination of unnecessary medication, reduction of treatment waiting time and cost saving. Pugna et al. (2016) used Lean Six Sigma in an automotive company to improve the assembly process for well tool handling and reduction of the defect. Indrawati and Ridwansyah (2015) solved the problem of inappropriate processing, defects and a high percentage of non-value-added activity in the iron ores industry by applying Lean Six Sigma. Sagnak and Kazancoglu (2016) enhance the efficiency of the environment by integrating lean six sigma and green approach to reduce the environmental pollution and minimize the product and service impact of unfavorable ecological. Simanová (2015) implement the methodology of Lean Six Sigma in furniture manufacturing to improve and assure the quality management system of

selected production processes. Lean Six Sigma has been invested by a lot of companies to retain competitive advantages (Manzouri et al., 2014).

Alhuraish et al. (2017) researched about the integration of Lean and Six Sigma methodologies via an online survey of experts who have the knowledge of lean manufacturing and Six Sigma. From the survey result, it was found that 9 companies out of 33 companies have been implemented Lean Six Sigma methodologies. The reasons for implementing both methodologies together are delivered rapid growth of process improvement, enhanced the customer satisfaction, raising of sales and enhancement of safety, sustainability and time savings. The experts stated that little and weak of improvement will be obtained from the separate implementation of Lean and Six Sigma methodologies. The results had shown that 4 companies out of 33 companies which had not implemented any methodologies from both. It was due to lack of investment, time and expertise from Lean and Six Sigma especially small and medium-sized enterprises.

The main resources required for Lean Six Sigma implementation are an expert who provided guidance and leading to continuous improvement. An education and training should be offered to an employee and students to get started with Lean Six Sigma knowledge learning. One can learn the training through online, university course and company training program. The most common Lean Six Sigma training courses available were – Lean Six Sigma Yellow Belt, Lean Six Sigma Green Belt, and Lean Six Sigma Black Belt. The Lean Six Sigma Yellow Belt training is aimed at those new to the world of Six Sigma who has a small role, interest, or need to develop foundational knowledge. Next, the Lean Six Sigma Green Belt operates in support or under the supervision of a Lean Six Sigma Black Belt in analyzes and solves quality problems in quality improvement projects. Last for the Lean Six Sigma Black Belt, it is a professional who can explain Six Sigma philosophies and principles, including supporting systems and tools, demonstrate team leadership, understand team dynamics and assign team member roles and responsibilities. After finishing the education and training, one must handle projects as experience in continuous process improvement with Lean Six Sigma. People learn from a project, not training - Ken Norton. If someone trained without an experience of project handling, it was considered as a waste and a form of over-production from Lean training.

For one company, they will hire an expert or Lean trained people as the human resource in implementing Lean and Six Sigma methodologies for further improvement and savings.

The most common methodology to be used in Lean Six Sigma is Define-Measure-Analyze-Improve-Control cycle (DMAIC). It is consisting of 5 interconnected processes in defining the right problem, measure process parameters and performance, analyze root cause, improve the process performance and control the gain. Basically, this methodology is used for large complexity of the problem. Therefore, it is not easy to manage and understand. According to M. Sokovic, D. Pavletic, and K. Kern Pipan, a continuous improvement tool called Plan-Do-Check-Act cycle (PDCA) is a simple tool that easy to understand by a large number of people. The PDCA cycle is a classic problem-solving methodology used in a Lean environment. Compared to DMAIC cycle, this methodology focuses on low complexity of the problem (Sokovic et al., 2010). Besides that, Define-Measure-Analyze-Improve-Control cycle is appropriately applied by Lean Six Sigma higher levels such as Green Belt and Black Belt. Lean Six Sigma Yellow Belt certified people suitable used PDCA cycle as their problem-solving methodology. Therefore, in this project, the PDCA cycle will be the main problem-solving methodology to be studied.

Undoubtedly, we live in the information and communication system era, computer and mobile application make the world inseparable. However, the mobile application is growing rapidly and gaining more popularity every year. A small hand holds mobile phone is no longer just a communication tool but also has the benefits with thousands of applications available for download and added unlimited fun for people's life (Md. Rashedul Islam, 2010). Now the Android application becomes more popular due to availability of free application development and abundant of application produced for touch screen mobile devices. This highly encourages people to use Android application. Moreover, Android system has the platform for application development which allowed Android application to gain further development with various applications of Android (Ma et al., 2014). Android developers can create their own application by writing application code in the Android application builder then generate useful and user friendly Android application (Singh et al., 2016).

Android is an open source mobile operating system based on the Linux kernel and developed by Google. Android is launched by Open Handset Alliance, which make up of more than 30 mobile phone technology companies involving Google, High Tech Computer (HTC), Telekom-Mobile, Qualcomm, Motorola, Samsung, China Mobile and Lucky Goldstar (LG). According to Open Handset Alliance, it declared that Android users can use Android application builder to innovate and develop new mobile application. Android becomes the first open and free mobile application platform (Guo-Hong, 2014). Now, Android has been widely used to develop various Android applications:

- Dong and Liu (2013) introduced Mandarin Language Study application with Android operating system. This application consists of various vocabulary lessons and multiple-choice questions. All the Mandarin words in this application are collecting from a Kwantlen Polytechnic University's textbook.
- Samal and Jena (2014) presented a mobile shop application by Android application. This mobile shop application improves the billing process faster and save a lot of time from the customer queue waiting time. All the selected products will generate a bar code. The billing process can be done by scanning the bar code and come out with a receipt.
- Mukherjee1 et al. (2015) developed a data saving application. The basic operations of application are creating, sort, view and delete the notes. Special features added are secure with password, voice navigation and voice searching for the notes.
- Rao and Krishna (2014) proposed a mobile health application to provide a health recommendation and exercise guidance. It is also suggested food tip for a different body mass index of people. Several functions also provided are body mass index and basal metabolic rate calculation, caloric expedition and report.

1.2 Problem Statement

After one finished Lean education and training, he/she will begin to manage a task. There have many sorts of improvement tools in the Lean Six Sigma that can offer a different form of improvement result. Even so, the practitioner often confused where and how to get going even though the education and training had covered many techniques and improvement tools. A readable and detailed starting guidance didn't provide on how to embark on going through their Lean knowledge and skills (Alhuraish et al., 2017). The education and training just involved in teaching and explaining of all improvement tools and problem-solving techniques by paper note supply and own takedown. In that respect are several problem-solving guidelines available on the net, however, it cannot be found in mobile application. The practitioner is easy to get lost on how to get started on problem-resolving. Thus, it is preferable for on-hand guideline provided. Nowadays Android has been widely applied to acquire plenty of an Android application. However, there is none of Android application which provides a problem-solving guideline.

1.3 Objective

To design an Android application in providing the guidelines of PDCA problem solving where the practitioners can refer to.

1.4 Scope of Work

In this project, Lean Six Sigma Yellow Belt problem-solving approach which is the Plan-Do-Check-Act will be studied. The detail guidance on PDCA steps will be furnished in the Android application. The appropriate tools to be applied in every step were prepared. An overlapping detection was created to prevent functional redundancy of tool deployment. The application created should be user-friendly and interactive within a squad.

CHAPTER 2: LITERATURE REVIEW

2.1 Chapter Overview

This chapter reviews and investigates the necessary guideline of PDCA that are required to carry on this thesis. Foremost, there will be topics regarding the steps of PDCA in order to infer the problem-solving methodologies. Kaizen concept will also be described, serving as a support for the understanding of PDCA. Lastly, tools that are going to be admitted in the guidelines will be determined.

2.2 Plan -Do-Check-Act Cycle

The PDCA cycle methodology was created by Walter A. Shewhart, an American statistical expert (Silva et al., 2017) and William Edward Deming, an American statistician (Sangpikul, 2017). This methodology had been implemented in Japanese companies in 1950s to utilize the PDCA concept successfully (Lodgaard and Aasland, 2011). The PDCA cycle also named Deming cycle, which named after William Edward Deming. At first, this methodology is employed as a tool in the area of quality control and management. But, it was accepted as an improvement methodology soon afterwards by applying in another field of management and continuous improvement had achieved (Sangpikul, 2017). The PDCA cycle had been defined as the major tool in problem solving and continuous improvement. According to Lodgaard and Aasland (2011), PDCA cycle is a systematic approach as a continuous improvement tool. Today, this tool has been expanded to be used widely in the many areas of industry, manufacturing and service. Silva et al. (2017) implement PDCA cycle for reducing the Cans Loss Index in a beverage company. Jovanović et al. (2017) applied PDCA approach to enhance the national policy and education on energy management in Serbia.

The PDCA cycle starts with Plan phase. Normally, a team spends their majority time and effort in this phase. In Plan phase, it started with a problem definition. Basically, different complexity level of problem or improvement opportunities will be rising at the same time in a program or process. It is not possible to investigate and solve all of it at the same time. Thus, a project selection tool called prioritization matrix helps in prioritize the project to

be selected by according to criteria and specification of the project. After a problem or improvement opportunity had been finalized, a problem statement and impact statement was prepared which revise a team along the planning process (Patel and Deshpande, 2017). It is important to convert the problem or improvement opportunities into measurable. A measurable objective can be achieved at the end of the improvement process. Next, understand the problem or improvement opportunities by collect related data on the current process flow. Two main tools that can be used to accomplish current condition understanding are flow chart and value stream mapping. After describing the current process, Pareto charts, histograms, run charts, scatter plot, and control charts are appropriate tools can be used to interpret and tabulate data on the process (Chakraborty, 2016).

After analyzing the current condition on the problem, an improvement target defined to evaluate the achievement at the end of PDCA cycle. An improvement target worked as a team encouragement and support to achieve it. It is difficult to measure the performance and run the improvement without target as direction. It is important to consider the obstacles which block the way to achieve the improvement target. These obstacles have the possibility to become the potential improvement solution. Then, identification of root cause of the problem is the key process to ensure the highest possible solution can be determined and bring to improve success. Cause and effect diagram or Fishbone diagram and the 5 Whys tools are the two main root cause identification tools. Moreover, one root cause is not sufficient to solve the problem completely. Thus, any possible root causes can be defined through brainstorming among team members. Once the root cause has been determined, carefully identified potential improvement solution and implement an action plan (Lodgaard and Aasland, 2011). Eliminate the root causes by the potential improvement solution to resist the recurrence of the problem.

Do phase started after improvement plan has been established. Do phase is about where implement the improvement plan that defined in the Plan phase. Do not confuse that this phase only involved in a trial run of the improvement plan. There may be more than one potential improvement solution can be implemented in the problem. A trial run before implementation will require to prioritize it. In Do phase, it is important to make sure that

the improvement solutions have fully caught on by the relevant person. After implementing the solution, collect and document the performance through observation. Then, evaluate the result of implementation in Check phase to verify the solutions. Compared the result with improvement target in the Plan phase to determine whether the improvement was a success and root causes have been eliminated (Patel and Deshpande, 2017). Pareto charts, histograms, run charts, scatter plots, control charts and radar charts are tools that can analyze the improvement. Restart the work at Plan phase if the improvement is not success. If not, continue to the Act phase (Chakraborty, 2016). In Act phase, a further work needed to standardize the solutions in proper way for team, relevant person and other organization. The standardization of improvement to ensure implementation can be sustained and document the work learning through all phases in the PDCA cycle (Lodgaard and Aasland, 2011).

2.2 Kaizen Concept

Kaizen concept can be combined in the PDCA cycle application. It is because the Kaizen concept has the same objective with lean which is continuous improvement and waste reduction (Maarof and Mahmud, 2016). Kaizen originated from two Japanese words “Kai” which means change and “Zen” which means good, the overall meaning of Kaizen will be making changes to become better. A common meaning of Kaizen is an improvement that involves everyone from top down and bottom up of management (Carvalho and Pereira, 2015). According to Chan and Tay (2016), the definition of Kaizen concept is the improvement of working environment and employee involvement in the company. Kaizen concept encouraged the company to involve their employee in the company improvement. Employee include operator has the chance to provide a recommendation and suggestion for the company improvement. This can strengthen the relationship between manager and employee to contribute and achieve on the company improvement together. This Kaizen concept can guarantee the improvement in the relationship and communication between manager and employee (Titu et al., 2010).

Kaizen is a long-term process. It will sustain the continuous improvement be long lasting. It is different with western company that seek for big changes that bring big result by innovate new technology and introduce new method of improvement. However, it will lead

to higher investment and long spending of time in bringing new improved technology. Kaizen concept depends on the method “small steps done many times”. This small improvement step only demands on the small expenditure of money. Kaizen is not a short-term adjustment to be done on the work flow. It becomes a process after adding on the small steps to bring the continuous improvement to the company. It should become a culture that trained for everyone in company from top level to low level of management, thus improves the flow regularly. A work flow in lean can work efficient and effectively (Ravindra and Pranay, 2016). Kaizen concept has some stages in the process of application are defined improvement area, analyze and choose the priority problem, evaluate the root cause of the problem, action planning of improvement, solution implementation, performance measure and set a standard for improvement. (Kamińska, 2015)

Kaizen is an umbrella-shaped concept that shelters different continuous improvement management techniques. It is a management philosophy that include specific principles and values in improvement management. Besides that, it has its own techniques in managing the improvement as shown in Figure 2.1 below (Sua´rez-Barraza et al., 2011).

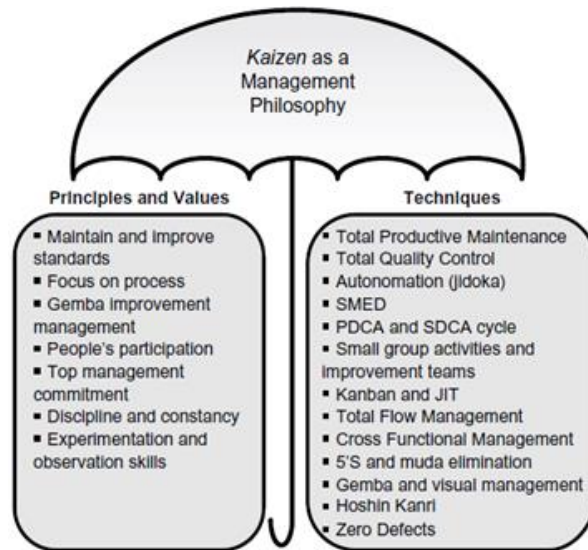


Figure 2. 1: Kaizen Umbrella-Shaped Concept

2.3 Lean Manufacturing Tool

It is important to select suitable and proper tool to implement in the project. The main objective for lean manufacturing tool is waste reduction by eliminate the non-value-added activities in the process. However, selection of lean manufacturing tool is the most

important step to avoid shifting of non-value-added activity to another process. The example of wrong selection of lean manufacturing tool is used wrong tool, use single tool and use all tools provided (Pavnaskar et al., 2003). It is quite challenging to choose the right tool during the implementation. Hence, a tool selection strategy should provide to ensure correct lean manufacturing tool selection. Furthermore, it is suggested to match the lean manufacturing tool selection with manufacturing problems (Karim and Arif - Uz - Zaman, 2013).

Dutta and Banerjee (2014) classified the right core lean manufacturing tools: cellular Manufacturing / one-piece flow production systems, 5S, just-in-time production, total productive maintenance, Kaizen, Kanban, six sigma and pre-production planning. Sundar et al. (2014) reviewed 19 lean manufacturing tool implementations: scheduling, employee perceptions, value stream mapping (VSM), takt time, line capacity, group technology, cellular manufacturing, u-line manufacturing system, line balancing, flow manufacturing, single minute exchange of die (SMED), small lot size/small batch, inventory, pull system with one-piece flow, Kanban, production levelling/Heijunka, quality at source, continuous improvement/Kaizen, and standardized work. According to B.Modi and Thakkar (2014), they recognized these tools as the key lean manufacturing tool: Kaizen, 5S, just-in-time, visual management, value stream mapping, Andon, Gemba, total productive maintenance, takt time, single minute change of die, cellular layout and supply chain management. William G. Sullivan (2002) agreed that VSM as an important tool in defining and analyzing the manufacturing waste.

2.4 Chapter Summary

To summarize this chapter, the PDCA cycle is started with “Plan” step to define the problem, root cause and determine the potential solutions. Then, “Do” step to test and implement the solution that will generate improvement to the problem. Last, “Check” the solution implementation whether it is successes or failed and “Act” to standard the improvement. The stages of Kaizen concept can be included to ensure the work more effective and efficient. The selection of lean manufacturing tool is important to have matching with manufacturing problem to guarantee the correct lean manufacturing tool on correct manufacturing problem solving.

CHAPTER 3: METHODOLOGY

3.1 Chapter Overview

This chapter described the development approach used in this project. It is presented how this thesis is conducted with the mentioned methods and approach in more details.

3.2 Development Approach

The development approach in this project was the iterative framework. According to this development approach, a project begins with breaking down the steps into few parts which the coding development, design and testing in repeated cycles. The reason for employing the iterative development approach was that it allows the additional features can be designed, developed and tested from previous running and feedback until a complete functional software application ready to be introduced to users (Nugroho et al., 2017). The iterative development approach in this project will be started with the software development tool selection, exploration, guideline collection, user interface design, coding, testing, implementation, result analysis and future development. The repeated processes were user interface design, coding and testing.

3.2.1 Android Software Development Tool Selection

The selection of tools to be used in the application development process is important to either succeed or break a project. In the beginning of this project was to choose the suitable tools that will be utilize in this Android software development project. Before choosing the tools, the requirements should be well-defined and understood to ensure suitable tool selection. In this project, the main requirement of tools selection in this project was non-programming platform to develop the Android application easily. A non-programming development tool is a program that used a graphical user interface by drag-and-drop graphical objects to produce a software application. It is required to take integration factor into the consideration in the selection of tools to assist the development process.

A selection of non-programming Android software development tools has been done on AppsGeyser, AppMakr, Shoutem, and MIT App Inventor. In AppsGeyser, it is requested

to choose the software template in starting the development process. Unfortunately, there are no suitable templates matched with this project (Refer to Figure 3.1). Next, for AppMakr (Refer to Figure 3.2), Shoutem (Refer to Figure 3.3) and MIT App Inventor (Refer to Figure 3.4 a), they have the graphical interface designer which used to design the software's interface by insert component provided. The examples of components that allow to insert are video, audio, camera, website, and other functions. However, MIT App Inventor has another unique function which is visual block programming editor. It applied the block program to establish a connector with the component inserted under certain condition to specify software's behavior (Refer to Figure 3.4b).

The consolidation between a tool and other tools can extremely impact the value appended to the Android software and the project. In this project, the integration between interface design and programming will be selected as the primary element in Android software development tool selection. Among the tools chosen, only MIT App Inventor meets the integration factor to integrate the graphical interface designer and block programming editor. This integration merges the interface design and block program to deliver high level of software development. Hence, the finalized Android software development tool to be applied was the MIT App Inventor.

CREATE APP

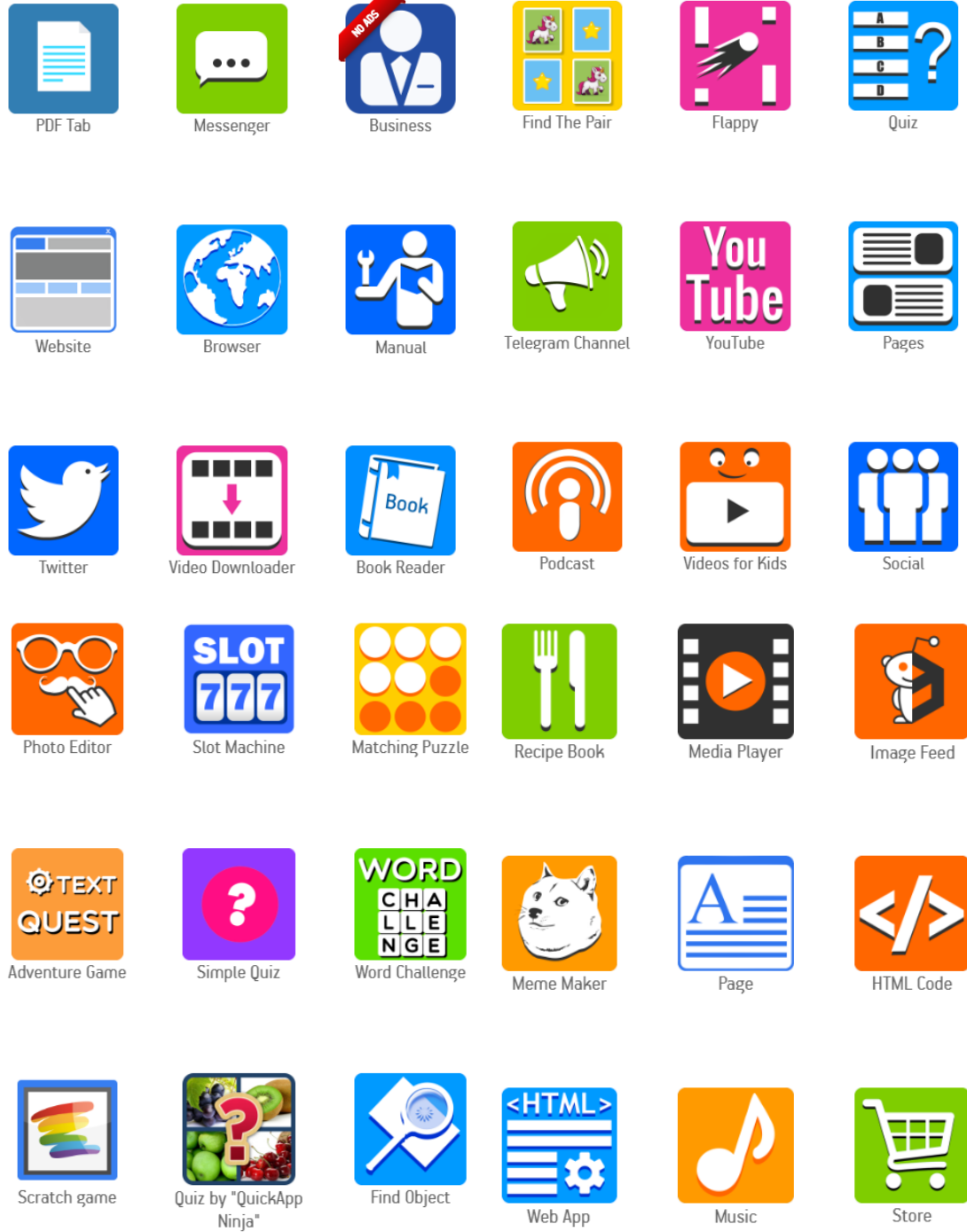


Figure 3. 1:Example of Software Templates Provided in AppsGeyser

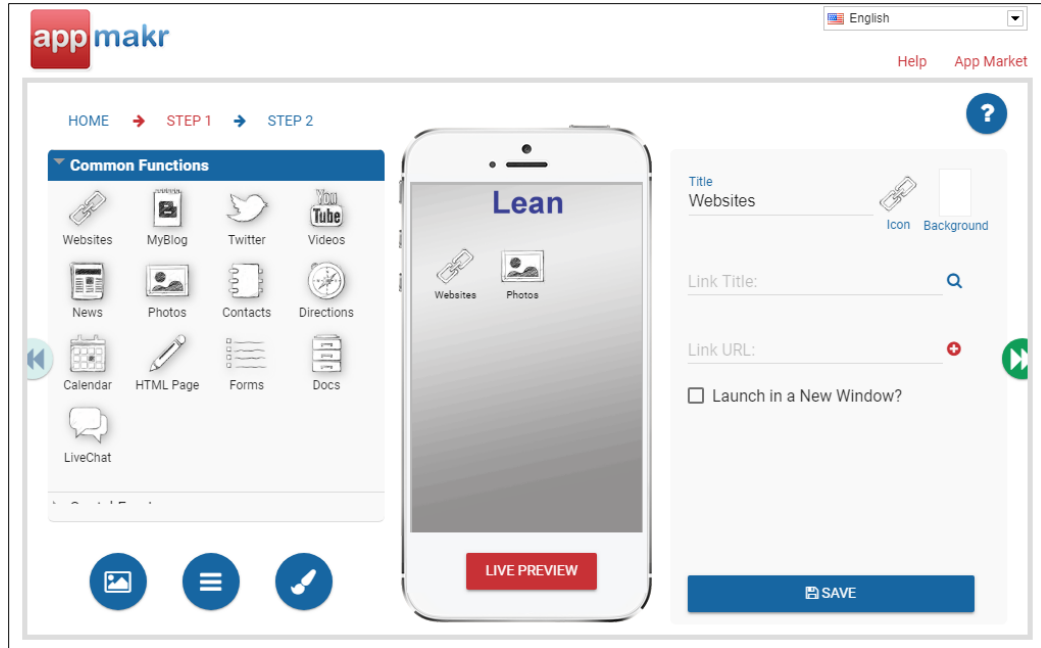


Figure 3. 2: Graphical Interface Designer Screen View of AppMakr

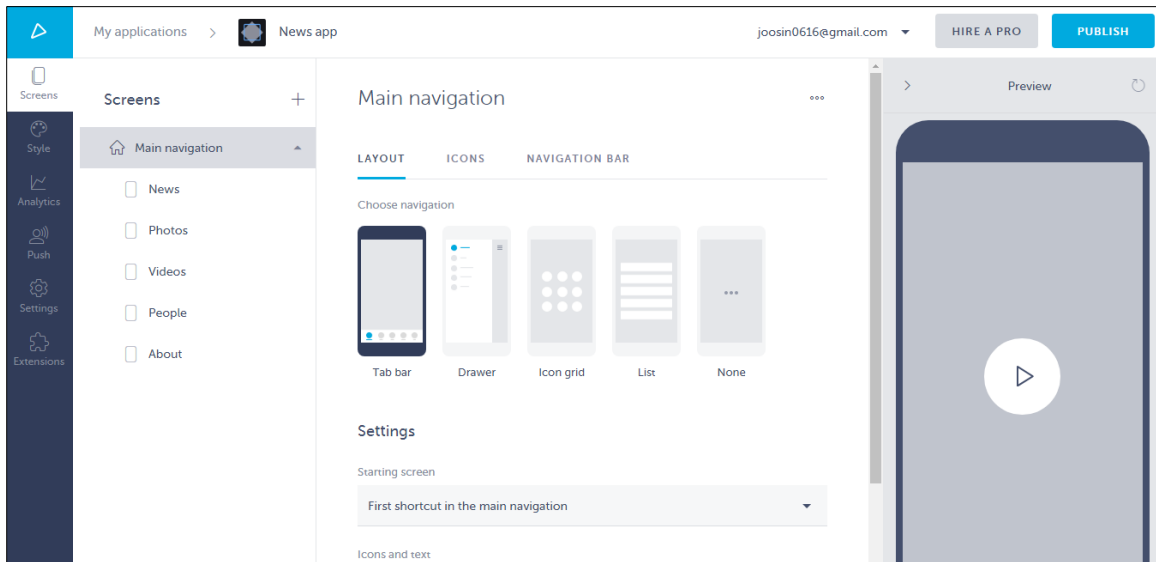
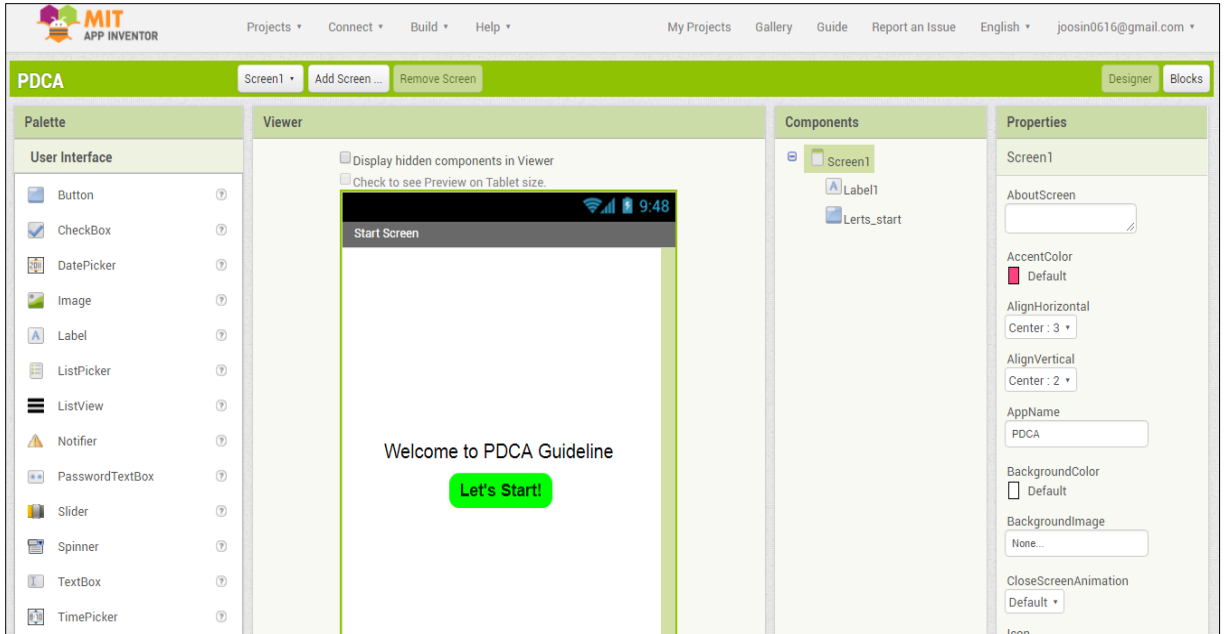
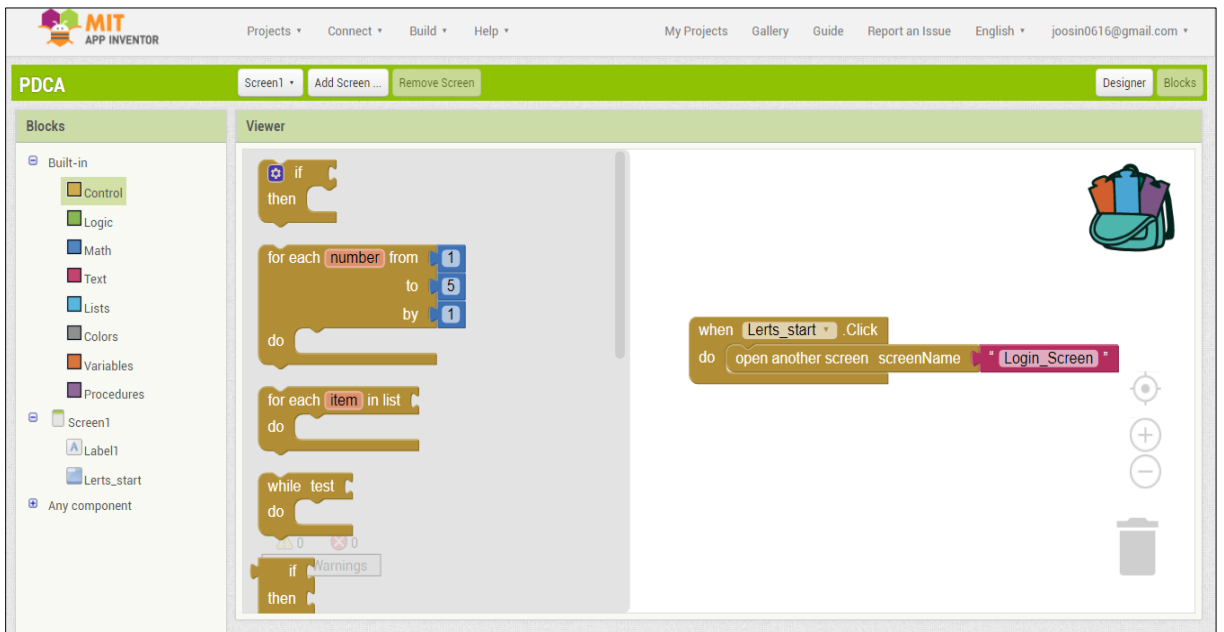


Figure 3. 3: Graphical Interface Designer Screen View of Shoutem



(a)



(b)

Figure 3. 4: Graphical Interface Designer and Block Editor of MIT App Inventor

(a) Graphical interface designer

(b) Block programming editor screen view of MIT App Inventor

3.2.2 Exploration on MIT App Inventor

MIT App Inventor is a block-based programming application which is developed by Google. It is currently developed by the Massachusetts Institute of Technology (Tadeusz Mikolajczyk, 2017). It does not request in programming code writing since it is based on drag and drop programming for graphical interface designer and block editor. Hence, it enables inexperienced users which lack of programming knowledge to create Android application for mobile devices. The drag and drop programming of visual blocks display the connection easier to understand by new learners. This web interface is available worldwide, with 3 million users from 195 countries that have created a total of 7 million Android applications. MIT App Inventor's access to plenty of fields, including education, design, research, government and enterprise (Lisa L Ruan, 2017). This block language decreases the typing error by users and provide easy understand with visual block design (Hwansoo Kang, 2015).

It begins with a visit to the MIT App Inventor website (<http://appinventor.mit.edu/explore/#>). Then click the “Create App!” at the top right corner as pictured in the Figure 3.5. The application will demand for authorization with a Google account.

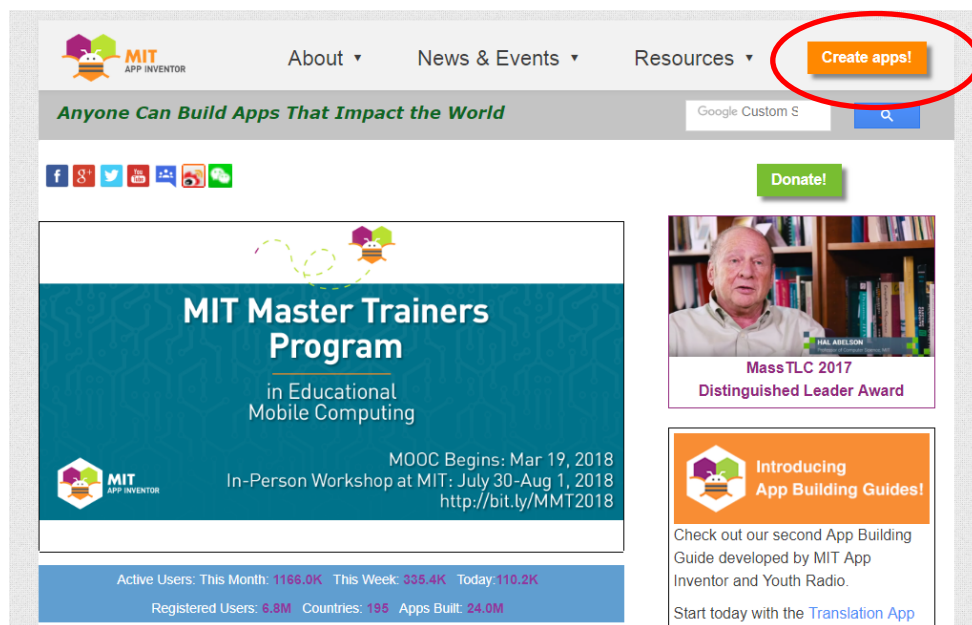


Figure 3. 5: MIT App Inventor Website

After logging on with own google account, take up a new project by keying in the new project name and confirm with OK. The main panel of application which is graphical interface designer screen will be shown as Figure 3.6 below. There are a few sections on the designer screen and the explanation of each section is provided by following the sequence number:

- *Control Button (1)* – Bar to add/delete screen, open designer component/block editor.
- *Palette (2)* – Toolbox to choose components and then drag them to the preview screen. It is split up into different types of components: user interface, layout, media, drawing and animation, maps, sensors, social, computer memory, connectivity, LEGO MINDSTORMS, experimental and extension.
- *Viewer (3)* – Application preview screen on mock phone provided
- *Components (4)* – Listed components used by rename and delete the components.
- *Media (5)* – Media uploading from computer include file, picture, video and music.
- *Properties (6)* – Set the properties of components in color, size, text, alignment, height, width and other behavior of the applications.

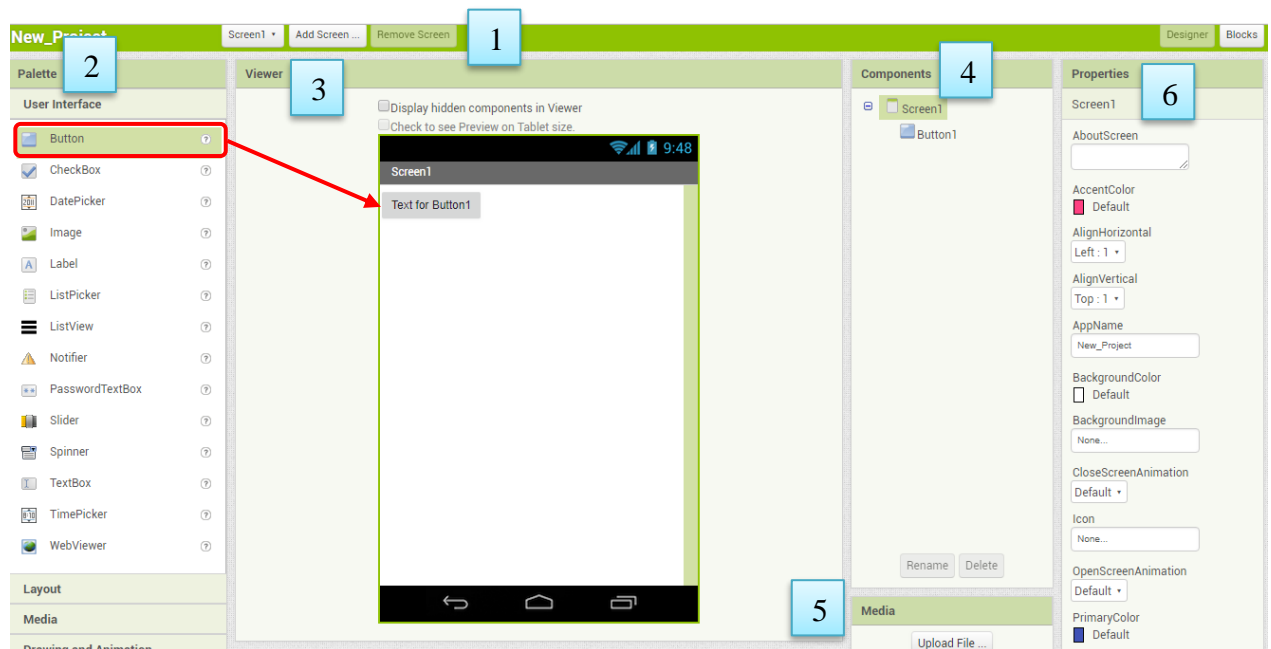


Figure 3. 6: Main panel of MIT App Inventor (Graphical interface designer)

In the designer screen, users drag the components and drop it on the viewer. The components consist of visible and non-visible as shown in Figure 3.7. If it is a visible component like Button, Checkbox, Image, Label, etc., which are sections of User Interface, the components are displayed on the mock phone screen. On the other hand, if it is a non-visible component like camera, accelerator sensor, fusion table control, etc., the components are displayed below the mock phone screen. The special properties of components can be customized in the properties section. The components access their built-in function with block program associated with their specific behavior and properties by switching from designer screen to block editor screen.

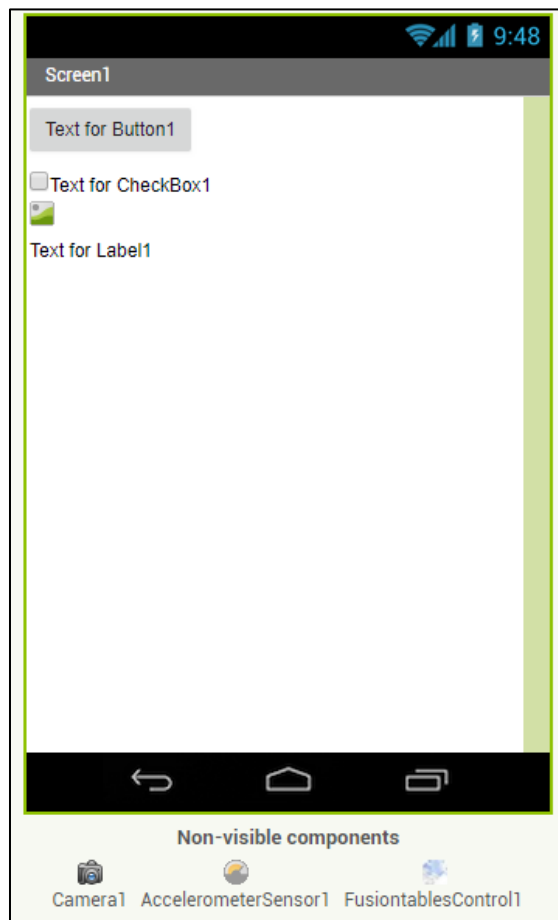


Figure 3. 7: Visible and Non-Visible Components

To switch from designer screen to block editor screen, clicked the “Blocks” button at the top right of the control button bar. The main panel of block editor will be shown as Figure 3.8 below. There are a few sections in the block editor screen and the explanation of each section is provided by following the sequence alphabets:

- *Built-in Drawer (a)* – Get the general built-in block behavior. These blocks are always available. They handle behavior like control, logic, math, text, lists, colors, variables and procedures.
- *Component Specific Drawer (b)* – Get the specific block behavior of components. It is only available for components chosen at the graphical interface designer.
- *Viewer (c)* – Preview of block built behavior and relationship. It is the workplace of block program.
- *Warning area (d)* – Yellow warning and red error alert. Yellow warning means that un-filled blocks and red error means wrong program building.
- *Backpack (e)* – A copy-paste feature works between projects and screens. It can copy and paste the blocks to other projects and screens.
- *Trash (f)* – Delete unnecessary blocks.

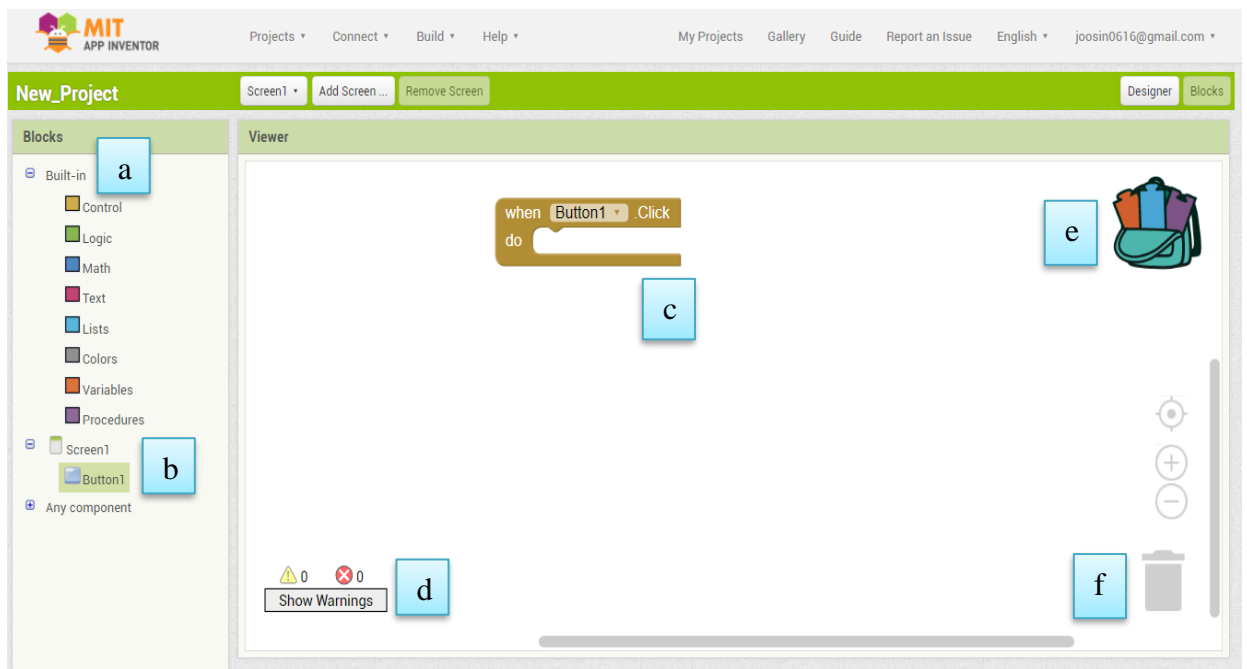


Figure 3. 8: Main panel of Block Editor MIT App Inventor

Just like designer screen, in the block editor, users drag the block and drop it on the viewer. The block display is the puzzle-looking pieces that connect together to perform in programmed behavior and situation. There are two primary types of blocks which act in different ways: event handler and commands.

Event Handler Blocks

The event handler blocks programmed on how the software respond to an event: when a button is pressed, when the camera takes pictures, when the screen is initialized, when fusion table control got results, etc. A word “when” will appear on the top left of the puzzle pieces in light brown color. This event handler block only can get from Component Specific Drawer under certain components. Some models of event handler blocks are presented in Figure 3.9 below.

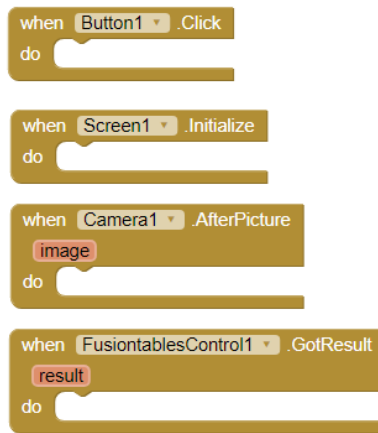


Figure 3. 9: Event Handler Blocks

Commands Blocks

After an event handler started, a sequence of commands required to create the event and action that perform in the software. There are two types of command blocks which are setter and getter block as indicated in Figure 3.10. However, it must use a setter command block with the word “set” at the left of the blocks to connect with event handler blocks. It is because only the puzzle piece of setter command block can be matched with event handler blocks. After the setter blocks, connect to getter block to completely specify the actions. The setter block cannot be associated with event handler blocks directly because

the puzzle parts are not matching. The puzzle pieces matching of event handles and control blocks are presented in Figure 3.11.

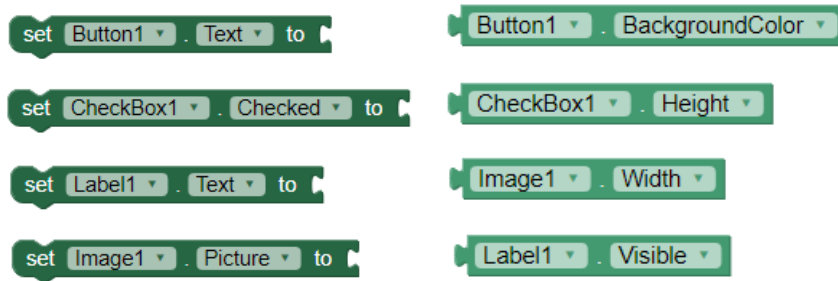










Figure 3. 10: Setter (Left) and Getter (Right) Commands Blocks



Figure 3. 11: Puzzle Piece Matching

The different block types are different colors. This allowed the users to differentiate the behavior and function blocks. More command blocks can generate from the built-in drawer. The overview of built-in blocks is shown in Table 3.1 below.

Table 3. 1: Overview of Built-In Blocks

 Control	Control the actions of software under given situations.
 Logic	A logical condition performs to test whether it is true or false
 Math	A mathematical equation and number can be used in block program.
 Text	A text control operation includes type, join, empty, trim, etc.
 Lists	A list control operation includes making a list, pick an item from the list, and so on
 Colors	Basic color available to choose.
 Variables	Create the variable to be practiced in the event and procedure.
 Procedures	Create a new procedure or function and call it back to avoid long sequence in one block.

Next, it can begin to try out the applications in real time on the mobile device by establishing the MIT App Inventor Companion application from Play Store as shown in Figure 3.12.



Figure 3. 12: Icon of MIT App Inventor Companion

Then, connect the MIT App Inventor browser with Artificial Intelligence (AI) Companion as shown in Figure 3.13. A few seconds will be needed for Quick Response (QR) code building to appear on the computer screen. MIT App Inventor Companion on the mobile device is opened and scanned the QR code/typed 6-character code in Figure 3.14. After a few seconds, the applications will build on the mobile device. Any update on computer will also change on a mobile device at the same time.

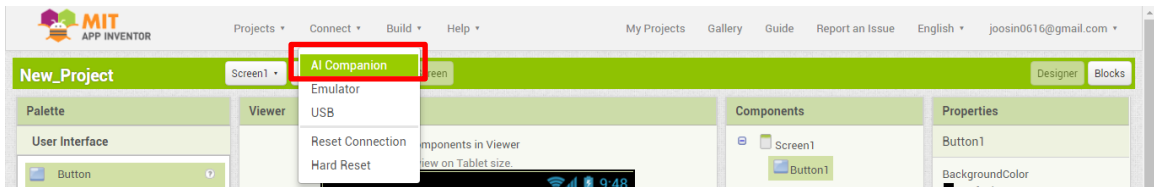


Figure 3. 13: Choose AI Companion to Connect

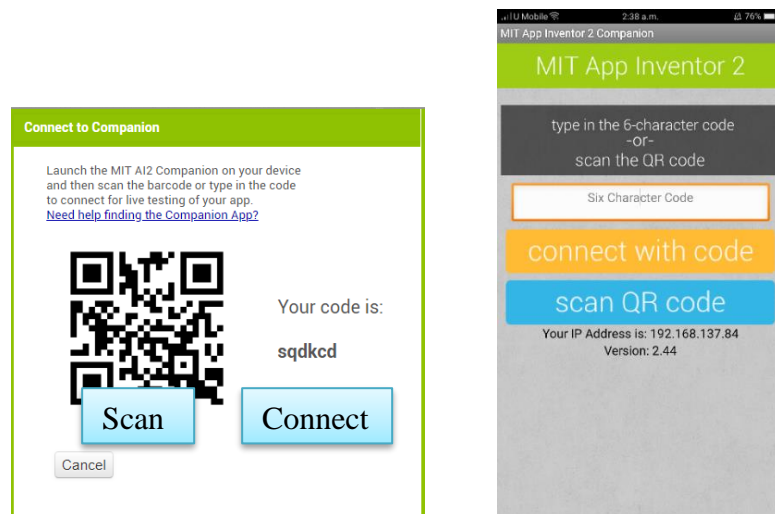


Figure 3. 14: Connect and Scan with QR Code

3.2.3 PDCA Guideline Collection

There is a total of the twelve steps in PDCA cycle and data collection of twelve steps was done. Before starting a project, it was required to form a project team to support the project. The construction of the twelve steps of PDCA cycle is Step 1 to 5 in the Plan phase, Step 6 to 8 in the Do phase, Step 9 in the Check phase and Step 10 to 12 in the Act phase. The guideline collected, then transfer to the design of the application.

3.2.3.1 Project Team Formation

A typical project team is comprised of more than 4 members who involve in the project. The team members stand for people who perform one or more in the process, supply something in the process and gain something from the process. In Table 3.2, it is shown that the role and responsibility of team members. The team members on the PDCA project include team leader, champion, mentor and normal team member.

Table 3. 2: Role and Responsibility of Team Member

Team Role	Responsibility
Team Leader	<ul style="list-style-type: none">• Act as the Chief Operating Officer of the project.• The responsibility is to lead and guide the team toward improvement.
Champion	<ul style="list-style-type: none">• A process owner who owns the solution delivered by the team.• The responsibility is to provide resources and support the implementation of the solution.
Mentor	<ul style="list-style-type: none">• A supervisor who assists the team leader• The responsibility is to provide direction when necessary and manage meeting for a progress update.
Team Member	<ul style="list-style-type: none">• A group of people who assist the team leader in achieving the project goal and improvement• The responsibility is to provide expertise and perform implementation of the solution.