**INTRODUCTION**

**1.1 Preamble :-**

The android application scales frequencies, governor and i/o scheduler of a device to provide just the right combination which will lead to maximum prolonged power consumption.

**1.2 Objective:-**

The project aims to introduce a certain application allowing the user to change hardware aspects of the device running the android OS to yield better power consumption from the device.

**1.3 Scope:-**

The application can be used by various type of users of android, whether some one uses it extensively performing processor extensive tasks or casual users concerned about the longevity of the battery life.

**1.4 Salient Features:-**

Ubiquitous:-It can be used on any version of android provided the device has superuser rights or is rooted and on any vendor's hardware whether its HTC,Samsung,LG etc.

Cost effective:- In place of purchasing applications which provide the same functionality,the application is free,small in size and will provide a significant power saving.

Constant Feedback and Adjustment:- The user can constantly see whether the application is performing its intended task or not and adjust parameters according to its current usage.

**1.5 Broad outline of project:-**

The application provides the user with the options to change the frequency,governer and I/O scheduler which the hardware supports,with these options the user can find the right settings for its usage and hardware that provides maximum power saving.

**1.6 About the subsequent chapters**

1.6.1. Chapter 2 : Literature Survey

Literature survey provides the background material for the project. It gives the various concept details of the project.

1.6.2. Chapter 3 : Analysis

Analysis is the first step of software project development. The objective of this is to develop a model that describes the software as it works to satisfy a set of customers requirements. In this phase we are analyzing the project with different perspective, analyzing the requirements, selecting the software process model, analyzing the future risk.

1.6.3. Chapter 4 : Design and Implementation

This chapter deals with the design model. Different UML diagrams like Activity, Use-case, etc. are drawn to explain the purpose and working of a project.

1.6.4 Chapter 5 : Testing

This chapter deals with tests that systematically uncover different classes of errors and to do so with minimum amount of time and effort.

**LITERATURE SURVEY**

Survey is performed under the following topics:

**2.1 Need of such an Application**

* Battery life is the bane of smartphone owners everywhere. It's lovely being able to browse the net, play games, watch videos and update your status from everywhere and anywhere. But doing so too much can turn your phone into an expensive paperweight- at least until you get home and get it plugged in.All is not lost though, as there are numerous ways to eke out a little (or a lot) more juice from your ailing battery.
* **2.2 Reasons for Discharging too early**
* Vibration:-
* The vibration function on your phone, along with haptic feedback, uses a tiny motor which rotates a weight at high speeds to turn electric energy into kinetic energy and cause the phone to vibrate.

Motor, operated for a short period of time before being turned off and then on again, creates short spikes of current which use more energy than a sustained level. The energy required to do all that is not insubstantial and is actually a bigger drain on the battery than a ringtone, which only requires a small vibration to produce sounds through the phones speaker

* User Display:-

Just having the screen on is one of the biggest battery drains for a phone, and obviously the bigger the screen, the bigger the drain. Having it brightly lit sucks the power down harder than an aggressive shower drain, as the phone requires more power to sustain a bright light than a dim one.

Screen simply converts electrical energy into light energy. On a full HD screen the energy required to change the color of each pixel is substantial. On top of that, brighter screens require more electrical energy to convert into light energy.

Many phones have very bright screens a,so the brightness can be dropped to around 40% or lower.Alternatively you could activate the 'auto brightness' mode that most phones have, which will automatically adjust the brightness as needed, dimming it when your surroundings are dark and brightening it when they're light.

It could also be worth adjusting how long it takes before your screen times out (switches itself off). If it stays on for two minutes every time you get a text, or check the time, that can quickly add up so consider lowering it to more like fifteen or thirty seconds.

Smartphone's Radios:-

Leaving Wi-Fi on without a connection,your phone will keep checking for Wi-Fi networks and constantly trying to connect to open ones, which uses power and can be a significant battery drain, so turn Wi-Fi off when you're not connected to a network.Similar principles apply to Bluetooth, GPS and 3G/4G. So while not using them, turn them off.Modern batteries and CPUs are designed to minimise these effects, so the issues aren't as bad as they used to be, but turning them off will definitely save some charge.Any app, system process or Google service can potentially drain your battery as they all require CPU power to run and some also download data (for example an app that's syncing).It's not always obvious which the main culprits are, especially as apps often run in the background, so it's worth checking and many phones come with task managers that make this simple.

If it turns out you've got a rogue app on your hands at least then you'll know to delete or disable it, or if for example you find that the screen is the main drain, you'll know to turn down the brightness.

### Heating issues

It can be due to the processor is being heavily worked by some rogue app and that increased workload will have a seriously negative effect on your battery life.This isn't always avoidable, but if you're not sure what's causing your processor to work so hard check your task manager, in case there's anything clogging things up in the background and stop anything that looks a bit suspicious.Beyond that, your phone's battery can get hot based on the surroundings it's in, so you can help it out a little by keeping it away from hot environments. That might mean not leaving it near a radiator or on the dashboard of your car, or near other electronics that are pumping out heat.

If a battery does get too hot (be it from heavy use or a warm environment) it can cause it to degrade faster and extreme temperatures can even cause the electrolyte in the battery to ignite, starting a fire (though this is really, really rare) In the short term, lithium-ion batteries like the ones found in smartphones can actually perform better at high temperatures, as the heat lessens the internal resistance,meaning the factors that slow the flow of current within the battery, such as the resistivity of the internal components along with ionic resistance caused during electrochemical reactions, is basically negated somewhat.

High temperatures can speed up electrochemical reactions, lessening this aspect of the resistance. However this also stresses the battery, causing it to degrade faster and hold charge a little less well.But extremely low temperatures are also best avoided as they increase the internal resistance of a lithium-ion battery, by slowing down the electrochemical reactions, causing it to work less efficiently.As such batteries that are especially cold are likely to see a decrease in performance, though cold conditions have no real long term effects on a battery, so it should return to normal once warm.Generally there's only a significant impact on the battery if the temperatures reach fairly extreme levels though and many phones even have built in ways to combat heat. Samsung devices have temperature controls built in to ensure that the device and the battery will never get to a detrimental temperature, so if the device heats up we can downscale the processor to reduce the temperature.

**2.3 Different User's Requirements**

Kids,teenagers and tech enthusiast:-They use their smartphones heavily,use computational expensive applications every now and them,play games which use cutting edge technology hardware.Also,they download and use a lot of applications.The app should be configured at highest frequency and governer to performance.

Youth and middle aged people and office going people:-They usually dont perform computationally expensive tasks but would constantly call,message and use social networking websites and applications like watsapp,facebook,twitter.The app should be configured on frequencies in the middle range and governer to powersave.

Old people:-They use the smartphone basically as a basic phone,they mainly call and text only.The app should be configured to lower frequencies that still provide stable usage.In this way the smartphone will last unusually longer.

**2.4 Steps of selecting the right combination**

* Decide what type of user you are
* Apply the options through the application and
* Analyze the output

Step 1: Decide what type of user you are

Step is self explanatory,of the three type listed above select the settings corresponding to your usage.

Step 2: Apply the options through the application

Select the options on the apps main activity and push apply.

Step 3: Analyze the output

Does the application is able to prolong the smartphone as you expected,for that click the battery option in the settings.

Study the total time the device was on,the screen was lit up,which applications used what percentage.On clicking the graph,you will find which radios were on like wifi,mobile 2g/3g connection for how much time.

If you are satisfied with the power save,keep the settings applied.Otherwise apply a new combination and then study its output.

**2.5 Working of the application**

Whenever a User select options and hit apply:-

The application changes the values stored in following files at location /sys/devices/system/cpu/cpu0/cpufreq :-

1)scaling\_min\_freq

2)scaling\_max\_freq

3)scaling\_governor

**2.6 Limitations of the Application**

(i) If device does not provide scaling options,then the application becomes ineffective.

(ii) The user has to be knowledgeable enough to select and deduce the right combinations.

(iii) Overclocking can heat up the device whereas underclocking can lead to lags and freezes.

Explanations

1. 1) The testing device has a frequency range in 1008-122 Mhz with options to set frequency at 122,240,320,480,
2. 600,600,1008 Mhz.The governer can be selected from set of performance,interactive,conservative ondemand, userspace,powersave, governers. The I/O scheduler could either be cfq or noop. But in devices if such vast options are not available,then application can do little as hardware does not support such scaling.
3. 2) The effectivenes of the application depends directly on its usage by the user,thus the user should be aware of the basic terminologies used and its applications inorder to have the desired effect.If a user does not study the relationship between frequency,governer,scheduler and power saving,then he might not have the right settings selected.
4. 3) Inorder to get the maximum performance all the time having minimum and maximum frequencies set the highest limit can lead to heating of the processor,also if the user sets the maximum frequency too low,then even the basic operations and tasks of the phone will produce lag and annoying freezez.S,the user should keep this in mind.

**2.7 APPLICATIONS.**

In addition of saving the battery,the user can use the application in following ways:

* If the user wants to play games,user can select the highest frequenies and performance governor temporarily.
* If the user is on a journey or on a flight,he can intentionally select the lowest frequencies , so that the phone can be used for a long time.

**ANALYSIS**

This chapter is a very important chapter in the PHASE-I of this project as it analyses each and every aspect of the project. All the research done for this project has been minutely analyzed here. The analysis has been carried out by carefully understanding the problem statement in greater depth.

**3.1 Conception Phase**

The conception phase consists of the following

3.1.1 Feasibility study:

*3.1.1.1. Time:*

Can it be deployed in the available time?

The waterfall model is being followed.As per adhering to the time limits,different features will be added in increments,basic features first and advanced at the later stages.Thus ensuring the completion of the project.

*3.1.1.2. Technical Feasibility:*

The project includes the usage of JDK,Android SDK,Android studio and android emulator.These technologies are free to use and are familiar and user friendly ,hence can be used without much difficulty.

*3.1.1.3. Operational Feasibility:*

User needs a device running Android OS.

3.1.2. Problem Definition:

Our main aim is to equip the user with certain settings which he/she can use according to its changing device usages,so that the device always has enough charge to get the user through the day without needing to charge the device every now and then.

**3.2 Requirement Analysis**

Requirement Analysis provides the software designer with the models that can be translated

into data, architectural, interface and procedural designs.

3.2.1. Requirement Specifications

*3.2.1.1. Goals:*

The goal of this project is to develop anandroid application that can be installed into the device,providing effective power utilization.

*3.2.1.2. Hardware and Software Requirements:*

*Hardware Requirements:*

For Developing android application,the Android Studio requires:

### Windows

* Microsoft® Windows® 8/7/Vista/2003 (32 or 64-bit)
* 2 GB RAM minimum, 4 GB RAM recommended
* 400 MB hard disk space
* At least 1 GB for Android SDK, emulator system images, and caches
* 1280 x 800 minimum screen resolution
* Java Development Kit (JDK) 7
* Optional for accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality

### Mac OS X

* Mac® OS X® 10.8.5 or higher, up to 10.9 (Mavericks)
* 2 GB RAM minimum, 4 GB RAM recommended
* 400 MB hard disk space
* At least 1 GB for Android SDK, emulator system images, and caches
* 1280 x 800 minimum screen resolution
* Java Runtime Environment (JRE) 6
* Java Development Kit (JDK) 7
* Optional for accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality

On Mac OS, run Android Studio with Java Runtime Environment (JRE) 6 for optimized font rendering. You can then configure your project to use Java Development Kit (JDK) 6 or JDK 7.

### Linux

* GNOME or KDE desktop
* GNU C Library (glibc) 2.15 or later
* 2 GB RAM minimum, 4 GB RAM recommended
* 400 MB hard disk space
* At least 1 GB for Android SDK, emulator system images, and caches
* 1280 x 800 minimum screen resolution
* Oracle® Java Development Kit (JDK) 7

*Software Requirements*

Development tools

JDK 6 (JRE alone is not sufficient)

• Android SDK (Linux / Windows version)

• Android Development Tools (ADT plug-in for eclipse)

• Android Emulator

* + - 1. 3.2.1.3 Functional and Non- functional Requirements
      2. *Functional Requirements should include:*
* The User has to provide the parameters to the application .
* The User has to observe the output to decide future changes are required or not.

*Non-functional requirements*

1. Usability:

This application should provide a good interface so that the user finds it easy to use.

1. Performance:

Speed requirements

The application should introduce the changes quickly enough so that the user can get back to using the device.

Safety critical requirements

The application should introduce changes,without freezing or heating up the device, damaging the battery or reducing its lifetime.

1. Reliability :

The application should be reliable and should not crash.

1. Security requirements:

The application gets it superuser access from super su application and has permissions for hardware controls and system tools for starting at startup.

3.2.1.4. Assumptions & Dependencies

User is acquainted with the android enable device and with the certain terminologies.

The user can understand the battery graph and information provided in the battery option of the settings menu.

**3.3 Software Process Model**

3.3.1. Model Used :

The waterfall model is a sequential design process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance.

3.3.2. Description

• Requirements:

Initiative step in which requirements are gathered by

communicating with the end users. It involves preparing estimation of user requirements on a whole and providing the enhancements out of the box to meet requirements of as much large portion of users on a whole.

• Analysis and Design: It includes analysis and design of software. It starts with designing creation of applications according to the set of requirements.

• Coding: It includes coding and implementation of the project.

• Testing: The application thus created is installed on the testing device and tested under different combinations of frequencies,governers to find the best combination.

* Maintenance: It includes delivery of software and also the feedback from the end user. The mentor of the project would be providing us with valuable suggestions and evolving needs so that they could be accommodated into the project.

3.3.3 Advantages & Disadvantages

* Advantages
* Simple and easy to use.
* Each phase has specific deliverables.
* Works well for the small projects where requirements are easily understood.

Diadvantages

* Very Rigid.
* Little flexibility and adjusting scope is difficult and expensive.
* Software is developed during the implementation phase, so no early prototypes
* of software are produced.

3.3.4 Why this model is used?

• It is very simple.

• It is linear in nature.

• Testing in inherent to every phase of the waterfall model.

• It is an enforced discipline approach.

• It is documentation driven, that is, documentation is produced at every stage.

**3.4 Planning Phase**

Project Resources

**ResourcesRequirements**

**CPU-**64 bit, 2.50 GHz dual core

**OS-**Windows

**Memory-**4 GB for developer and evaluation use

**Disks-**512 GB

**Android SDK-**Android SDK 1.0.7

**Java-**Java 7

**JRE-**JRE 1.7.1

**Testing Device-**Samsung Galaxy S Duos s7562

3.4.2. Project team organization

The team consists of a single member. The member has both formal and informal relationship with the project guide . The top level problem solving is guided by the project guide.

**3.5 Risk Analysis**

3.5.1. Definition

Risk analysis is the method used to identify the potential threats and risks to the system and thus, try to mitigate the same in order to make the system development process reliable and minimize the uncertainty involved. The risks are identified, accessed for impacts and henceforth methods are used to lessen it.

3.5.2. Project Risk analysis & Management

The risks are directly related with the project and its development. Thus, it involves risk in the implementation of the project are schedule, resources and complexity problems.

*Schedule:* Limited time of one year’s academic session that might be insufficient for completion of project along with curriculum.

*Resources:* College’s computer is assigned to each computer with limited specifications and member's laptop,cell phone and tablet.

*Technical Risks:* These are the risks that involve the design, implementation, interface and maintenance threats to the software. Thus, it takes into account the reliability of the project and its security.

3.6 Complete Analysis Model

In analysis model, the approach used is object oriented model. Such an approach models a system as a group of interacting objects, that is , a problem Is represented by objects which are characterized by its class, its state and its behavior. Object oriented analysis focuses on what the system does and not on how it does that. The objective of object oriented analysis is to develop a model that describes software as it works to satisfy a set of customer defined requirements. The deliverable result at the end of this phase is the required document. Ideally this document states in clear and precise fashion what is to be built. The document tries to capture the requirements from the customer’s perspective by defining the goals and interactions at a level removed from implementation details. The OOA begins with the definition of the use case scenarios that describe how a system is to be used.

**DESIGN AND IMPLEMENTATION**

**4.1 Design Models**

4.1.1 Use-case Diagram



4.1.2 Activity Diagram



4.1.3 Data Flow Diagram

Level 0



**TESTING**

**5.1 Testing Objectives**

Our objective is to design tests that systematically uncover different classes of errors and to do so with minimum amount of time and effort.

A number of rules that can serve well as testing objectives:

* Testing is a process of executing a program with the intent of finding an error.
* A good test case is one that has a high probability of finding an as-yet undiscovered error.

A secondary benefit of testing is that it demonstrates that the software appears to be working as stated in the specifications.The data collected through testing can also provide an indication of the software’s reliability and quality.

**5.2 Testing principles**

Before applying methods to design effective test cases, a software engineer must understand the basic principles that guide software testing.

* All tests should be traceable to customer requirements.As we have seen,the objective of software testing isto uncover errors.It follows that the most severe defects from the customer’s point of view are those that cause the program to fail to meet its requirements.
* Tests should be planned long before testing begins.Test planning can begin as soon as the requirements model is complete.Detailed definition of test cases can begin as soon as design model has been solidified.Therefore, all tests can be planned and designed before any code hs been generated.
* The pareto principle applies to software testing.Stated simply,the pareto principle implies that 80 percent of all errors uncovered during testing will likely be traceable to 20 percent of all program components.The problem,of course,is to isolate these suspect components and to thoroughly test them.
* Testing should begin “in the small” and progress toward testing “in the large”.The first tests planned andexecuted generally focus on individual components.As testing progresses,focus shifts in an attempt to find errors in integrated clusters of components and ultimately in the entire system.
* Exhaustive testing is not possible.The numbr of path permutations for even a moderately sized program is exceptionally large for this reason,it is impossible to execute every combination of paths during testing.it is possible, however ,to adequately cover program logic and to ensure that all conditions in the component level design have been exercised.
* To be more effective,testing should be counducted by an independent third party.By most efctive,we mean testing that has the highest probability of finding errors the primary objective of testing.

**5.3 Testing Methods**

5.3.1 Black Box Testing

Black Box Testing focuses on functional requirements of the software.That is Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.Black Box testing attempts to find erros on following categories:-

* Incorrect missing functions.
* Interface errors.
* Errors in data structures or external database access.
* Performance errors.

5.3.2 Unit Testing

Unit Testing is a level of software testing process where individual units/components of a software/system are tested.The purpose is to validate that each unit of software performs as designed.A unit is smallest testable part of software.It usually has one or few inputs and usually a single output.In procedural programming a unit maybe an individual program, function, procedure etc.In object oriented programming,the smallest unit is a method,which may belong to a base/super class, abstract class of derived/child class.It is white box oriented and can be conducted in parallel for multiple components.Following are modules we tested individually to ensure that the information properly flows in and out of program unit:

* Graphics loading module.
* Frame rendering module
* Game runtime control module

5.3.3 Integration Testing

Integration testing is the testing process in software testing process in software that when two or more modules are interact and produced result satisfies with tis original functional requirement or not.Integrated testing is fall in Black box testing.Integrated testing will start afte completion of unit testing.Integration testing finds the bugs that occur when two or more model integrated.Main purpose of integrated testing is to indentifying the functional ,requirement and performance level bugs.When modules not integrated, they perform as per requirement but when they integrated, functional, requirement and performance related issues will occurs due to the integration.

5.3.4 System Testing

System Testing tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets Quality Standards. This type of testing is performed by specialized testing team. System Testing is the first step in the Software Development Life Cycle, where the application is testes as whole. The application is tested thoroughly to verify that it meets the functional and technical specifications.The application is tested in an environment which is very close to the production environment where the application will be deployed. System testing enables us to test,verify and validate both the business requirements as well as the Applications Architecture.

5.3.5 Acceptance Testing

Acceptance testing, a test technique performed to determine whether or not the software system has met the requirement specifications. The main purpose of this purpose of this test is to evaluate the system’s compliance with business requirements and verify if it has met the required criteria for delivery to end users. The acceptance test of our project is done by our Respected teachers and our classmates and friends and they are satisfied with our performance.

* Some of the Test Cases are:
* Whether the user is able to access the app easily or not.
* User is getting quick response or not.
* Whether emulator works properly or not.

**CONCLUSION**

`

The developed application will help the user to get satisfactory performance while using minimal power from the device. The application provides significant battery boost over the conventional usage, , also the user can revert back the changes to original settings as well. The application is successfully deployed on one android device and and gives satisfactory results

**REFERENCES**

* **http://www.stackoverflow.com**
* **http://www.developer.android.com**
* **https://www.youtube.com/channel/UCB4dWKbwgl3U3ohBmrT7k3w**
* **http://coderanch.com/**