

	Page
TABLE OF CONTENTS	i
LIST OF FIGURES	iii
CERTIFICATION DECLARATION	1
ABSTRACT	2
ACKNOWLEDGEMENTS	3
CHAPTER 1: INTRODUCTION	
1.1 Background	
1.2 Problem Statement	
1.3 Objective	4
1.4 Scope of Study	
1.5 Relevancy Study	
1.6 Feasibility Study	
CHAPTER 2: LITERATURE REVIEW	
2.1 Report Recommends Shift to Digital Educational Resources Within 5 Years	
2.2 Of Algorithm & Human Writers	8
2.3 How to build a university mobile application: best practice and insight	
2.4 Summary	
CHAPTER 3: METHODOLOGY	
3.1 Iterative and Incremental Development	
3.1.1 Planning	
3.1.2 Requirement	
3.1.3 Analysis & Design	
3.1.4 Implementation	
3.1.5 Testing & Evaluation	
3.1.6 Deployment	
3.2 Development, Testing & Evaluation Tools	
3.2.1 Development Tools	
3.2.2 Testing & Evaluation Tools	
3.3 Project Activities	

3.3.1 Project Activities for FYP 1	
3.3.2 Project Activities for FYP 2	
3.4 Development Progress	
3.4.1 Current Progress	
3.4.2 Future Progress	
3.4.3 Identified Problem	
3.4.4 Solution	
CHAPTER 4: RESULT & DISCUSSION	24
CHAPTER 5: CONCLUSION AND RECOMMENDATION	28
REFERENCES	30
APPENDICES	31

TABLE OF FIGURES

	Page
Figure 1 : Android SDK	16
Figure 2: Samsung Galaxy Y S5360	17
Figure 3: Asus Nexus 7	17
Figure 4: FYP 1 Gantt chart	18
Figure 5: FYP 2 Gantt chart	18
Figure 6: Internship Time Tracker Main Page for User Input	18
Figure 7: Internship Time Tracker Main Page choosing date	19
Figure 8: Internship Time Tracker Output Page	20
Figure 9: Internship Time Tracker Output Page Picking Date	21
Figure 10: Internship Time Tracker Output Page Showing Note Detail	22
Figure 11: Internship Time Tracker Output Page Delete Confirmation	23
Figure 12: Survey Question Number One	24
Figure 13: Survey Question Number Two	25
Figure 14: Survey Question Number Three	26
Figure 15: Survey Question Number Four	27

CERTIFICATION DECLARATION

I hereby certify that this report and the project it describes is my original work and is not taken from the work of others save and to the extent that such work has been cited and acknowledged in the report.

Student:

Signed: _____ Student No: _____ Date: _____

Supervisor:

Signed: _____ Date: _____

ABSTRACT

This project focuses on an android application of writing a digital logbook. The sole purpose of this project is to assist industrial internship students in doing their internship daily logbook report. Students tend to neglect their daily logbook report as the logbook is a hassle to bring along to office especially if the task involves travelling outside of office.

The main objective of this application will be to focus on mobility for students to do their daily logbook. It comes with a simple interface which can be easily use and understand without any proper training or tutorial. By recreating the logbook into an android application, students can deal with their daily report while waiting for bus when going to work just by using their smartphone. At the end of the day, students just need to show or transfer the created logbook to their supervisor for approval. The simplicity of the application will relieve students from the burden of their logbook report.

This project is going to take about 4-6 months of development time and I am going to use Iterative and Incremental Development to develop this application.

ACKNOWLEDGEMENTS

Alhamdulillah, all the praise and gratitude is only for Allah, by His blessing and guidance throughout these two (2) semesters enabling me to complete my final year project successfully.

I would like to express my special gratitude to University Technology PETRONAS for providing a chance to conduct my final year project for two (2) semesters in partial fulfilment for Bachelor of Information and Communication Technology. This project gives very meaningful experiences and lesson to me to be exposed on the real working environment as my preparation after graduation.

My abundant thanks to my supervisor, Dr Lukman Bin Ab Rahim, who give strong commitment to guide and support me all the way through this challenging project period. He always teaches me the ethics of working and the knowledge in handling every single thing at for my project.

Last but not least, thanks to everyone involved in the completion of my final year project, whether directly or indirectly.

CHAPTER 1 INTRODUCTION

1.1 Background

Logbook report is one of the major components during student's industrial internship. It is essential for students to log every activity done during internship so supervisor can track what they did during work hours. At the end of the day, company supervisor will go through the logbook and put a signature as sign of approval and validation of the logged activities done by the students.

Students will need to log their activities daily until the end of their industrial internship period. They must write in details of all activity done so that later on respective university supervisor can track and know what their students did and whether they learn new things or not. University's supervisor will set a date for a visit where company supervisor, university supervisor and students will sit together and go through student's logbook. At this point, logbook is very crucial to reflect student's action and activities in the company. A bad logbook will give negative impact on their mark and at the same time might give bad impression about the company in the eyes of university supervisor. Thus, it is important for students to be equipped with easiest and comfortable method in doing their logbook as this can ease out many party including both supervisor and university industrial internship unit.

This research will focuses on an android application which can help students in writing their logbook. It will provide usability, which is an easy to use and simple interface. Besides that, this application will provide mobility as students will use it with their phone instead of bringing their logbook around and I can also help students and university supervisor to communicate and discuss about logbook easier.

1.2 Problem Statement

Students tend to neglect their daily logbook report as the logbook is a hassle to bring along to office especially if the task involves travelling outside of office. Some students do not like to bring their logbook to work with them as they afraid they might leave it at the office. When they get back from work, they will face difficulty in doing their report because they can't recall what activities has been done during the day. Because of this, they write based on what they remember and this will cause bad logbook writing.

Some students do not want to bring along their logbook as they think of it as a hassle. This is especially when students need to do a lot of outdoor works such as visiting site and tower maintenance. Afraid of misplace the logbook; most students decide to keep it at home and do the report later. This also causes the students to face difficulty when writing their report later on which produce bad logbook writing.

1.3 Objective

The objectives of this project are:

- ❖ To design a mobile application for students to create report.
- ❖ To provide mobility for students in creating daily report.
- ❖ To replace conventional way of report writing with digitalized report.

1.4 Scope of Study

The general aim of this study is to identify problems face by industrial internship students in creating daily report and design a mobile application to tackle those problems. The main purpose of the study is to create a mobile application which reflects the same component as in the logbook report. Besides that, the application will also need a way for company supervisor to create a digital approval which validates students report. But for the time being, company supervisor can use a marking sheet provided by university to validate and mark students' performance.

Due to the big size of original logbook report, it is crucial to find the most efficient ways in recreating the component inside mobile application. This project will mainly use Iterative and Incremental Development method. A mobile application development tool, Sencha, will be used for development and the testing will be done by developer and some internal beta tester.

1.5 Relevancy Study

The relevancy of this project is on the problem statement where it is an existing problem among industrial internship students. For many years, logbook report has been a very stressful problem where it is hard to keep track and remember of tasks that we have done for the day and write a report about it at the end of the day. Besides that, there is also a case where a batch does not receive a logbook during industrial internship which causes problems because they do not have any medium to do the logbook report. At last, variety of logbook format were created and causes problems when university supervisor come for visit and collecting logbook report

By developing a mobile-based application, it helps not only students but also industrial internship unit where they do not need to rush on giving logbook report to students as students can use the application to write their report. Besides that, any problem regarding logbook such as late logbook distribution can easily be solved because all the report is done in the phone.

1.6 Feasibility Study

I have done some feasibility study for this project. I have reflected it in from of Strength, Weakness, Opportunity, and Threat (SWOT):

<p>Strength</p> <ul style="list-style-type: none"> ▪ Develop on the popular platform that is Android OS. ▪ Easy to approach students by publishing application in Google Play. ▪ One of a kind in the market. 	<p>Opportunity</p> <ul style="list-style-type: none"> ▪ Focus on Industrial Internship students where all universities applied Industrial Internship ▪ Application can be expand to other platform if successful.
<p>Weakness</p> <ul style="list-style-type: none"> ▪ Cannot be used by other platform ex. Windows Phone, iOS. ▪ User need to be familiar with Touch Keypad. 	<p>Threat</p> <ul style="list-style-type: none"> ▪ Android is open source thus causing lot of potential competitors. ▪ University rejection of digitalize report

Table 1: SWOT Analysis

CHAPTER 2 LITERATURE REVIEW

Logbook has been used for ages. It is a method of recording activities which can be used to track back what has been done daily, weekly, monthly or even annually. Logbook provides an important role for ages, example, and a ship that create logbook which reflect their travel route and activities from day to day. This logbook later then might be read by important person to track the progress of the ship during its sailing day. Thus, logbook must be created effectively and efficiently so that people can read and understand what written in it.

2.1 Report Recommends Shift to Digital Educational Resources within 5 Years

Of 2012, most students already equipped themselves with today's technologies such as tablet and smartphones. From this, we can see an opportunity to shift from classic logbook report to digital report. Breakthrough of technologies nowadays makes this possible even on small size smartphones. According to some studies, it is possible that report writing will be shift to digital in five years time.

(Leila Meyer, 2012) A new report from the State Education Technology Directors Association (SETDA) points to the importance of shifting K-12 schools in the United States from printed textbooks to digital educational resources. This proves that digital technologies will be exposed to kids since school which makes digital report is a suitable approach to replace current method of report writing. Besides that, there are also a lot of other factor that show digital report might be beneficial to all parties. For example, the number of students equipped with smartphones and tablet are very high which make it easy to implement digital report writing in university. Furthermore, by using digitalization of data, it is easier to update and track progress through smartphones.

Digitalization of information is the future of replacing current report writing. With the right method and push, we might live in a world of digital, where information is at the end of our finger. With technology all around us, there are a lot that we can

achieve with digitalization of information. From every aspect, there will be a lot of benefit from digitalization such as saving more space and man power to sort and keep every report in a company.

2.2 Of Algorithm & Human Writers

In today's world, algorithm is widely used especially in computer related such as Artificial Intelligence. For example, Cleverbot, where user can chat with a bot. Cleverbot use Artificial Intelligence where it learns from people in context and imitation. Thus, it can answer any question that user ask and can reply to any user input consistently. Even though it is just a bot created using algorithm, the answer provided by the cleverbot is amazing that we might think it is not a bot.

In an article written by Gini Dietrich, she discuss on the reliability of algorithm to replace human writers. In her article, she shows the advantages and disadvantages of algorithm in writing a story or report. One of the most interesting disadvantages is that an algorithm cannot create nuances and subtle touches that are very important in story and writing report. But, there is also advantages when using algorithm such as algorithm can manage certain things like keyword better than human.

There is also a discussion between the efficiency of using algorithm or human writers in creating story and report. For example, both human writers and algorithm might create the same story or report but the art of language between is the question here. Human writers can write more artistic story or reports than algorithm. We cannot teach art to computer where art is very important in writing.

But still, it might be inevitable to prevent algorithm from replacing human. As we can see around us, as technology advances, many human-based machine and tasks were replaced by algorithm such as big robot and computer. It does not mean that we won't need human beings to do such thing anymore but it will limit the involvement of human.

2.3 How to build a university mobile application: best practice and insight

Nowadays, wireless internet connection is a common thing in universities around the world. Wi-Fi technology gives a very huge benefit especially for those with mobile gadget such as smartphones. Developing a mobile application is very challenging especially when the scope of user is student. Besides that, there are also other hindrances such as strategy of development to focus on.

In an article by Kerry Eustice on How to build a university mobile application: best practice and insight, he shows some example of mobile application development in university. According to the article, the best way to create a new application in a university is to create an application based on the existing medium such as creating a mobile version of university website based on their existing website. This is the quickest and relatively low cost to develop an application.

Besides that, his article also stated that a good idea needs to fuse with usability, design and marketing. In another words, the application that going to be develop must be user friendly especially to students, a design and process that satisfy the university and an application that can be market outside the university. Furthermore, it is advised to consider the user experience in the delivery of new application. Dr. David Currie, a database manager, development and alumni relations office at Aberystwyth University also stated that the important thing during early development is to identify the most relevant platform for the initial release and focus on it.

Summary

All literature above has helped me a lot for my application development. Each article contains very beneficial information that helps me to in deciding the behavior of my application. For example, the first article shows me that digitalization of education materials is starting and it is going to be achieved in near future. This prove that my

application will be beneficial in creating digitalize report when all education materials are digitalize.

In the second article, it helps me to decide how my application will work. At initial planning, I decided that my application should automatically generate report based on data input by user. But after reading and studying the article, it shows that with current technology, algorithm is still not suitable to be used for automatic report writing and human writer is more powerful than algorithm. Thus, I decided that my application will need user itself to write their report.

Lastly, the third article, give a lot of advice on how to create an application for university and students. It teaches me on what to focus on during early development in order to create a good application. Besides that, it also give me an idea on how to develop my application with lesser times and cost.

CHAPTER 3 METHODOLOGY

3.1 Iterative and Incremental Development

This project adopts Iterative and Incremental method. It involved iterative development and construction of prototypes. By adopting this method, this project will be segmented into several phases which are Planning, Requirements, Analysis & Design, Implementation, Testing, Evaluation and Deployment. Here is the action that is done and will be done based on Iterative and Incremental Development

3.1.1 Planning

Planning phase has been done after the proposal of project. Discussion has been done with my supervisor, Dr. Lukman, regarding the features and identified issues such as securities, complexity, process flow and others that need to be attended. Besides that, I also did a study to find out the best platform for the application to be developed. Then, I did some research on development tools to be used to develop the application.

3.1.2 Requirement

Requirement phase is to find and study specific details to be implemented to the application such as Interface, Function and Performance. Below are the details for each requirement:

- Interface Requirement – Application will require three interfaces where each interface will has its own input or output. The first interface will be for input where user will key in all the details required to create a report such as student's position and task. Then student can start writing their report. Second interface will be used to display a calendar where user can pick a date to view any report written on that date. If the chosen date contains a

report, 3rd interface will pop-up and display the output of that is stored on that date.

- **Functions Requirement** – Show how the application will handle input/output data. Application will handle data by receiving input and store it in lite database inside Android. For output display, application will only grab required information from database and display it on the right interface.
- **Performance Requirement** – Show what the acceptable performance for the application is. In this case, application must be able to run on most today's smartphones without any performance issues such as system hang. Besides that, application can be run in the background without using much processing power to the extent that user didn't notice it is running on the background

3.1.3 Analysis & Design

Proper analysis will be needed to identify the specification of the behavior and the solution to be built. In this case, I must analysis any potential threat that the application might encounter such as memory leak where it will affect user's phone. Result from the analysis will be converted into detailed design and it will be implemented into the application. For the design, it consists of 2 interfaces that are Report Writing, View Report. Report Writing interface is where user will key in all required data to create a report. While View Report interface is where user can view all report that they have created.

1.4 Implementation

During this phase, developer will focus on development of the core function of the application. Prototype will be release to internal tester which is the developer himself to weed out bug. At the same time, development will still continue to polish and enhance the prototype. Besides that, bug fixing will also start. Thus, every new prototype released will be included with enhanced & new feature, bug fixed or both.

3.1.5 Testing & Evaluation

Testing and evaluation will start in parallel when the first prototype is build. The application will be test and evaluate on three main components which are performance, stability and usability. The application will be tested on Samsung Galaxy Y and Asus Nexus 7. For performance, I need to ensure that the application must meet the Performance Requirement. For stability, I must ensure that the application is stable to be used without any functionality breaking bug. Cosmetic issue will be put aside until as the priority will be on stability. For usability, I must ensure that the application is easy to use and it will not cause confusion for user. When the prototype is stable enough, it is distribute to selected user for further testing and evaluation.

3.1.6 Deployment

At one point of the development, testing and evaluation, a stable prototype will be marked as Release Candidate (RC). A newer prototype with enhanced feature and more stable will be marked as RC where the previous RC prototype will obsolete. RC prototype means that it will be the one that is going to be delivered on the dateline. But, the prototype will continue to be enhanced and added with more features to achieve optimum level of performance and quality.

3.2 Development, Testing & Evaluation Tools

Tools are very essential in every software development. To ensure smooth development, testing and evaluation, I did some research to find suitable tools that can provide more benefits and higher efficiency for each phase. Below are the tools for those phases:

3.2.1 Development Tools

After some research, I have decided to use Android as the platform for the application. This is because most of the students are using Android-based phone. Besides that, there is a lot of Android Developer Community where I can get help and guide to develop my application. For the development tools, I have decided to use Android SDK. Android SDK is a development tool provided by Google to develop Android application. Android SDK has its own community-driven forum where one can ask for help and guidance.



Figure 1 Android SDK

Development workstation is very important when doing Android development as most of Android development tools and emulator demand lot of processing power. After some research, I decided on a machine that is sufficed for the development of my application:

CPU:	AMD Phenom II x4 945 3.0MHZ
Memory:	6GB DDR3 RAM
HDD:	500GB
GPU:	ATi Radeon HD6870 Toxic 1GB DDR5

3.2.2 Testing & Evaluation

Testing & evaluation is very important in order to produce the most stable and high quality application. For this application, testing is going to be done in two phase. The first one will be in Android Emulator that is going to be run through the workstation and second test will be on actual Android-Based smartphone. Smartphone that will be used in this test is Samsung Galaxy Y S5360. Below is the specification for Galaxy Y S5360:

Screen Size: 240 x 320 pixels, 3.27 inches
Memory: 256MB
CPU: 830MHz ARMv6



Figure 2: Samsung Galaxy Y S5360

Besides that, testing will also be done on Asus Nexus 7 tablet. Below is the specification for Asus Nexus 7:

Screen Size: 800 x 1280 pixels, 7 inches
Memory: 1024MB
CPU: Quad-core 1.2 GHz Cortex-A9



Figure 3: Asus Nexus 7

3.3 Project Activities

3.3.1 Project Activities for FYP 1

Activities\Week	1	2	3	4	5	6	7	8	9	10	11	12	13
Finding Supervisor	X												
Research on propose title		X	X										
Propose final chosen title			X										
Project Analysis				X	X	X	X						
Planning System Design							X	X	X	X			
Project Testing										X	X		
Market Survey							X	X	X	X	X		
Analyze Data												X	X

Figure 4: FYP 1 Gantt chart

3.3.2 Project Activities for FYP 2

Activities\Week	1	2	3	4	5	6	7	8	9	10	11	12	13
UI Design	X	X											
System Design		X	X										
UI Development			X	X	X								
System Development				X	X	X	X						
Testing						X	X	X	X				
Bug Fixing								X	X	X	X		
Implementation											X	X	
Delivery												X	X

Figure 5: FYP 2 Gantt chart

3.4 Application Interface

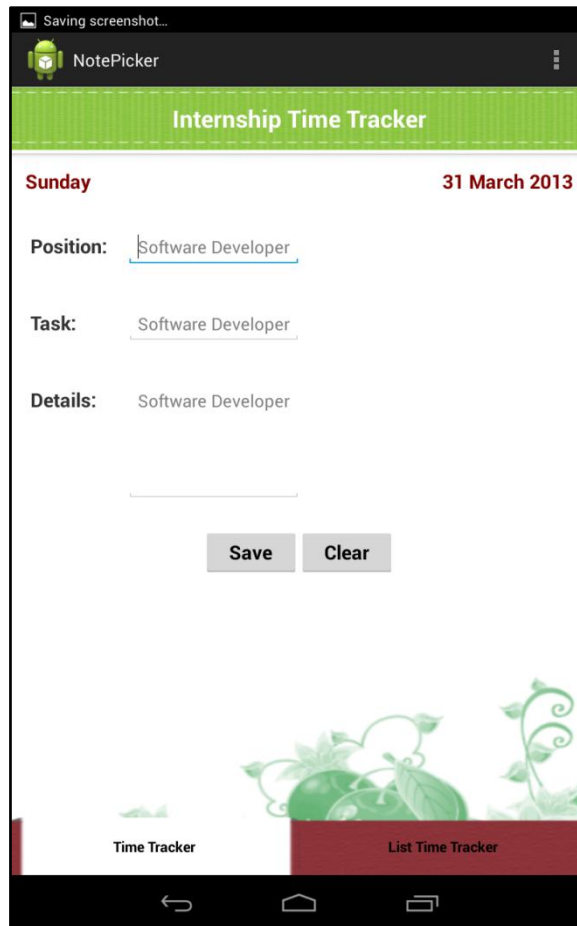


Figure 6: Time Treadmill Main Page for User Input

This is the main interface for Internship Time Tracker. User will need to key in their input in the provided space. Below is the guideline for each input:

Position: Position of students in the company. Example, software tester, software developer, etc. The reason students need to key in every time to do a report is because student might hold different position throughout their internship period.

Task: Title of the task for the activity. For example, Bug Testing, Code Maintenance, Test Automation Script, etc.

Details: A brief details regarding the task. User can write as long as they want.

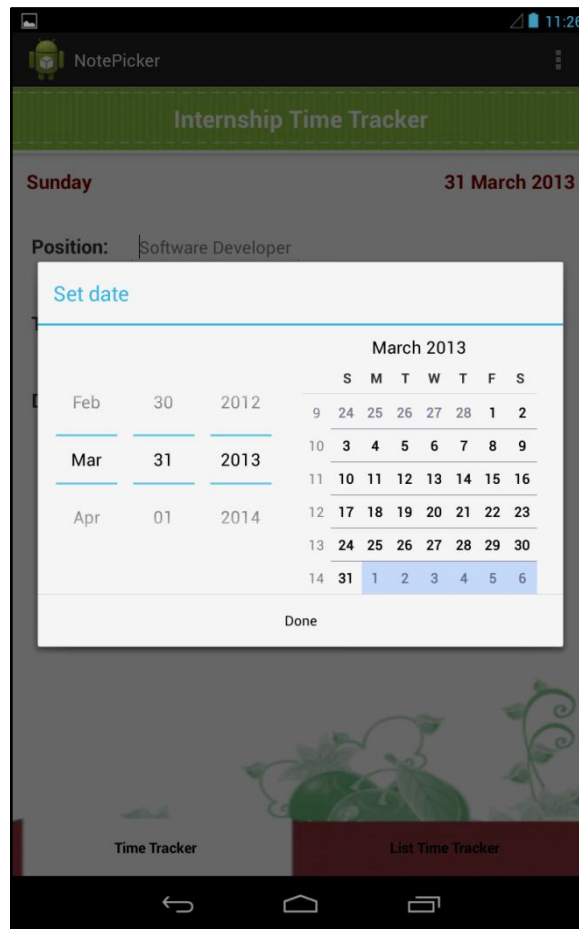


Figure 7: Time Treadmill Main Page choosing date

On the top right of the application is the date. Default date will be set to current date but user still can choose which date they want to write a note. By clicking on the date, the application will open an overlay interface which shows date picker on the left and calendar on the right. User then can choose which date that they want from either date picker or calendar.

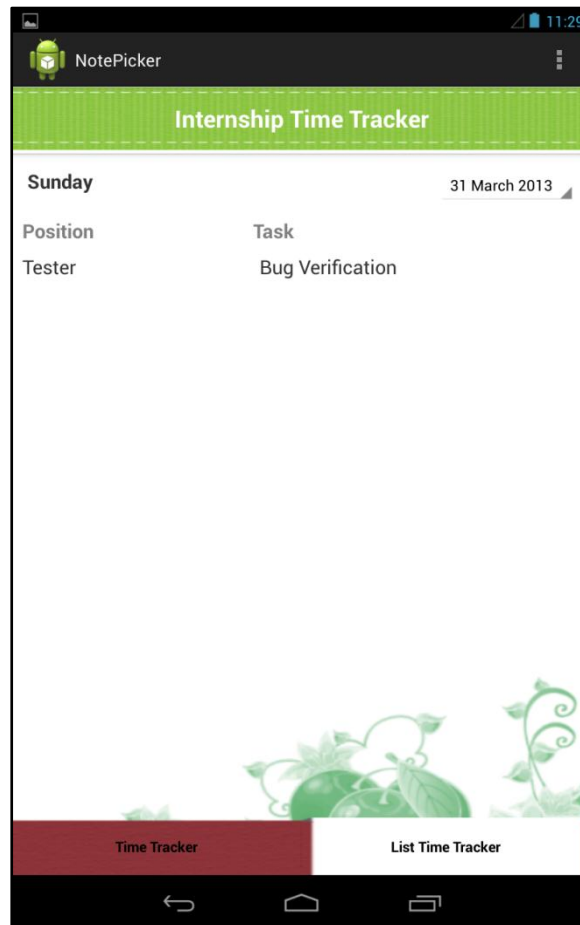


Figure 8: Time Treadmill Output Page

This is the second page of the application where saved note can be view. On the interface, it will show latest date on the top right, day according to the date at the top left of the interface, list of “Position” and list of “Task” on selected date.

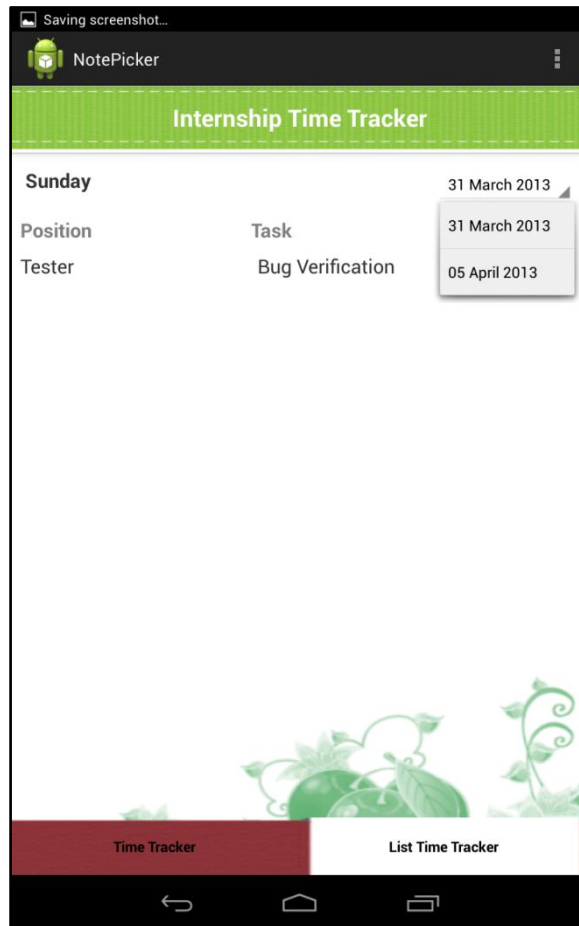


Figure 9: Time Treadmill Output Page Picking Date

To view other saved note, user just need to simply click on the date at the top right of the interface. A list of date will be showed based on note that user had written on the showed date. User can click on any showed date and the “Position” and “Task” list will be refresh to show data from selected date.

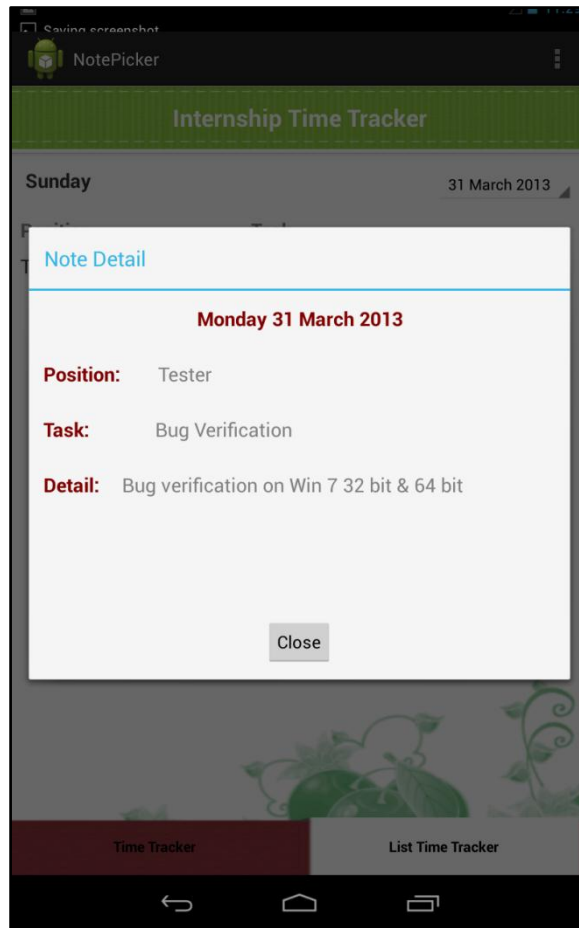


Figure 10: Time Treadmill Output Page Showing Note Detail

To view full details of the note, user can click on listed “Position” or “Task”. An overlay interface will pop up as above and show the full details based on selected “Position” or “Task”. The details will be based on what user has key in on previous page.

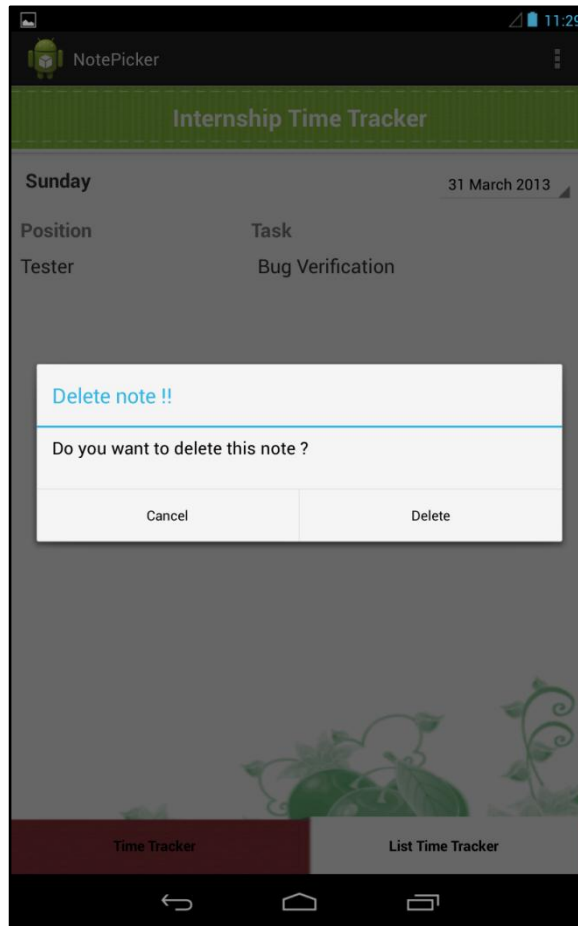


Figure 11: Time Treadmill Output Page Delete Confirmation

In this page user can delete any unused or wrong input data. To delete, user just need to click on the selected “Position” or “Task” for a few second. An overlay pop up will show up and prompt user for the confirmation of deletion as in picture above. To delete user need to click on the “Delete” button and to cancel user need to click on “Cancel” button.

CHAPTER 4

RESULT & DISCUSSION

4.1 User Feedback Survey

As I have finished the prototype of the application, it's time to gather some feedback regarding the prototype to help in the improvement of Time Treadmill. To do so, I have done surveys to get feedback from current students who are doing internship. I have created a survey with four questions and it was given to current industrial internship students aged between 20 to 22. Most of the participants were doing their internship in Kuala Lumpur and some in Sabah & Sarawak. There are 40 people who participate in this survey. Below is the result of the survey:

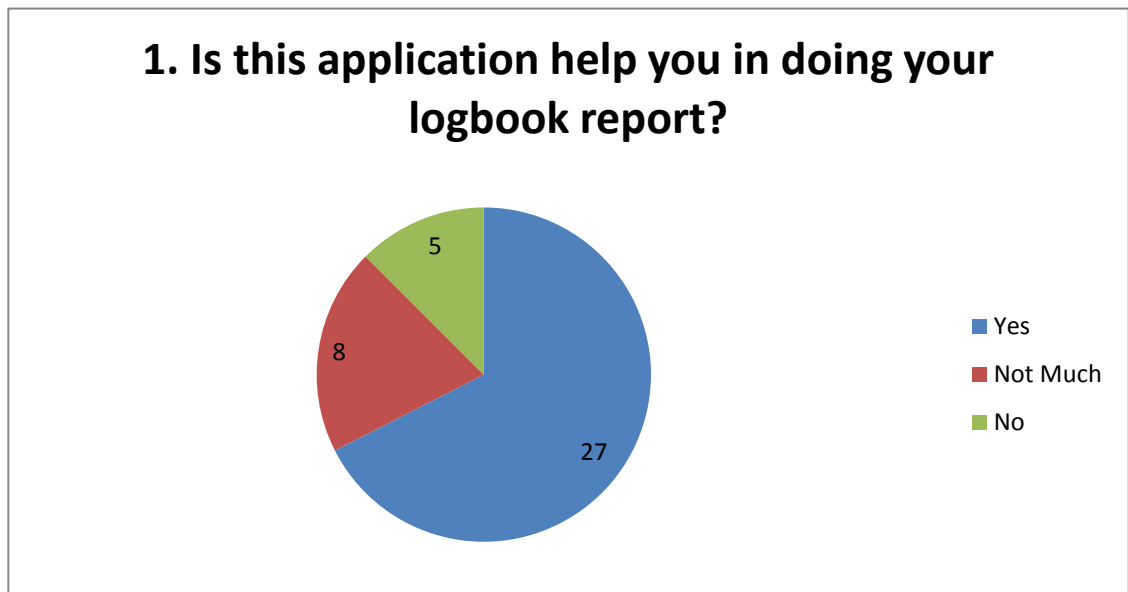


Figure 12: Survey question number one

The first question for this survey is to see user reaction regarding the application. It asks on whether this application has helped them in doing their daily activity report. According to above result, more than half of the participants agree that this application has help them in doing their daily activities report. They say that a mobile application is a very brilliant idea to help them record their daily activity as they always have their phone with them all the time. Besides, some also say that the application is very simple and easy to use. About 8 participants state that the application is not very useful to them. This is because they are doing desk work during internship which makes it easy for them to write their report directly using provided logbook. Another 5 participant's state that the application is not helping them at all. One of the factors is because they have no

problem carrying their logbook to their office. Some also said that they leave their logbook in the office so that they can write their report daily before leave for home.

The next question in the survey is to ask participants if the application has the potential to substitute current conventional method of writing logbook report.

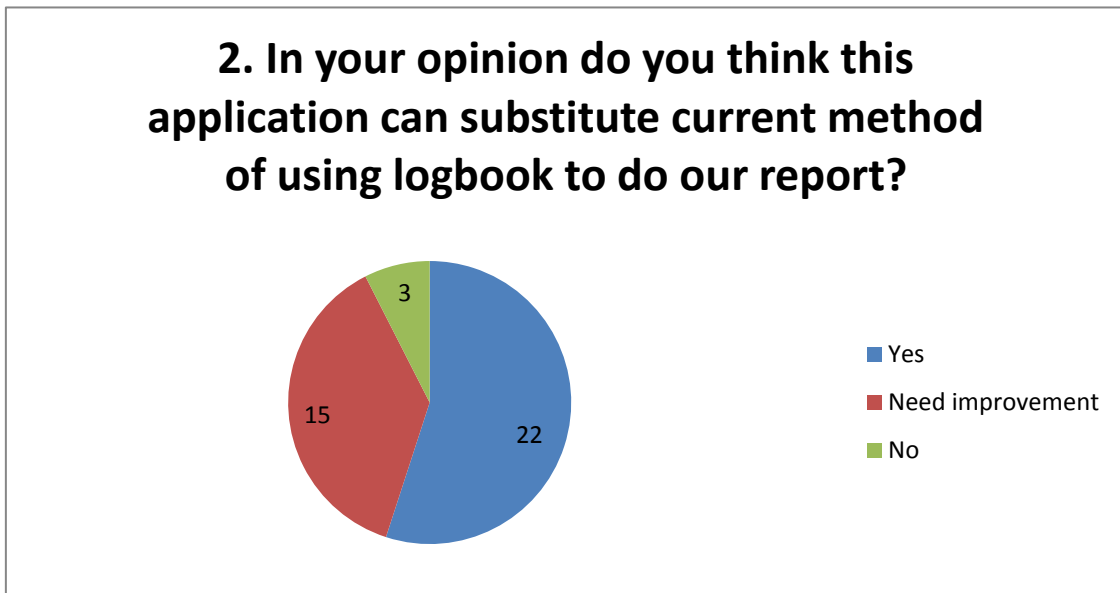


Figure 13: Survey question number two

According to the result, 22 participants agree that the application might be a good effort to substitute current conventional method of daily activities report writing. This is a positive indication that show mobile application is a suitable medium to be used in education sector. Another 15 participants state that the application needs more improvement if it wants to be used to substitute current logbook. As this application is still in early stage, it can be argue that this application will need more improvement to make it better and more efficient in order to serve it purpose. But, about 3 participants state no and refuse the idea of substitute current method of report writing with mobile application.

The third question is asking participants regarding improvement that they want to see in this application. There are three choices which are “Picture Snap”, “Voice Recording” & “Timestamp”. Besides that, participants can also give suggestion on what improvement that they want to see in this application.

3. What improvement do you think will be useful to be implement in the application

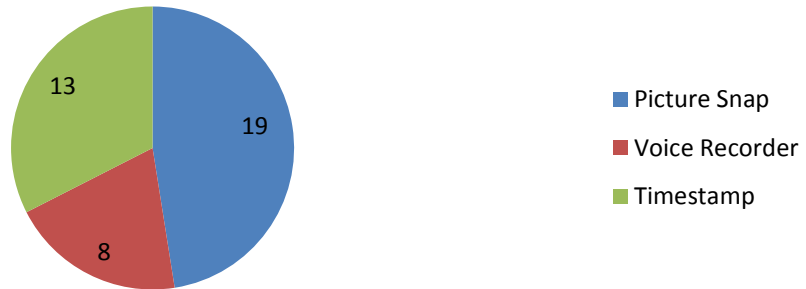


Figure 14: Survey question number three

Based on the results, most participants interested in “Picture Snap” improvement for this application. “Picture Snap” will allow user to snap a picture and write a report regarding snapped picture. This is very useful especially for engineering students because they can easily take picture of any work-related that is to be included in their report. Second improvement that they like is the “Timestamp” feature. Currently, this application does not have timestamp feature in it. It only takes current date as report indicator. Timestamp will be very useful especially for those who involve in a lot of meeting during their internship. Lastly, the least preferred feature is “Voice Recorder”. Maybe because most of the participants are engineer, so “Voice Recorder” feature will be less useful for them.

The last question in this survey is to rate this application based on its usefulness to the participants. There are 5 options from 1 to 5 with 1 is the lowest and five is the highest.

4. From 1-5 (1 is lowest and 5 is highest) how will you describe this application in terms of usefulness?

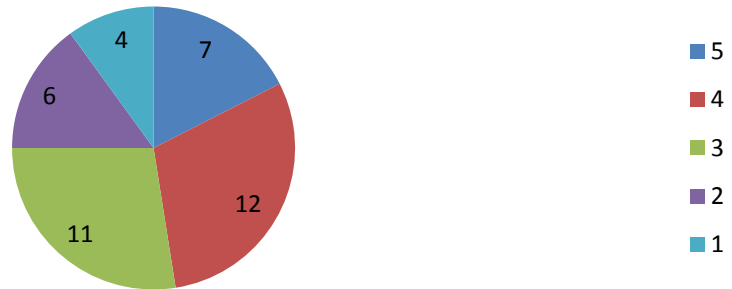


Figure 15: Survey question number four

From the survey, most participants rate this application at “4” with 12 participants. With basic functionality of the prototype, “4” can be considered satisfying enough. Another 11 participants rate at “3”, 7 participants rate at “5”, 6 participants rate at “2” and 4 participants rate at “1”.

Based on the result, this application is worth to be developed as most of the participants give positive and building comment regarding this application. This show that students are supporting this project as it helps them with their task. Thus, it can be concluded that this application has served its main objective which is to design a mobile application that help students with their logbook report.

CHAPTER 5 CONCLUSION & RECOMMENDATION

Internship time treadmill will bring a difference and benefit to industrial internship students in creating their report. This project highlights the usage of mobile application to assist students in doing their internship logbook report. The design and development of mobile application logbook which reflect the real logbook will help students to get use to the application easily. It is not only beneficial for students, but might also help to ease supervisor and industrial internship unit in tracking and rate students' progress. Still, the main contribution of this project will be for internship students in order to create logbook report efficiently and effectively.

After finishing the prototype and release it to a number of participants, I have received positive feedback and supportive suggestion to further improve this application. Indirectly, this show that students are supporting this application and want to see that this application be more useful and effective in serving its purpose. This can be prove by all the positive feedback and respond from survey participants.

Besides that, most of the participants agree that this application has helped them in writing their logbook report. Thus, it can be concluded that this application has accomplish its main objective; to design a mobile application that help students with their logbook report. As for second objective, it's going to take some time to be fully achieve because to provide full mobility mean to support all operating system currently available in the market. But, currently this application only supports Android OS. For the third objective, it will consume more time as it will need a very effective and thorough research and discussion with Industrial Internship Unit if I want to use this application to substitute current logbook.

For future plan, this application will be keep updating with new features through feedback from users such as “Picture Snap” and “Timestamp”. I want to keep this project as community-based that is consists of students. This will help me to better understand their problem during internship and improve this application. Furthermore, this application will also be develop to support other platform as well such as Windows Mobile and iOS.

REFERENCES

Wikipedia (2012) Logbook.

Available at: <http://en.wikipedia.org/wiki/Logbook>

The Journal (Meyer, 2009) Report Recommends Shift to Digital Educational Resources Within 5 Years.

Available at: <http://thejournal.com/articles/2012/09/24/report-recommends-shift-to-digital-educational-resources-within-5-years.aspx>

SpinSucks(Dietrich, 2012) Can Algorithms Replace Human Writers?

Available at: <http://spinsucks.com/communication/can-algorithms-replace-human-writers-part-two/>

GreenAnswers (2009).

Available at: <http://greenanswers.com/q/32498/forests-trees-plants/how-many-are-cut-down-make-paper-every-year>

Techsoup (2009) Paper Shocking Statistics.

Available at: <http://www.techsoup.org/greentech/paper/page10834.cfm>

Mashable Tech (Laird, 2012) In A Relationship: College Students and Their Smartphones.

Available at: <http://mashable.com/2012/06/30/smartphones-college-students-infographic/>

GSM Arena (2011) Samsung Galaxy Y S5360.

Available at: http://www.gsmarena.com/samsung_galaxy_y_s5360-4117.php

TheGuardian (Eustice, 2011) How to build a university mobile application: best practice and insight.

Available at: <http://www.guardian.co.uk/higher-education-network/2011/dec/08/building-university-mobile-apps>

APPENDICES

Time Treadmill Feedback Survey

Age: _____

Year: _____

Place of Internship: _____

Please tick [] for the answer of each of the questions below

1. Do you think the application help you in doing your logbook report?

- Yes
- Not Really
- No

Kindly state any comment based on your answer

2. In your opinion do you think this application can substitute current method of using logbook to do our report?

- Yes
- Need more improvement
- No

3. What improvement do you think will be useful to be implement in the application

- Picture Snap
- Voice Recorder
- Timestamp

Kindly state if you have any other suggestion:

From 1-5 (1 is lowest and 5 is highest) how will you describe this application in terms of usefulness?

1

2

3

4

5

Time Treadmill Source Code

DBAdapter.java:

```
package com.example.notepicker;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteDatabase.CursorFactory;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;
import android.widget.Toast;

public class DBAdapter {
    private SQLiteDatabase mDB;

    public static final String TAG = "DBAdapter";

    private DatabaseHelper mDbHelper;

    private Context mContext;

    private int DATABASE_VERSION = 2;

    private static final String TimeTracker = "CREATE TABLE note (date text, position
text, task text, detail text)";

    private static final String DATABASE_NAME = "Database_Note";

    public static class DatabaseHelper extends SQLiteOpenHelper {

        public DatabaseHelper(Context context, String name, CursorFactory factory, int version)
        {
            super(context, name, factory, version);
        }
    }
}
```

```
}
```

```
@Override
```

```
public void onCreate(SQLiteDatabase db) {
```

```
// TODO Auto-generated method stub
```

```
db.execSQL(TimeTracker);
```

```
}
```

```
@Override
```

```
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
```

```
// TODO Auto-generated method stub
```

```
Log.i(TAG, "Upgrading DB");
```

```
db.execSQL("DROP TABLE IF EXISTS users");
```

```
onCreate(db);
```

```
}
```

```
}
```

```
public DBAdapter(Context ctx) {
```

```
this.mContext = ctx;
```

```
mDbHelper = new DatabaseHelper(ctx, DATABASE_NAME,  
null,DATABASE_VERSION);
```

```
}
```

```
public DBAdapter open() {
```

```
mDbHelper = new DatabaseHelper(mContext, DATABASE_NAME,  
null,DATABASE_VERSION);
```

```
if (mDbHelper != null)
```

```
mDB = mDbHelper.getWritableDatabase();

return this;

}

public void close() {

mDbHelper.close();

}

public void insertNote(String date, String position, String task, String detail)

{

ContentValues insertNote = new ContentValues();

insertNote.put("date",date);

insertNote.put("position",position);

insertNote.put("task",task);

insertNote.put("detail",detail);

long result = mDB.insert("note", null, insertNote);

if (result!=0)

{

Toast.makeText(mContext, "Save note successful", Toast.LENGTH_SHORT).show();

}

}

public Cursor getNote(String date) {

return mDB.query("note", new String[] { "date", "position",

"task", "detail" }, "date = " + date + "",

null, null, null, null);

}
```

```
public Cursor getNote() {
return mDB.query("note", new String[] { "date", "position",
"task", "detail" },null,null, null, null, null);
}

public void deleteNote(String date, String position, String task)
{
mDB.delete("note", "date=? AND position=? AND task=?", new
String[]{date,position,task});
}
}
```

TimeTracker.java:

```
package com.example.notepicker;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.DatePickerDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.database.Cursor;
```

```
import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.DatePicker;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

public class TimeTracker extends Activity {

    private TextView datePicker;

    private TextView dayOfWeek;

    private Context mContext;

    private Button btnSave;

    private Button btnClear;

    private Date mDate;

    private int mDay;

    private int mYear;

    private int mMonth;

    private Calendar mCalendar;

    private DatePickerDialog mDatePicker;

    private String iDate;

    private DBAdapter mDB;

    private EditText ePosition,eTask,eDetail;

    @SuppressWarnings("SimpleDateFormat")

    @Override
```



```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.time_tracker);
    mDB = new DBAdapter(TimeTracker.this);
    mDB.open();
    dayOfWeek = (TextView)findViewById(R.id.txtDayOfWeek);
    datePicker = (TextView)findViewById(R.id.datePicker);
    mCalendar = Calendar.getInstance();
    ePosition = (EditText)findViewById(R.id.ePosition);
    eTask = (EditText)findViewById(R.id.eTask);
    eDetail = (EditText)findViewById(R.id.eDetails);
    mContext = this;
    mDay = mCalendar.get(Calendar.DAY_OF_MONTH);
    mMonth = mCalendar.get(Calendar.MONTH);
    mYear = mCalendar.get(Calendar.YEAR);
    Date nDate = new Date(mYear,mMonth,mDay);
    iDate = String.valueOf(nDate.getTime());
    String format = "dd MMMM yyyy";
    String format1 ="EEEE";
    SimpleDateFormat sf = new SimpleDateFormat(format);
    SimpleDateFormat sf2 = new SimpleDateFormat(format1);
    dayOfWeek.setText(sf2.format(mCalendar.getTime()));
    datePicker.setText(sf.format(mCalendar.getTime()));
    mDatePicker = new DatePickerDialog(mContext,new
    DatePickerDialog.OnDateSetListener() {
```

```
@SuppressWarnings("deprecation")

public void onDateSet(DatePicker view, int year,int monthOfYear, int dayOfMonth) {

    mYear = year;

    mMonth = monthOfYear;

    mDay = dayOfMonth;

    mDate = new Date(mYear, mMonth, mDay);

    iDate = String.valueOf(mDate.getTime());

    switch (mDate.getDay()) {

    case 1:

        dayOfWeek.setText("Sunday");

        break;

    case 2:

        dayOfWeek.setText("Monday");

        break;

    case 3:

        dayOfWeek.setText("Tuesday");

        break;

    case 4:

        dayOfWeek.setText("Wednesday");

        break;

    case 5:

        dayOfWeek.setText("Thursday");

        break;

    case 6:

        dayOfWeek.setText("Friday");
```

```
break;

case 7:

    dayOfWeek.setText("Saturday");

break;

default:

break;

}

String format = "dd MMMM";

SimpleDateFormat sf = new SimpleDateFormat(format);

//datePicker.setText(sf.format(mDate));

String date = sf.format(mDate)+" "+(mDate.getYear());

datePicker.setText(date);

}

},mYear,mMonth,mDay);

datePicker.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

mDatePicker.show();

}

});

btnSave = (Button)findViewById(R.id.btnSave);
```

```
btnSave.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        // TODO Auto-generated method stub

        String position = ePosition.getEditableText().toString().trim();

        String task = eTask.getEditableText().toString().trim();

        String detail = eDetail.getEditableText().toString().trim();

        if(position.equals("") || task.equals("") || detail.equals(""))

        {

            AlertDialog.Builder alert = new AlertDialog.Builder(mContext);

            alert.setMessage("Please check your input carefully!");

            alert.setTitle("Warning!");

            alert.setPositiveButton("Ok", new DialogInterface.OnClickListener() {

                @Override

                public void onClick(DialogInterface dialog, int which) {

                    // TODO Auto-generated method stub

                }

            });

            alert.show();

        }

        else
```

```
{
    mDB.insertNote(iDate, position, task, detail);
    ePosition.setText("");
    eTask.setText("");
    eDetail.setText("");
}
}
});

btnClear = (Button)findViewById(R.id.btnClear);
btnClear.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        ePosition.setText("");
        eDetail.setText("");
        eTask.setText("");

        int day = mCalendar.get(Calendar.DAY_OF_MONTH);
        int month = mCalendar.get(Calendar.MONTH);
        int year = mCalendar.get(Calendar.YEAR);

        Date cDate = new Date(year,month,day);
        iDate = String.valueOf(cDate.getTime());
        String format = "dd MMMM yyyy";
        String format1 = "EEEE";

        SimpleDateFormat sf = new SimpleDateFormat(format);
```

```
SimpleDateFormat sf2 = new SimpleDateFormat(format1);
dayOfWeek.setText(sf2.format(mCalendar.getTime()));
datePicker.setText(sf.format(mCalendar.getTime()));
}
});
}
}
```

ListTimeTracker.java:

```
package com.example.notepicker;

import java.sql.Date;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashSet;

import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.AlertDialog;
```

```
import android.app.Dialog;

import android.content.DialogInterface;

import android.database.Cursor;

import android.os.Bundle;

import android.view.View;

import android.widget.AdapterView;

import android.widget.AdapterView.OnItemClickListener;

import android.widget.AdapterView.OnItemLongClickListener;

import android.widget.AdapterView.OnItemSelectedListener;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.ListView;

import android.widget.Spinner;

import android.widget.TextView;

import android.widget.Toast;

public class ListTimeTracker extends Activity{

private Spinner spinnerDate;

private TextView dayOfWeek;

private DBAdapter mDB;

ArrayAdapter<String> listDate;

ArrayList<String>arr;

private ListView listTask;

NoteAdapter noteAdapter;

private String formatDate;
```

```
@SuppressWarnings("deprecation")
@SuppressLint("SimpleDateFormat")
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.list_time_tracker);
    spinnerDate = (Spinner)findViewById(R.id.spinnerDate);
    dayOfWeek = (TextView)findViewById(R.id.txtDayOfWeek);
    noteAdapter = new NoteAdapter(ListTimeTracker.this);
    mDB = new DBAdapter(ListTimeTracker.this);
    mDB.open();
    arr = new ArrayList<String>();
    listDate = new ArrayAdapter<String>(this,android.R.layout.simple_spinner_item);
    listDate.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
    ;
    spinnerDate.setAdapter(listDate);
    listTask = (ListView)findViewById(R.id.listTimeTracker);
    listTask.setAdapter(noteAdapter);
    Cursor mCursor = mDB.getNote();
    mCursor.moveToFirst();
    while (!mCursor.isAfterLast()){
        arr.add(mCursor.getString(0));
        mCursor.moveToNext();
    }
    HashSet<String> hs = new HashSet<String>();
```



```
hs.addAll(arr);

arr.clear();

arr.addAll(hs);

Collections.sort(arr);

formatDate = "EEEE dd MMMM";

String format = "dd MMMM";

String format1 = "EEEE";

SimpleDateFormat sf = new SimpleDateFormat(format);

final SimpleDateFormat sf1 = new SimpleDateFormat(formatDate);

SimpleDateFormat sf2 = new SimpleDateFormat(format1);

for(int j=0;j<arr.size();j++)

{

Date mDate = new Date(Long.parseLong(arr.get(j)));

listDate.add(sf.format(mDate)+" "+mDate.getYear());

}

listDate.notifyDataSetChanged();

noteAdapter.notifyDataSetChanged();

spinnerDate.setOnItemClickListener(new OnItemSelectedListener() {

@Override

public void onItemClick(AdapterView<?> adapter, View arg1,

int position, long arg3) {

// TODO Auto-generated method stub

long day = Long.parseLong(arr.get(position));

String strDay = arr.get(position);

Date nDate = new Date(day);
```

```
switch (nDate.getDay()) {  
  case 1:  
    dayOfWeek.setText("Sunday");  
    break;  
  case 2:  
    dayOfWeek.setText("Monday");  
    break;  
  case 3:  
    dayOfWeek.setText("Tuesday");  
    break;  
  case 4:  
    dayOfWeek.setText("Wednesday");  
    break;  
  case 5:  
    dayOfWeek.setText("Thursday");  
    break;  
  case 6:  
    dayOfWeek.setText("Friday");  
    break;  
  case 0:  
    dayOfWeek.setText("Saturday");  
    break;  
  default:  
    break;  
}
```

```
Cursor dayCursor;

dayCursor = mDB.getNote(strDay);

dayCursor.moveToFirst();

noteAdapter.clearAdapter();

while(!dayCursor.isAfterLast())

{

Note mNote = new
Note(dayCursor.getString(0),dayCursor.getString(1),dayCursor.getString(2),dayCursor.
getString(3));

noteAdapter.addNote(mNote);

dayCursor.moveToNext();

}

noteAdapter.notifyDataSetChanged();

}

@Override

public void onNothingSelected(AdapterView<?> arg0) {

// TODO Auto-generated method stub

}

});

listTask.setOnItemClickListener(new OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> adapter, View arg1, int arg2,
long arg3) {
```

```
// TODO Auto-generated method stub

final Dialog mDialog = new Dialog(ListTimeTracker.this);
mDialog.setContentView(R.layout.note_detail);
Button btnClose = (Button)mDialog.findViewById(R.id.btnClose);
TextView txtPosition = (TextView)mDialog.findViewById(R.id.txtPosition);
TextView txtTask = (TextView)mDialog.findViewById(R.id.txtTask);
TextView txtDetail = (TextView)mDialog.findViewById(R.id.txtDetail);
TextView txtDate = (TextView)mDialog.findViewById(R.id.txtDate);
Note mNote = (Note)adapter.getItemAtPosition(arg2);
txtPosition.setText(mNote.getPosition());
txtTask.setText(mNote.getTask());
txtDetail.setText(mNote.getDetails());
Long tempDate = Long.parseLong(mNote.getDate());
Date dt = new Date(tempDate);
txtDate.setText(sf1.format(dt)+" "+dt.getYear());
mDialog.setTitle("Note Detail");
btnClose.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

mDialog.dismiss();

}

});

mDialog.show();
```

```
}  
});  
listTask.setOnItemLongClickListener(new OnItemLongClickListener() {  
  
    @Override  
    public boolean onItemLongClick(final AdapterView<?> adapter, View arg1,  
    final int pos, long arg3) {  
        // TODO Auto-generated method stub  
        final AlertDialog.Builder deleteDialog = new  
        AlertDialog.Builder(ListTimeTracker.this);  
        deleteDialog.setTitle("Delete note !!");  
        deleteDialog.setMessage("Do you want to delete this note ?");  
        deleteDialog.setPositiveButton("Delete", new DialogInterface.OnClickListener() {  
  
            @Override  
            public void onClick(DialogInterface dialog, int which) {  
                // TODO Auto-generated method stub  
                String date, position, task;  
                Note n = (Note)adapter.getItemAtPosition(pos);  
                date = n.getDate();  
                position = n.getPosition();  
                task = n.getTask();  
                mDB.deleteNote(date, position, task);  
                noteAdapter.deleteNote(pos);  
                listDate.notifyDataSetChanged();  
            }  
        }  
    }  
});
```

```
noteAdapter.notifyDataSetChanged();
}
});

deleteDialog.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

// TODO Auto-generated method stub

}

});

deleteDialog.show();

return false;

}

});

}

}
```

Note.java:

```
package com.example.notepicker;

import java.util.Date;

public class Note {

private String mDate;

private String mPosition;

private String mTask;
```

```
private String mDetails;

public Note(String date,String ps,String tsk, String dtl)

{

this.mDate = date;

this.mPosition = ps;

this.mTask = tsk;

this.mDetails=dtl;

}

public String getDate()

{

return this.mDate;

}

public String getPosition()

{

return this.mPosition;

}

public String getTask()

{

return this.mTask;

}

public String getDetails()

{

return this.mDetails;

}
```

```
public void setDate(String date)
{
    this.mDate = date;
}

public void setPosition(String position)
{
    this.mPosition = position;
}

public void setTask(String task)
{
    this.mTask = task;
}

public void setDetail(String detail)
{
    this.mDetails = detail;
}
}
```

NoteAdapter.java:

```
package com.example.notepicker;

import java.util.ArrayList;

import java.util.List;

import android.content.Context;

import android.view.LayoutInflater;
```



```
import android.view.View;

import android.view.ViewGroup;

import android.widget.BaseAdapter;

import android.widget.TextView;

public class NoteAdapter extends BaseAdapter{

    Context mContext;

    List<Note> mNote;

    public NoteAdapter(Context mContext)

    {

        this.mContext=mContext;

        this.mNote = new ArrayList<Note>();

    }

    public void addNote(Note n)

    {

        mNote.add(n);

    }

    public void clearAdapter()

    {

        mNote.clear();

    }

    public void deleteNote(int pos)

    {

        mNote.remove(pos);

    }

}
```

```
@Override

public int getCount() {

// TODO Auto-generated method stub

return mNote.size();

}

@Override

public Object getItem(int position) {

// TODO Auto-generated method stub

return mNote.get(position);

}

@Override

public long getItemId(int position) {

// TODO Auto-generated method stub

return 0;

}

@Override

public View getView(int position, View convertView, ViewGroup parent) {

// TODO Auto-generated method stub

Note lst = mNote.get(position);

LayoutInflater inflater = LayoutInflater.from(mContext);

if(convertView==null)

{

convertView=inflater.inflate(R.layout.one_note, null);
```

```
}  
  
TextView pos = (TextView) convertView.findViewById(R.id.txtPosition);  
  
TextView task = (TextView) convertView.findViewById(R.id.txtTask);  
  
pos.setText(lst.getPosition());  
  
task.setText(lst.getTask());  
  
//TextView id = (TextView) convertView.findViewById(R.id.txtIDmonan);  
  
//id.setText(lst.getID());  
  
return convertView;  
  
}  
  
}
```

NotePicker.java:

```
package com.example.notepicker;  
  
import android.app.TabActivity;  
  
import android.content.Intent;  
  
import android.os.Bundle;  
  
import android.view.Menu;  
  
import android.view.View;  
  
import android.widget.TabHost;  
  
import android.widget.TabHost.OnTabChangeListener;  
  
import android.widget.TabHost.TabSpec;  
  
  
public class NotePicker extends TabActivity {  
  
    @SuppressWarnings("deprecation")
```

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_note_picker);  
    final TabHost tabHost = getTabHost();  
    final TabSpec dsmonan = tabHost.newTabSpec("Time Tracker");  
    View indicator11 = getLayoutInflater().inflate(R.layout.time_tracker_bg,null);  
    dsmonan.setIndicator(indicator11);  
    Intent lmonanIntent = new Intent(this,TimeTracker.class);  
    dsmonan.setContent(lmonanIntent);  
    tabHost.addTab(dsmonan);  
  
    TabSpec nguyenlieu = tabHost.newTabSpec("List Time Tracker");  
    View indicator12 = getLayoutInflater().inflate(R.layout.list_time_tracker_bg,null);  
    nguyenlieu.setIndicator(indicator12);  
    Intent nguyenlieuIntent = new  
    Intent(this,ListTimeTracker.class).addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);  
    nguyenlieu.setContent(nguyenlieuIntent);  
    tabHost.addTab(nguyenlieu);  
}
```

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {  
    // Inflate the menu; this adds items to the action bar if it is present.  
    getMenuInflater().inflate(R.menu.activity_note_picker, menu);  
}
```

```
return true;
```

```
}
```

```
}
```