

EMPOWERING PARENTS TO NAVIGATE THE DIGITAL LANDSCAPE

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

**ROHITH KUMAR A(Reg.No - 40110068)
PARITOSH ATTKRTI (Reg.No – 40110905)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade “A++” by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHISALAI,

CHENNAI – 600119

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **Paritosh A(40110905)**
And **Rohith Kumar Allam(40110068)** who carried out the project entitled
"Empowering Parents To Navigate The Digital Landscape" under my supervision
from November 2023 to April 2024.

Internal Guide

Dr. SREEKRISHNA M. M.E., Ph.D.

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

HEAD OF THE DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Submitted for Viva voce Examination held on 15/04/2024
Chennai - 600 119

Internal Examiner
External Examiner

DECLARATION

I, Paritosh Attketi(40110905), hereby declare that the Project Report entitled "Empowering Parents To Navigate The Digital Landscape" done by me under the guidance of Dr. SREEKRISHNA M.M.E., Ph.D, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering

DATE: 15/04/2024

PLACE: Chennai

Paritosh A
SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **The Board of Management of Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. SREEKRISHNA M. M.E. Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Today's rapidly advancing technological landscape, addressing child security concerns has become paramount. This model aims to alleviate parental worries by ensuring maximum security and live tracking for their children. Our system guarantees real-time monitoring, sending parents updates on their child's recent SMS and calls. This proactive approach empowers parents to safeguard their children from potential threats. Additionally, our model offers a solution for emergencies, allowing children to quickly send a message with their current location via Short Message Services (SMS). Validation of this system was conducted through rigorous testing on the Android platform, ensuring its reliability and effectiveness. Furthermore, our model incorporates advanced features to enhance child safety and parental peace of mind. Alongside live tracking and SMS updates, it includes geofencing capabilities, allowing parents to set virtual boundaries. If a child crosses these predefined boundaries, instant alerts are sent to the parent's device, enabling swift action in case of unauthorized movements. Moreover, the system utilizes machine learning algorithms to analyze incoming messages and calls for potential threats, providing an added layer of security. Suspicious or concerning communications trigger immediate notifications to parents, allowing them to intervene promptly. In addition to real-time monitoring, the system maintains a historical log of the child's activities, providing parents with insights into their daily routines and whereabouts. This comprehensive approach ensures that parents have the tools and information needed to protect their children effectively in today's digital age. Testing on the Android platform validates the reliability and functionality of our proposed system, offering a practical solution for child safety concerns. Ultimately, our model strives to offer comprehensive and proactive solutions, empowering parents to safeguard their children's well-being in an increasingly complex world. By prioritizing child security through innovative technological solutions, we aim to provide parents with peace of mind and ensure the safety of their children in today's dynamic and ever-evolving digital environment.

TABLE OF CONTENTS

Chapter No	TITLE	Page No.
	ABSTRACT	v
	LIST OF FIGURES	vii
	LIST OF ABBREVIATIONS	Viii
1	INTRODUCTION	1
2	LITERATURE SURVEY	3
	2.1 Inferences from Literature Survey	3
3	SYSTEM ANYLISIS	
	3.1 Existing System	5
	3.2 Proposed System	5
4	REQUIREMENT ANALYSIS	
	4.1 Function and non-functional requirements	6
	4.2 Hardware Requirements	7
	4.3 Software Requirements	7
	4.4 Architecture	7
5	SYSTEM DESIGN	
	5.1 Introduction of Input design	8
	5.2 UML Diagram	9
	5.3 Data Flow Diagram	15
6	IMPLEMENTATION AND RESULTS	
	6.1 Modules	16
	6.2 Software Installation	16
	6.3 Software Development Life Cycle	27
7	SYSTEM STUDY AND TESTING	
	7.1 Feasibility study	56
	7.2 Types of test & Test Cases	57
8	CONCLUSION	60
9	REFERENCES	61

LIST OF FIGURES

FIGURE NO	FIGURE NAME	Page No.
4.4	Flow of Application	7
5.2.1	Use Case Diagrams	10
5.2.2	Class Diagram	11
5.2.3	Sequence Diagram	12
5.2.4	Activity Diagram	13
5.3.5	Data Flow Diagram	15
6.2	Software Installation	16
6.3	Software Development Life Cycle	28
6.3	Android Architecture	31
6.3	Android Operating System	39

LIST OF ABBREVIATIONS

SMS	-	Short Message Service.
GPS	-	Global Positioning System
JDK	-	Java Development Kit
SDK	-	Software Development Kit

CHAPTER 1

INTRODUCTION

1.1 Motivation

In today's rapidly advancing technological landscape, child security is paramount for parents. Our innovative system alleviates parental concerns by offering real-time tracking of their children, automatically sending last SMS and call details to parents, and enabling children to send emergency messages with their current location via SMS. This Android-based solution provides peace of mind and enhances child safety.

1.2 Problem Statement

The problem at hand revolves around addressing parental concerns for child safety in an increasingly connected world. Parents worry about their children's security, especially when they use smartphones. The challenge is to develop a robust system that offers real-time tracking, automatic communication of call and SMS logs, and emergency messaging capabilities to enhance child safety and ease parental worries.

1.3 Objective of the Project

The primary objective of this project is to develop and implement a comprehensive child safety system on smartphones, aimed at providing parents with real-time tracking, automatic communication of call and SMS logs, and emergency messaging features. This system aims to enhance child security while offering peace of mind to parents.

1.4 Scope

The scope of this project encompasses the design, development, and implementation of a child safety system for smartphones. It includes the creation of a user-friendly mobile application for parents and children, real-time GPS tracking, automated SMS and call log sharing, and an emergency messaging feature. Testing and validation will be conducted on the Android platform, ensuring a robust and reliable solution for child security.

1.5 Project Introduction

Today smart phones are the basic need of the user today, these smart phones, providing lots of features which make our life so simple and easier. This paper is focused with the safety of children. Today child safety is an important issue.

the world as child crime is rapidly growing across the world in this paper we have discussed how a smart phone provides safety and monitoring for the parents so that they can easily track their children according to their requirement. This proposed application is developed on android platform for this application the basic techniques required mentioned this represents the android application uses GPS and SMS services and GPS help in locating the child's location by using Google maps. GPS is useful for tracking child and also provides the information where the child is currently located as well as it also informs the parents how long his child is far away from his parents. SMS services used when smart phones do not support internet connectivity in this case child is able to send a text message or exact location in the parents. This system is going to help the parents to track the location of their children without informing them because their movement is displayed on the parent device through Google maps as well as they received calculated distance of their child from themselves. This application is also helpful for girls mostly studying or doing a job from far away from their home. In case of any emergency on just one click or shake their mobile they are able to send their current location via SMS to their parents. In this application parents are able to create a Geo-reference boundary according to their choice called Geo-fencing, at a single time multiple Geo-fences can be created. This application uses Google maps API to show location on map. Although the parents can get the sms's and calls log were be displayed to parent according to how many children in this usage the Kids can get much safer from the threaten calls.

CHAPTER 2

LITERATUR SURVEY

2.1 INFERENCES FROM LITREATURE SURVEY

Geo-Fencing,available at: <http://www.fieldtechnologiesonline.com>

A geofence is a virtual perimeter for a real-world geographic area.[1] A geo-fence could be dynamically generated—as in a radius around a point location, or a geo-fence can be a predefined set of boundaries (such as school zones or neighborhood boundaries). The use of a geofence is called geofencing, and one example of usage involves a location-aware device of a location-based service (LBS) user entering or exiting a geo-fence. This activity could trigger an alert to the device's user as well as messaging to the geo-fence operator. This info, which could contain the location of the device, could be sent to a mobile telephone or an email account.

Aditi Gupta; Vibhor Harit; Child Safety & Tracking Management System by Using GPS, Geo-Fencing & Android Application: An Analysis; 12-13 February 2016

Today, technology is growing rapidly and providing all essential and effective solutions for every requirement. Nowa day's child security is an important area of concern. This model is developed to rectify the worries of parents regarding their child security. In this scenario, Our system ensures maximum security and ensures live tracking for their kids because parent worries are genuine. This paper proposeda model for child safety through smart phones that provides the option to track the location of their children as well as in case of emergency children is able to send a quick message and its current location via Short Message services. This proposed system is validated by testing on the Android platform.

Child Safety & Tracking Management System

Life would have been like a walk in a park without technology. But that is not the case as with each day the technology is only getting improved. And with the growing technology the concern towards an individual's safety is also growing. Child safety is one such matter of concern in today's world. In this paper we discuss majorly on child safety, here we make use of GPS and wireless networking techniques. Mobile phones have been a major influence on human life but the effect of it varies for an adult and child. So here with the help of digital jewelry.

phone which although is not a complete smart phone but consist of the all the applications that are required. Apart from this we have made use of GPS, Geo-fencing, SMS techniques to track the child's location and to inform about it via telephony communication.

Harshal Chaudhary; Dr. Ranjana Zinjore; Dr. Varsha Pathak ;Parent-Hook: A Child Tracking System based on Cloud URL; 30-31 October 2020

Child kidnapping, missing child and child harassment are the world-wide problem related child safety. The children of age group 4 to 8 years are innocent and subject to kidnapping in frequent cases. Parents are always worried regarding their children's security mainly when they visit crowded public places and travel in widely physically located places. Number of applications is being developed to guard children in every manner. This paper introduces a Parent-Hook product which is designed for child tracking if the child is lost. The Parent-Hook is a safety band without the sensor or any chip harmful for the children can be put on the whist of the children. This band is easy to carry which is made up of soft cotton webbing with parent contact information with QR Code and Cloud URL.

N. Senthamilarasi, N.Divya Bharathi, D.Ezhilarasi, R.B.Sangavi; Child Safety Monitoring System Based on IoT; 23/09/2023

The overall percentage of child abuselements filed nowadays in the world is about 80%, out of which 74% are girl children and the rest are boys. For every 40 seconds, a child goes missing in this world. Children are the backbone of one's nation, if the future of children was affected, it would impact the entire growth of that nation. Due to the abuselements, the emotional and mental stability of the children gets affected which in turn ruins their career and future. These innocent children are not responsible for what happens to them. So, parents are responsible for taking care of their own children. But, due to economic condition and aims to focus on their child's future and career, parents are forced to crave for money. Hence, it becomes difficult to cling on to their children all the time. In our system, we provide an environment where this problem can be resolved in an efficient manner. It makes parents to easily monitor their children in real time just like staying beside them as well as focusing on their own career without any manual intervention

CHAPTER 3

SYSTEM ANALYSIS

3.1 Existing System:

Child safety and tracking is a major concern as the more number of crimes on children are reported nowadays. With this motivation, a smart IoT device for child safety and tracking is developed to help the parents to locate and monitor their children. The system is developed using LinkIt ONE board programmed in embedded C.

Disadvantages

- The child could not produce the exact alert command during a panic condition.
- The command produced may not match with the previously stored command.
- This project requires manual intervention

3.2. Proposed System

This application is designed for locating missing children. The solution represented in this app takes the advantages of smart phones which offer rich features like Google maps, GPS, SMS etc. Some of the best works implemented in past relies on SMS based tracking which is not helpful to get an accurate location in our proposed system we have provided real time tracking. We have added Geo-fencing and Emergency messaging services to enhance the system and whenever children login's into the app the parent can get whatever sms received and whatever calls were did by children's were sent to parent and they can view according to there chidrens's Here we are going to implement like, children need to maintain the phone and our app, and then it can be possible with android environment.

Advantages

- We can get the data from cloud so it is fast and accurate.
- Through mobile GPS we can get the location exactly.
- SMS service to get the location message to device.
- Safe guard from unknown threats.

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Function and non-functional requirements **Functional and non-functional requirements:**

Requirement's analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and non-functional requirements.

Functional Requirements: These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Examples of functional requirements:

- Authentication of user whenever he/she logs into the system
- System shutdown in case of a cyber-attack.

Non-functional requirements: These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.

They basically deal with issues like:

- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Reusability
- Flexibility

Examples of non-functional requirements:

- Emails should be sent with a latency of no greater than 12 hours from such an activity.
- The processing of each request should be done within 10 seconds.

4.2 Hardware Requirements

- **H/W System Configuration: -**
- Processor - I3/Intel Processor
- RAM - 8 GB
- Hard Disk - 1TB

4.3 Software Requirements

- Operating System - Windows 10
- JDK - java
- Plugin - Kotlin
- SDK - Android
- IDE - Android studio
- Database` - server script, my sql

4.4 Architecture

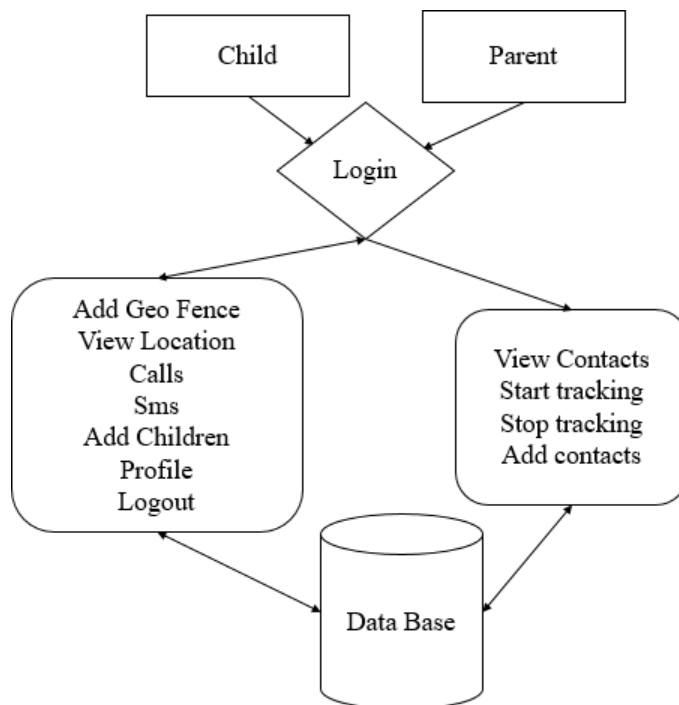


Figure 4.4: Flow of Application

CHAPTER 5

SYSTEM DESIGN

5.1 Introduction of Input design INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

- Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
- It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
- When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

- Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
- Select methods for presenting information.
- Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

5.2 UML Diagram

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: A Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modelling language.
- Encourage the growth of OO tools market.
- Support higher level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

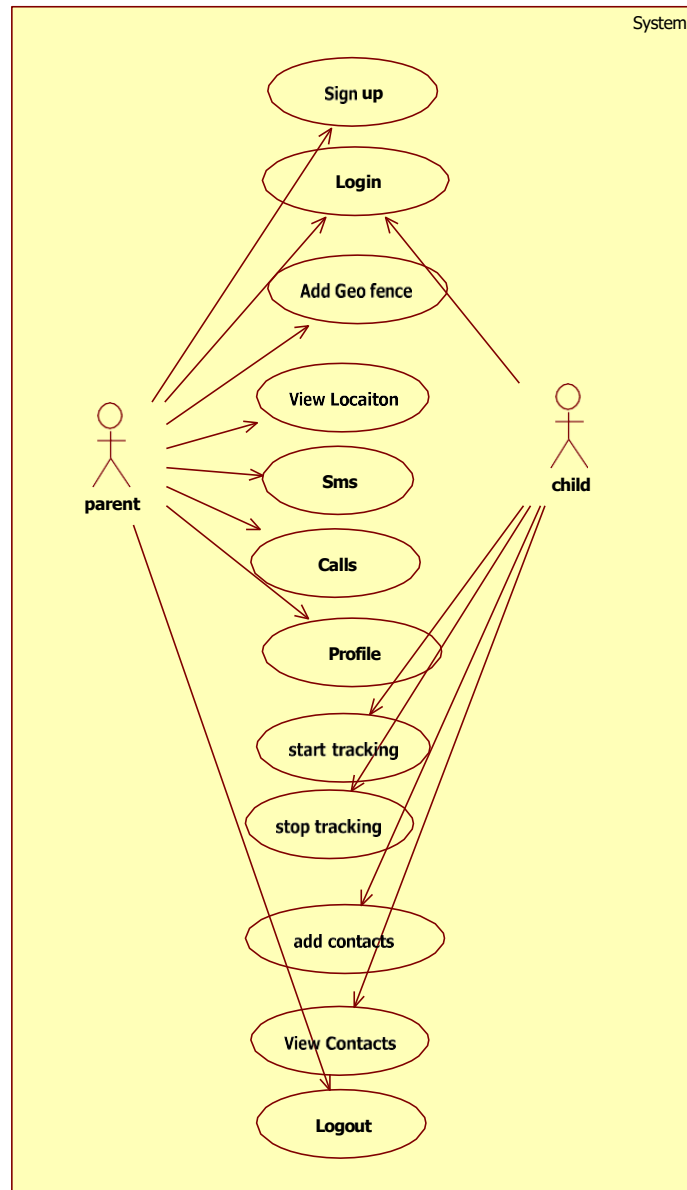


Figure 5.2.1:Use Case Diagram

CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

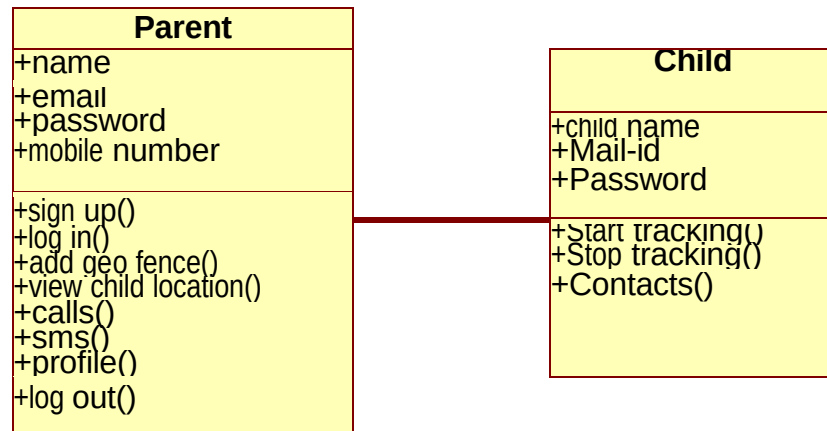


Figure 5.2.2: Class Diagram

SEQUENCE DIAGRAM:

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

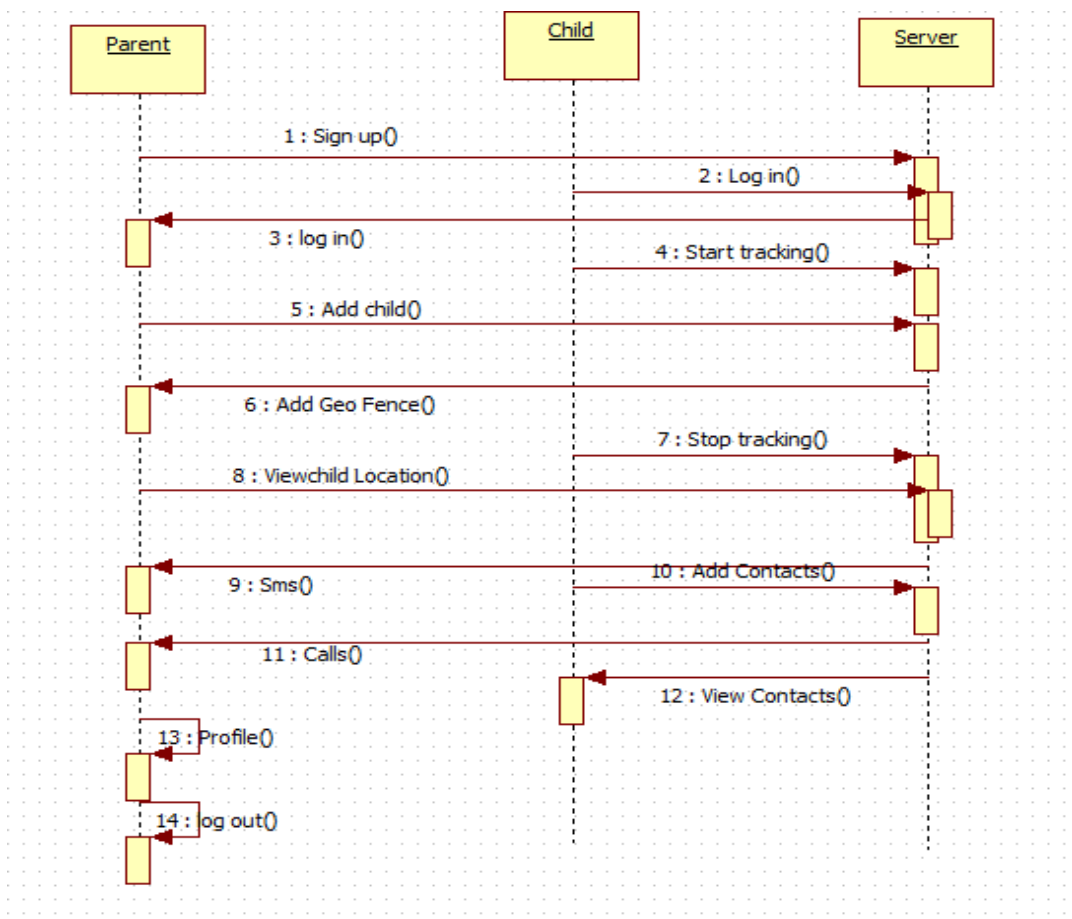


Figure 5.2.3: Sequence Diagram

COLLABORATION DIAGRAM:

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

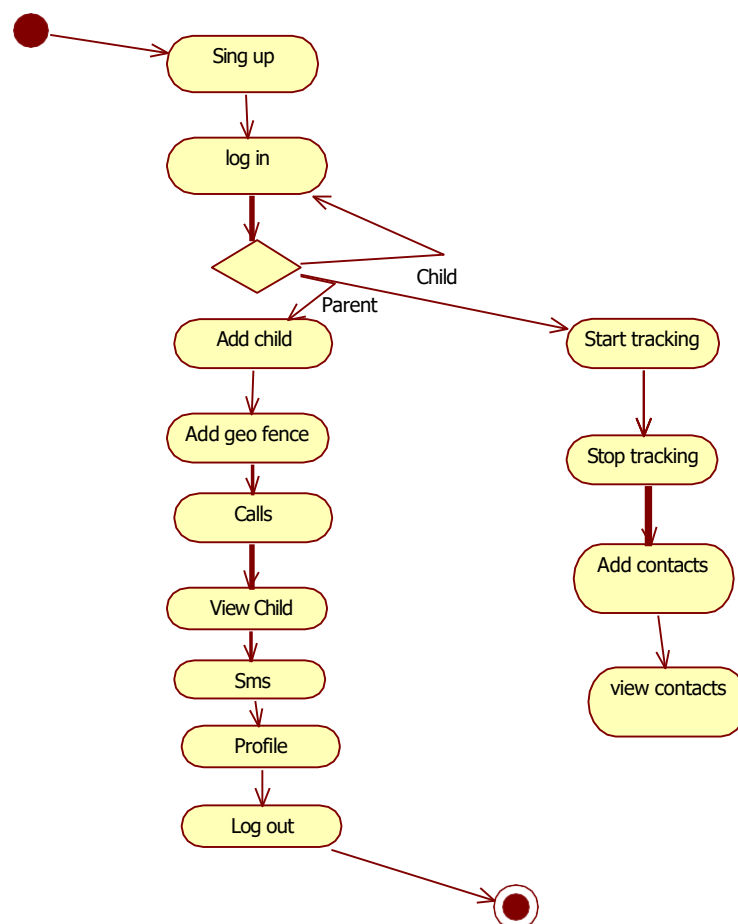


Figure 5.2.4: Activity Diagram

COMPONENT DIAGRAM:

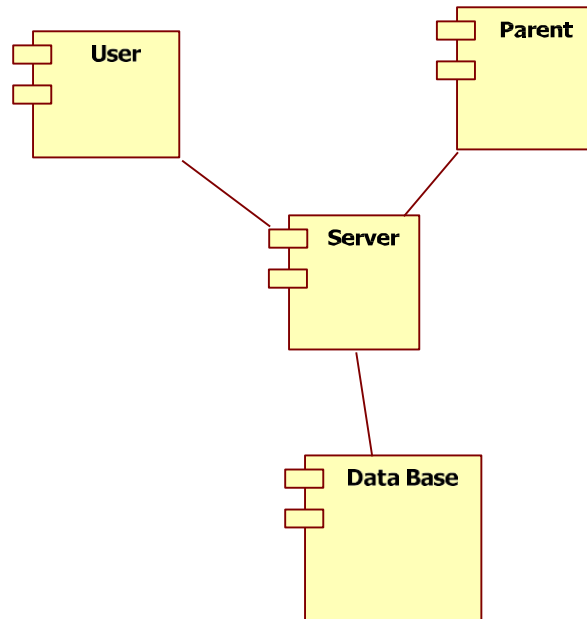


Figure 5.2.5: Component Diagram

DEPLOYMENT DIAGRAM:

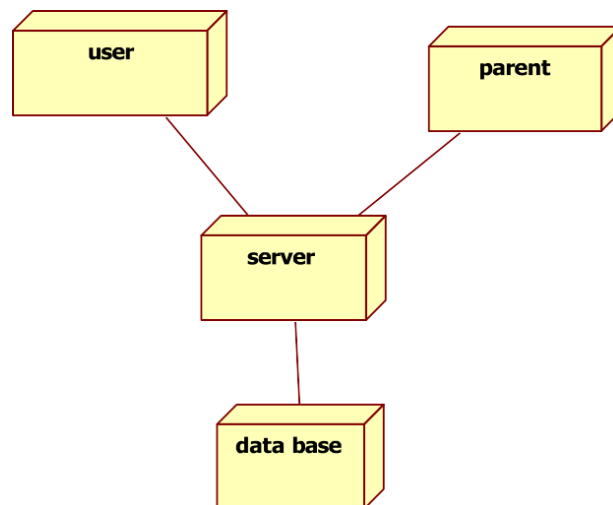


Figure 5.2.6:Deployment Diagram

ER Diagram:

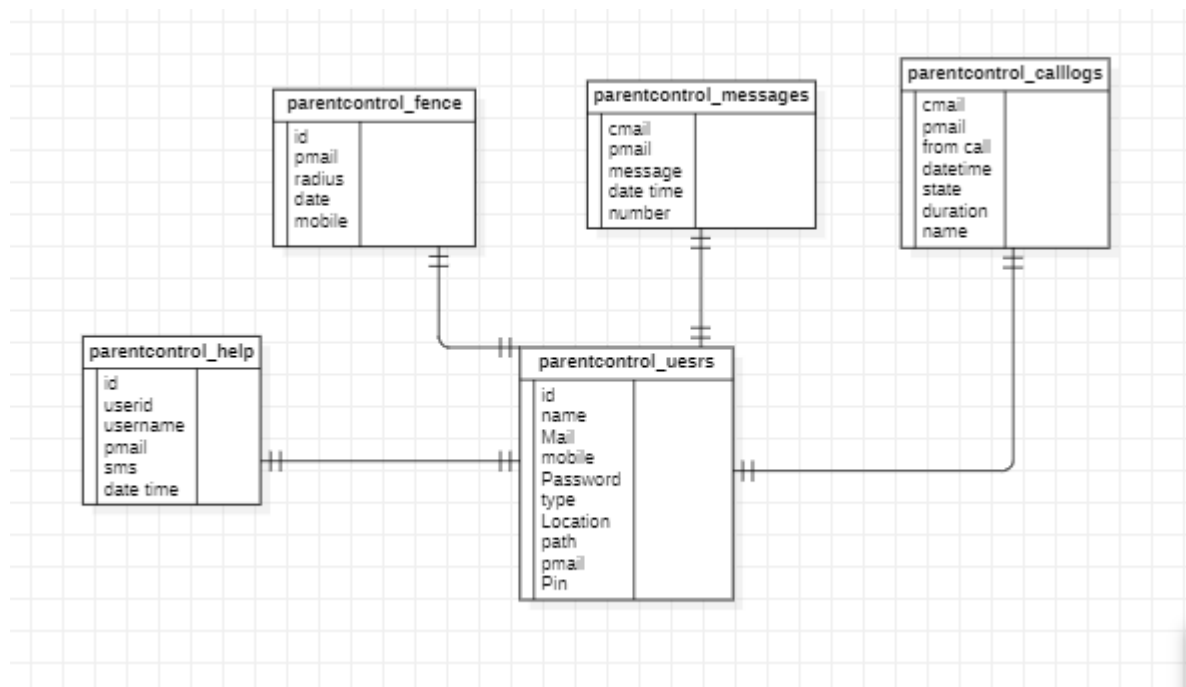


Figure 5.2.7: E R Diagram

5.3 Data Flow Diagram:

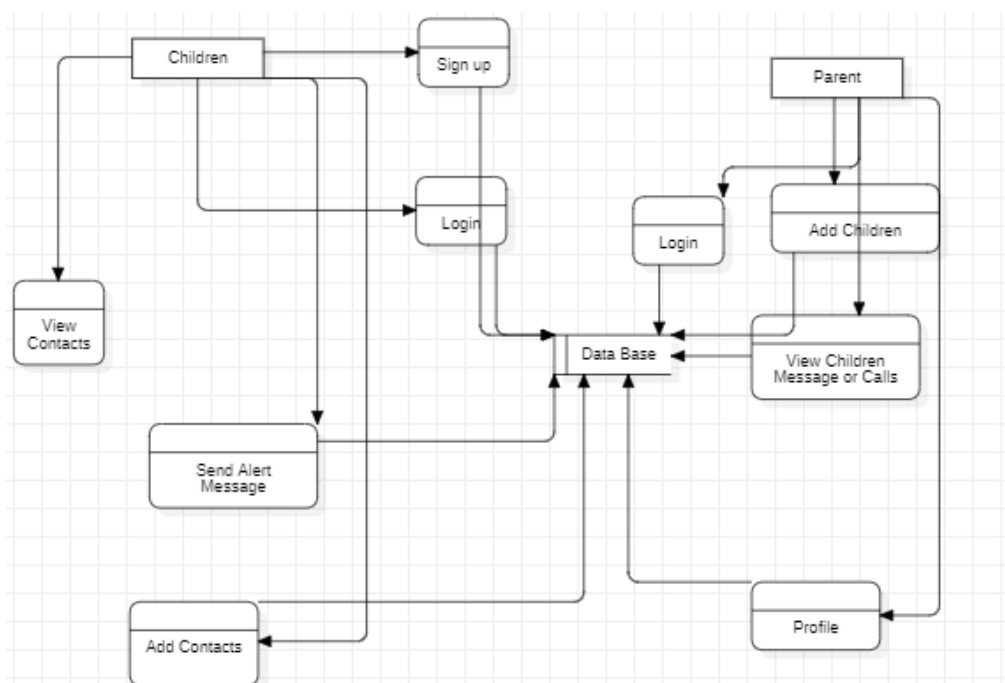


Figure 5.3: Data Flow Diagram

CHAPTER 6

IMPLEMENTATION AND RESULTS

6.1 Modules Parent:

Parents can sign up and log in to their accounts, adding their children's profiles. They can monitor their children's messages and call logs. Parents also have the option to establish geo-fenced areas for their children. If the child crosses these boundaries, the parent receives instant notifications.

User:

Users can input emergency contacts, access and manage the contact list, and initiate calls to these designated contacts when necessary.

ANDROID ENVIRONMENT

6.2 Software Installation Software Installation of JDK kit

This Java Development Kit (JDK) allows you to code and run Java programs. It's possible that you install multiple JDK versions on the same PC. But it's recommended that you install only latest version.

How to install Java for Windows

Following are the steps for JDK 8 free download for 32 bit or JDK 8 download 64 bit and installation

Step 1) Go to [link](#) Click on JDK Download for Java

Java SE 8

Java SE 8u271 is the latest release for the Java SE 8 Platform.

- [Documentation](#)
- [Installation Instructions](#)
- [Release Notes](#)
- [Oracle License](#)
 - [Binary License](#)
 - [Documentation License](#)
 - [BSD License](#)
- [Java SE Licensing Information User Manual](#)
 - [Includes Third Party Licenses](#)
- [Certified System Configurations](#)
- [Readme Files](#)
 - [JDK ReadMe](#)
 - [JRE ReadMe](#)

Oracle JDK

-  [JDK Download](#)
-  [Server JRE Download](#)
-  [JRE Download](#)
-  [Documentation Download](#)
-  [Demos and Samples Download](#)



Figure:6.2.1: JDK Download

Step 2) Next,

1. Accept License Agreement
2. Download Java 8 JDK for your version 32 bit or JDK 8 download for windows 10 64 bit.

Solaris SPARC 64-bit	88.75 MB	 jdk-8u271-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	134.42 MB	 jdk-8u271-solaris-x64.tar.Z
Solaris x64	92.52 MB	 jdk-8u271-solaris-x64.tar.gz
Windows x86	154.48 MB	 jdk-8u271-windows-i586.exe
Windows x64	166.79 MB	 jdk-8u271-windows-x64.exe

Figure:6.2.2:Windows Version

Step 3) when you click on the Installation link the popup will be open. Click on I reviewed and accept the Oracle Technology Network License Agreement for Oracle Java SE and you will be redirected to the login page. If you don't have an oracle account you can easily sign up by adding basics details of yours.

You must accept the [Oracle Technology Network License Agreement for Oracle Java SE](#) to download this software. ✕

☒ I reviewed and accept the Oracle Technology Network License Agreement for Oracle Java SE

You will be redirected to the login screen in order to download the file.

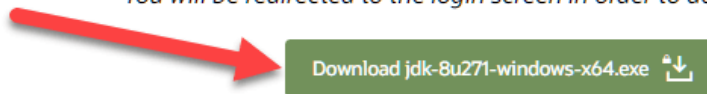


Figure 6.2.3: Download The JDK

Step 4) once the Java JDK 8 download is complete, run the exe for install JDK.

Click Next

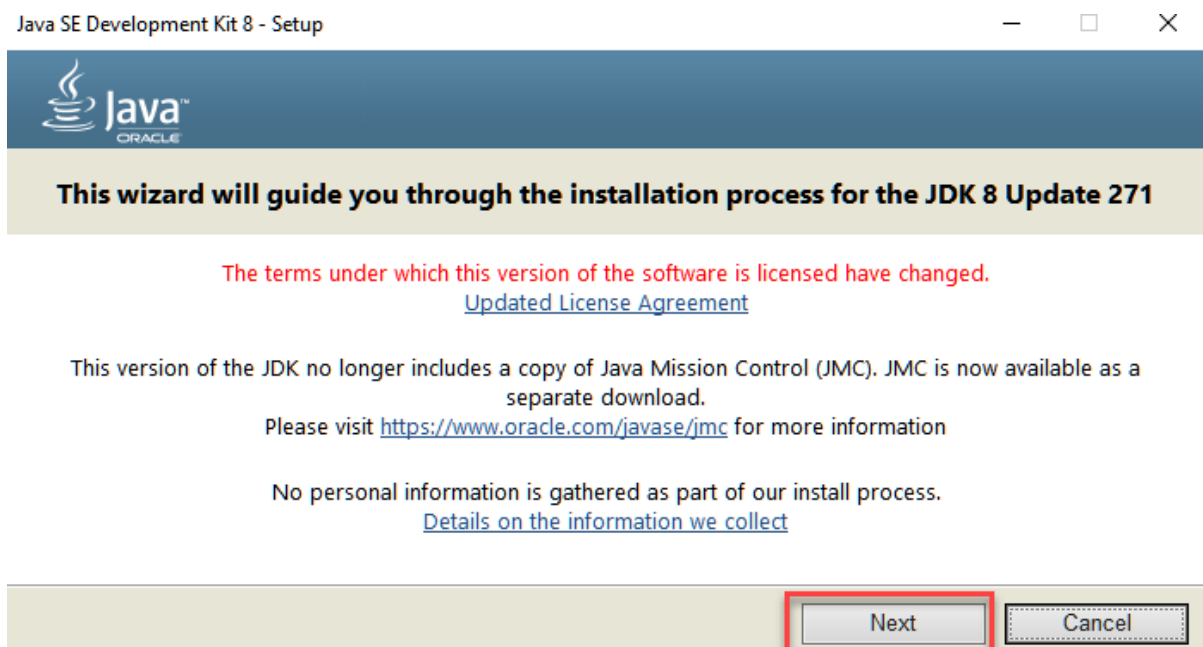


Figure:6.2.4:Installation

Step 5) Select the PATH to install Java in Windows and click next.

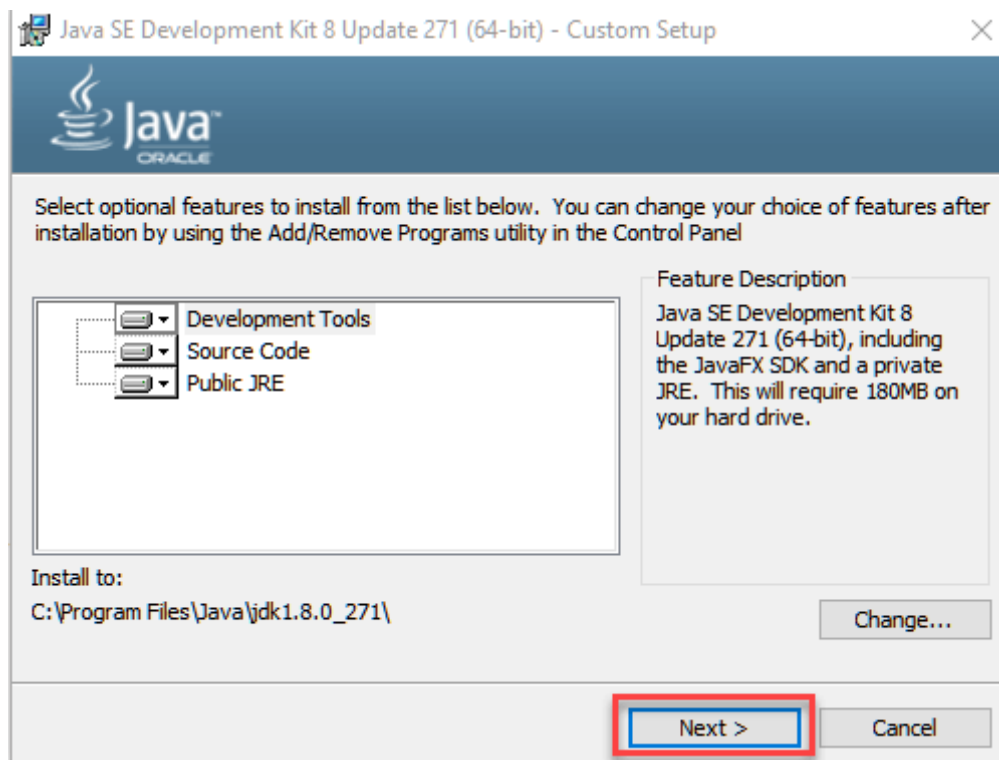


Figure:6.2.5:Installation Step 1

Step 6) Once you install Java in windows, click Close



Figure:6.2.6:Installation Step 2

How to set Environment Variables in Java: Path and Class path

The PATH variable gives the location of executable like javac, java etc. It is possible to run a program without specifying the PATH but you will need to give full path of executable like **C:\Program Files\Java\jdk-13.0.1\bin\javac A.java** instead of simple **javac A.java**

The CLASSPATH variable gives location of the Library Files. Let's look into the steps to set the PATH and CLASSPATH

Step 1) Right Click on the My Computer and Select the properties

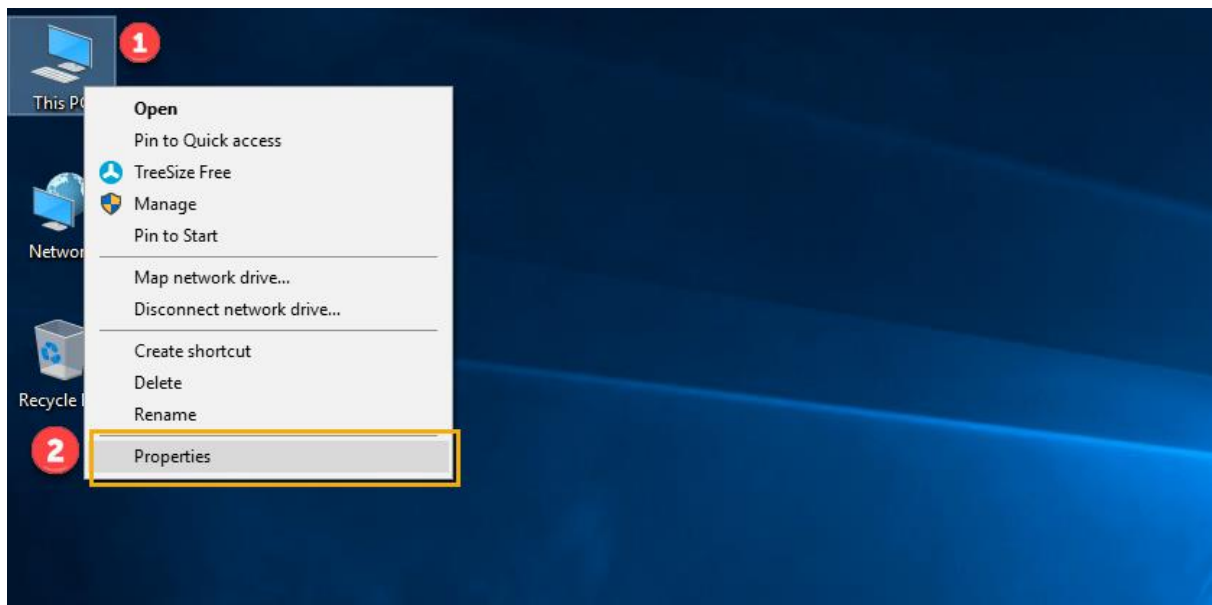


Figure:6.2.7:Installation Step 2

Step 2) Click on advanced system settings

Step 3) Click on Environment Variables

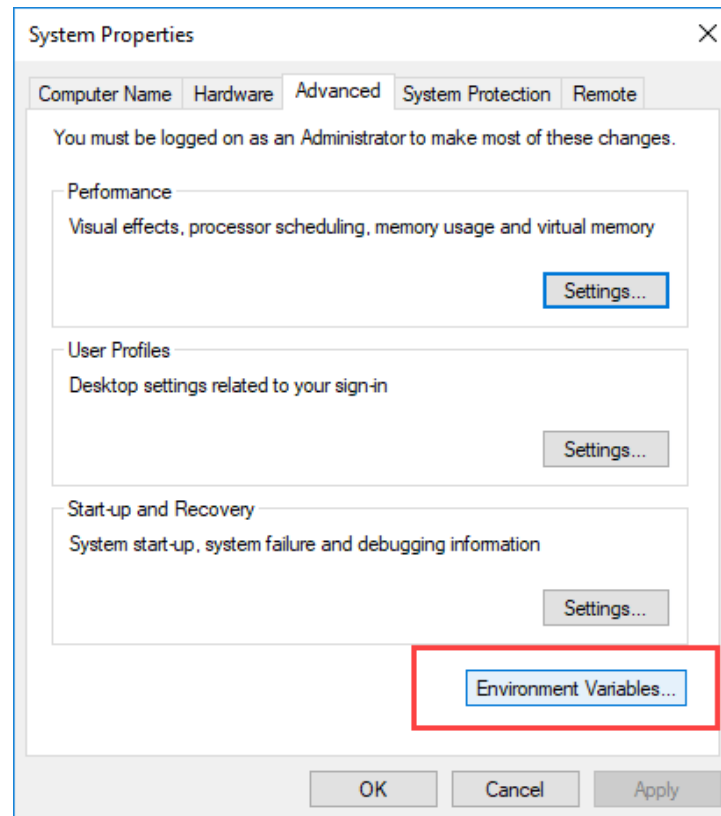


Figure:6.2.6:Installation Step 3

Step 4) Click on new Button of User variables

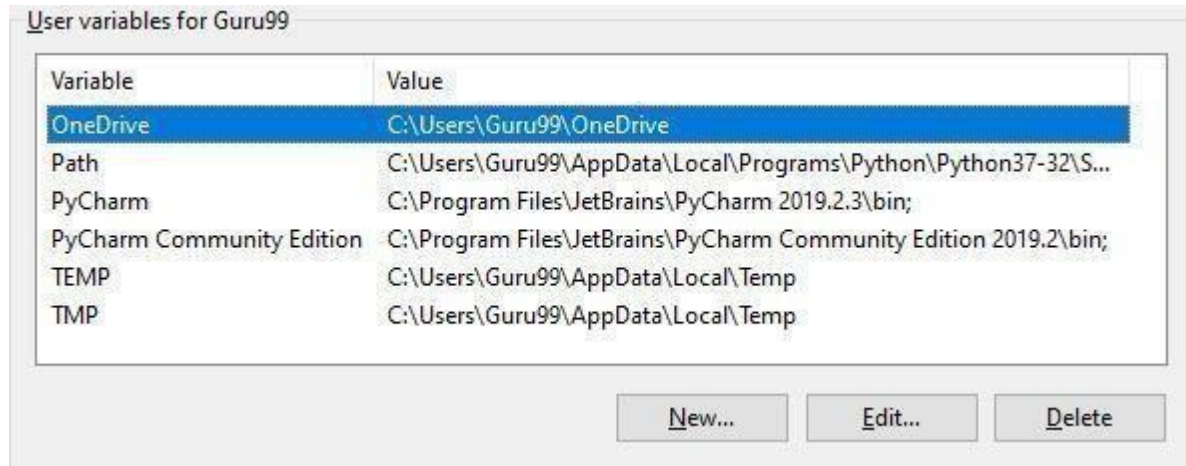


Figure:6.2.7: User variables

Step 5) Type PATH in the Variable name.

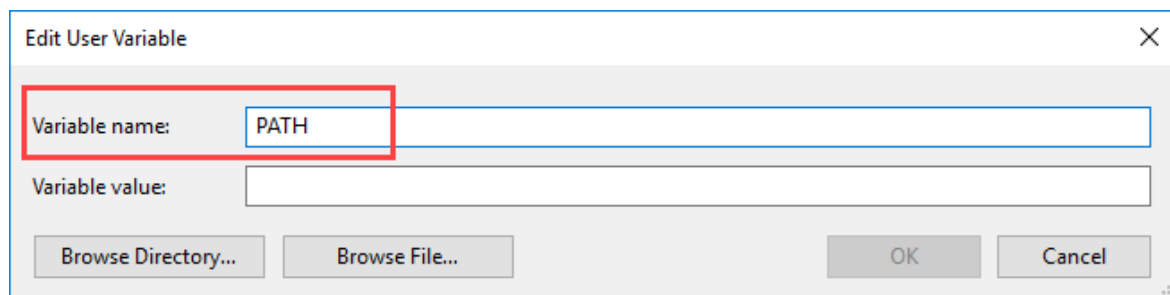


Figure:6.2.8: Variable name

Step 6) Copy the path of bin folder which is installed in JDK folder.

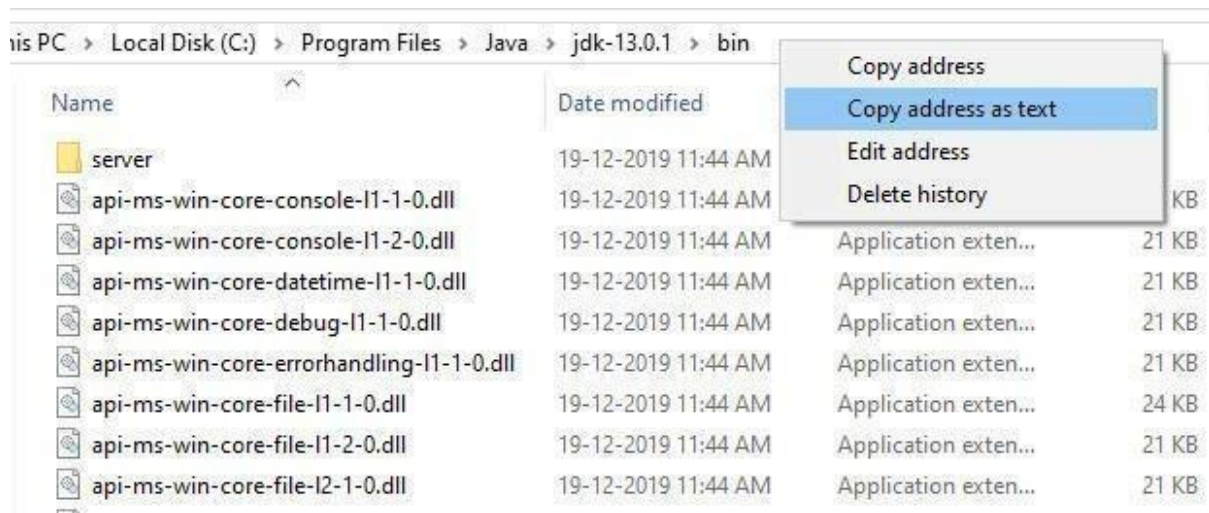


Figure:6.2.9: installed in JDK folder

Step 7) Paste Path of bin folder in Variable value and click on OK Button.

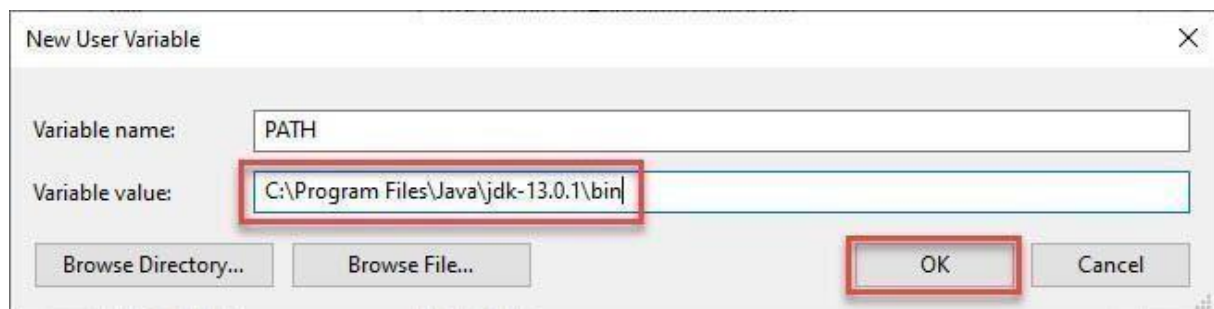


Figure:6.2.10: OK Button

Note: In case you already have a PATH variable created in your PC, edit the PATH variable to

PATH = <JDK installation directory>\bin; %PATH%; Her

Step 8) You can follow a similar process to set CLASSPATH.

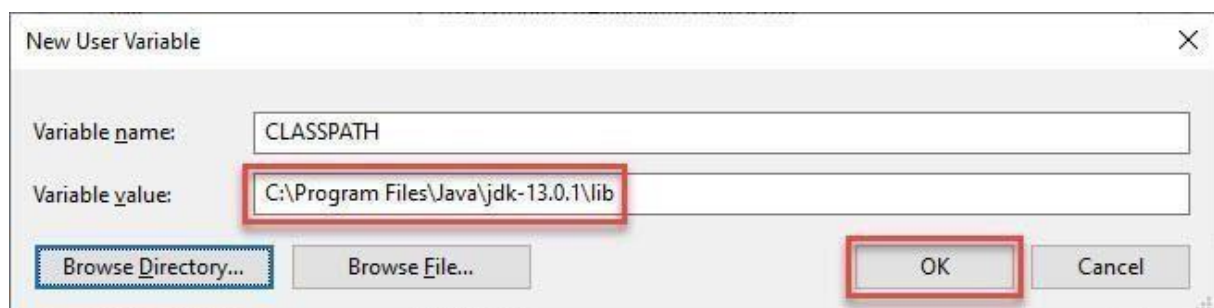


Figure:6.2.11: Class path

Note: In case you java installation does not work after installation, change classpath

CLASSPATH = <JDK installation directory>\lib\tools.jar;

Step 9) Click on OK button

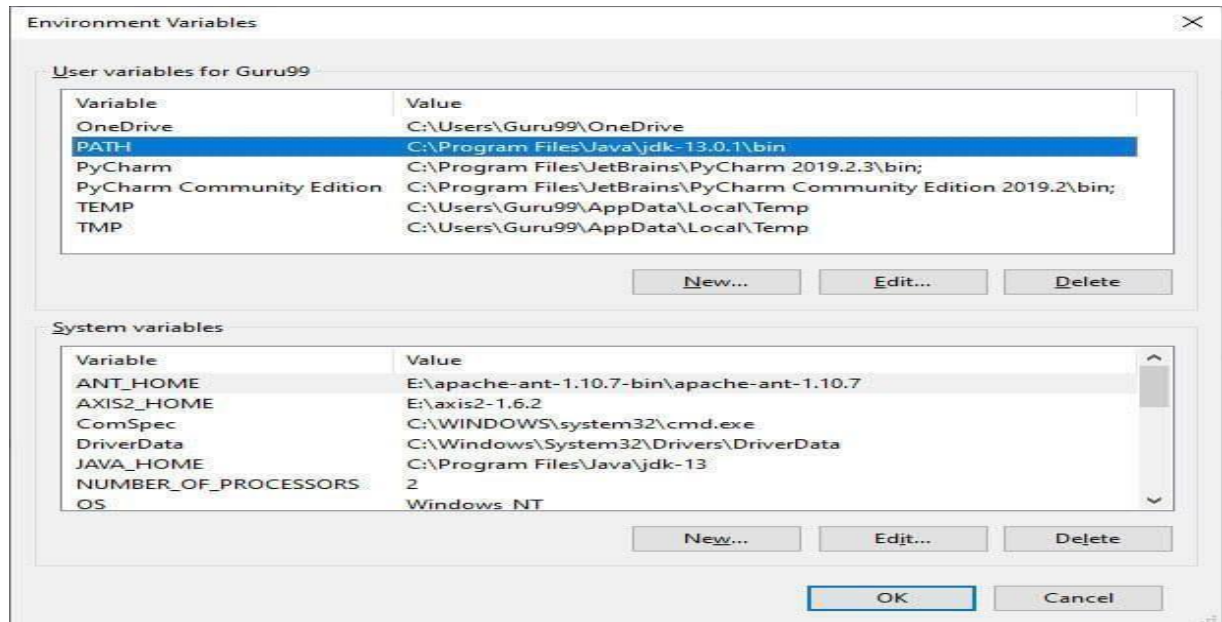


Figure:6.2.12: OK button

Step 10) Go to command prompt and type java commands. If you see a screen like below, Java is installed.

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.535]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Guru99>javac
Usage: javac <options> <source files>
where possible options include:
  @<filename>          Read options and filenames from file
  -Akey[=value]         Options to pass to annotation processors
  --add-modules <module>(<module>)*
                        Root modules to resolve in addition to the initial modules, or all modules
                        on the module path if <module> is ALL-MODULE-PATH.
  --boot-class-path <path>, -bootclasspath <path>
                        Override location of bootstrap class files
  --class-path <path>, -classpath <path>, -cp <path>
                        Specify where to find user class files and annotation processors
  -d <directory>        Specify where to place generated class files
  -deprecation          Output source locations where deprecated APIs are used
  --enable-preview      Enable preview language features. To be used in conjunction with either -source or --release.
  -encoding <encoding>  Specify character encoding used by source files
  -endorseddirs <dirs>  Override location of endorsed standards path
  -extdirs <dirs>       Override location of installed extensions
```

Figure:6.2.13: Execute the Code

3.Android Studio IDE and SDK Installation:

Installing Android software is probably the most challenging part of this project. It takes times - from 30 minutes to n hours to forever - depending on your luck, your programming knowledge, and your PC. You probably need a fairly decent PC(with 8GB RAM) and 10GB of free disk space to run the Android emulator!!! Running

On "actual" Android phone/tablet requires much lesser resources.

Step 0: Pre-Installation Check List

- Before installing Android SDK, you need to install Java Development Kit (JDK). Read "[How to install JDK](#)". Ensure that your JDK is at or above 1.8 You can check your JDK version with command "javac -version".
- Uninstall older version(s) of "Android Studio" and "Android SDK", if any.
- The installation and many operations take a LONG time to complete. Do NOT stare at your screen or at the ceiling. Browse through the "Android Developers" @ <https://developer.android.com>. For developers, check out the "Developer Guides".
- We need to install two huge packages:
 - Android Studio (IDE) (about 1 GB), which is an Integrated Development Environment (IDE) based on IntelliJ (a popular Java IDE); and
 - Android SDK (Software Development Kit) (about 5 GB) for developing and testing Android apps.

Step 1: Install "Android Studio IDE"

Reference: "Install Android Studio" @ <https://developer.android.com/studio/install>.
(For Windows)

- Check that environment variable JAVA_HOME is set to the JDK installation directory via command "set JAVA_HOME". Otherwise, Follow the steps [HERE](#).
- Check the system requirements for Android Studio/SDK @ <https://developer.android.com/studio#Requirements> e.g., For Win
- Go to "Android Studio" @ <https://developer.android.com/studio> ⇒ Click "Download Android Studio 3.6.x for Windows 64-bit (749MB)", e.g., "android-studio-ide-192.xxxxxxx-windows.exe".
- Run the downloaded installer ⇒ You may watch a short video @ <https://developer.android.com/studio/install>.

- In "Choose Components", select "Android Studio" and "Android Virtual Device".
- In "Configuration Settings Install Location", accept the default "C:\Program Files\Android\Android Studio".
- In "Choose Start Menu Folder", accept the default ⇒ Install.
- Launch Android Studio. Continue to the next Step.
- by default, the "Android Studio IDE" will be installed in "C:\Program Files\Android\Android Studio", and the "Android SDK" in "c:\Users\username\AppData\Local\Android\Sdk".

Notes: You can also use the ZIP version: Download the Windows 64-bit ZIP version (about 1.5GB) ⇒ UNZIP into a folder of your choice ⇒ Run "bin\studio64.exe" to launch the Android Studio ⇒ It will enter the "setup" for the first launch ⇒ "Do not Import Settings" ⇒ In "Welcome", click "Next" ⇒ In "Install Type", choose "Custom" (so that you can see what is going on) ⇒ In "Select Default JDK Location", use default ⇒ In "Select UI Theme", choose one that you like ⇒ In "SDK Components Setup", select "Android Virtual Device (1.05GB)" ⇒ Take note of the "Android SDK Location" with default of "C:\Users\username\AppData\Local\Android\Sdk" ⇒ In "Emulator Settings", use default ⇒ In "Verify Settings", check the settings and choose "Finish" ⇒ In "Download Components", click "Details" and check that nothing fails ⇒ Wait ⇒ Wait ⇒ Wait.

Step 2: Installing Android SDK

This step takes a long time as you need to download about 3GB of zip data, and expand to 5 GB of disk data, even for the minimum configuration.

Note: You can actually copy the SDK from another computer with the same OS.

(For Windows and macOS)

- Launch Android Studio ⇒ It will run the "setup" wizard for the first launch.
- Choose "do not import previous settings".
- In "Welcome", choose "next".
- In "Install Type", choose "Standard" (default).
- In "Select UI Theme", choose one that you like (or default).

- In "SDK Components Setup", make sure that "Android Virtual Device" is selected and take note of the SDK directory (by default @ c:\Users\username\AppData\Local\Android\Sdk for Windows, "~/Library/Android/sdk" for macOS) ⇒ Finish ⇒ Wait ⇒ Wait ⇒ Wait ⇒ Wait ⇒ Wait ⇒ Wait.
- (For Windows) Use "File Explorer" to check the SDK installed directory. Take note that the "App Data" is a hidden directory. You need to choose "View" menu ⇒ Uncheck "Hidden Items" to see this directory. (For macOS) Use "Finder" to check the SDK installed directory.
- Also use "Android Studio" to check the SDK packages installed by selecting "Configure" (at the bottom of Android Studio) ⇒ "SDK Manager" ⇒ "Android SDK" (sidebar):
 - Under "SDK Platforms" tab:
 - Android 10.0 (Q) (API Level 29)
- 2. Under "SDK Tools" tab:
 - Android SDK Build Tools 30-rc1
 - Android Emulator (30.0.0)
 - Android SDK Platform-Tools (29.0.6)
 - Intel x86 Emulator Accelerator (HAXM installer) (7.5.6)

STEPS FOR EXECUTING THE PROJECTS

Step 1:

Open Android Studio

Step2:

Choose a virtual device or Physical device from the menu

Step3:

Click on the project Run

Step4:

View the application performance on virtual or Physical device.

6.3.SOFTWARE DEVELOPMENT LIFE CYCLE

The meaning of Agile is swift or versatile. "Agile process model" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance. Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.

Actually, Agile model refers to a group of development processes. These processes share some basic characteristics but do have certain subtle differences among themselves. A few Agile SDLC models are given below: Crystal A tern Feature-driven development Scrum Extreme programming (XP) Lean development Unified process In the Agile model, the requirements are decomposed into many small parts that can be incrementally developed.

The Agile model adopts Iterative development. Each incremental part is developed over an iteration. Each iteration is intended to be small and easily manageable and that can be completed within a couple of weeks only. At a time one iteration is planned, developed and deployed to the customers. Long-term plans are not made.

Agile model is the combination of iterative and incremental process models. Steps involve in agile SDLC models are:

- Requirement gathering
- Requirement Analysis
- Design Coding
- Unit testing
- Acceptance testing

The time to complete an iteration is known as a Time Box. Time-box refers to the maximum amount of time needed to deliver an iteration to customers. So, the end date for an iteration does not change.

The central principle of the Agile model is the delivery of an increment to the customer after each Time-box.

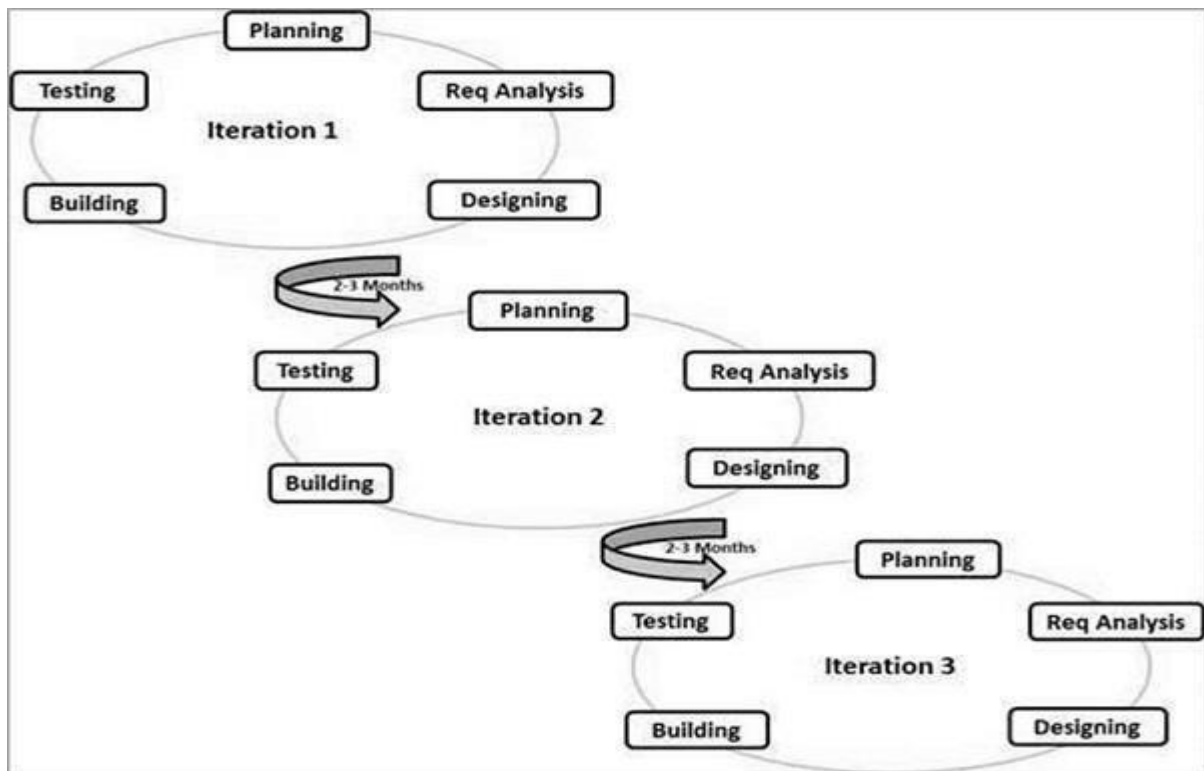


Figure:6.3.1:Software Development Cycle

Principles of Agile model:

- To establish close contact with the customer during development and to gain a clear understanding of various requirements, each Agile project usually includes a customer representative on the team. At the end of each iteration stakeholders and the customer representative review, the progress made and re-evaluate the requirements.
- Agile model relies on working software deployment rather than comprehensive documentation.
- Frequent delivery of incremental versions of the software to the customer representative in intervals of few weeks.
- Requirement change requests from the customer are encouraged and efficiently incorporated.
- It emphasizes on having efficient team members and enhancing communications among them is given more importance. It is realized that

- enhanced communication among the development team members can be achieved through face-to-face communication rather than through the exchange of formal documents.
- It is recommended that the development team size should be kept small (5 to 9 people) to help the team members meaningfully engage in face-to-face communication and have collaborative work environment.
- Agile development process usually deploys Pair Programming. In Pair programming, two programmers work together at one work-station. One does code while the other reviews the code as it is typed in. The two programmers switch their roles every hour or so.

Advantages:

- Working through Pair programming produce well written compact programs which has fewer errors as compared to programmers working alone.
- It reduces total development time of the whole project. Customer representatives get the idea of updated software products after each iteration. So, it is easy for him to change any requirement if needed.

Disadvantages:

- Due to lack of formal documents, it creates confusion and important decisions taken during different phases can be misinterpreted at any time by different team members.
- Due to the absence of proper documentation, when the project completes and the developers are assigned to another project, maintenance of the developed project can become a problem.

SOFTWARE ENVIRONMENT

Software Environment

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. Google Inc. purchased the initial developer of the software, Android Inc., in 2005.

Android's mobile operating system is based on the Linux kernel. Google and other members of the Open Handset Alliance collaborated on Android's development and release.

The Android Open-Source Project (AOSP) is tasked with the maintenance and further development of Android. The Android operating system is the world's best-selling Smartphone platform.

The Android SDK provides the tools and APIs necessary to begin developing applications Android platform using the Java programming language. Android has a large community of developers writing applications ("apps") that extend the functionality of the devices. There are currently over 250,000 apps available for Android.

Features: -

- **Application framework** enabling reuse and replacement of components
- **Dalvik virtual machine** optimized for mobile devices
- **Integrated browser** based on the open source Web Kit engine
- **Optimized graphics** powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional)
- **SQLite** for structured data storage
- **Media support** for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- **GSM Telephony** (hardware dependent)
- **Bluetooth, EDGE, 3G, and WIFI** (hardware dependent)
- **Camera, GPS, compass, and accelerometer** (hardware dependent)
- **Rich development environment** including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE

Android Architecture



Figure:6.3.2: Android Architecture

Libraries

Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework. Some of the core libraries are listed below:

- **System C library** - a BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices
- **Media Libraries** - based on Packet Video's Open CORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG
- **Surface Manager** - manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications
- **LibWebCore** - a modern web browser engine which powers both the Android browser and an embeddable web view
- **SGL** - the underlying 2D graphics engine

- **3D libraries** - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the included, highly optimized 3D software rasterizer
- **Free Type** - bitmap and vector font rendering
- **SQLite** - a powerful and lightweight relational database engine available to all applications

Android Runtime

Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language.

Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently. The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dx" tool.

The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

Linux Kernel

Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack. The Linux kernel is an operating system kernel used by the Linux family of Unix-like operating systems. It is one of the most prominent examples of free and open source software. The Linux kernel is released under the GNU General Public License version 2 (GPLv2), (plus some firmware images with various licenses), and is developed by contributors worldwide. Day-to-day development takes place on the Linux kernel mailing list. The Linux kernel was initially conceived and created by Finnish computer science student Linus Torvalds in 1991. Linux rapidly accumulated developers and users who adapted code from other free software projects for use with the new operating system. The Linux kernel has received

contributions from thousands of programmers. Many Linux distributions have been released based upon the Linux kernel. The Linux kernel has extensive support for and runs on many virtual machine architectures both as the host operating system and as a guest operating system. The virtual machines usually emulate Intel x86 family of processors, though in a few cases PowerPC or ARM processors are also emulated. At Google, the team led by Rubin developed a mobile device platform powered by the Linux kernel. Google marketed the platform to handset makers and carriers on the premise of providing a flexible, upgradable system. Google had lined up a series of hardware component and software partners and signaled to carriers that it was open to various degrees of cooperation on their part. Speculation about Google's intention to enter the mobile communications market continued to build through December 2006. Reports from the BBC and The Wall Street Journal noted that Google wanted its search and applications on mobile phones and it was working hard to deliver that. Print and online media outlets soon reported rumors that Google was developing a Google-branded handset. Some speculated that as Google was defining technical specifications, it was showing prototypes to cell phone manufacturers and network operators.

Hardware running Android

The main supported platform for Android is the ARM architecture.

The Android OS can be used as an operating system for cellphones, netbooks and tablets, including the Dell Streak, Samsung Galaxy Tab, TV and other devices.[68][69] The first commercially available phone to run the Android operating system was the HTC Dream, released on 22 October 2008.[70] In early 2010 Google collaborated with HTC to launch its flagship [71] Android device, the Nexus One. This was followed later in 2010 with the Samsung-made Nexus S.

The early feedback on developing applications for the Android platform was mixed. Issues cited include bugs, lack of documentation, inadequate QA infrastructure, and no public issue-tracking system. (Google announced an issue tracker on 18 January 2008.) In December 2007, Merge Lab mobile startup founder Adam Macbeth stated, "Functionality is not there, is poorly documented or just doesn't work... It's clearly not ready for prime time." Despite this, Android-targeted.

the week after the platform was announced. The first publicly available application was the Snake game. The Android Dev Phone is a SIM-unlocked and hardware-unlocked device that is designed for advanced developers. While developers can use regular consumer devices purchased at retail to test and use their applications, some developers may choose not to use a retail device, preferring an unlocked or no-contract device.

The Android software development kit (SDK) includes a comprehensive set of development tools.[80] These include a debugger, libraries, a handset emulator (based on QEMU), documentation, sample code, and tutorials. The SDK is downloadable on the android developer website. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.4.9 or later, Windows XP or later. The officially supported integrated development environment (IDE) is Eclipse (currently 3.5 or 3.6) using the Android Development Tools (ADT) Plugin, though developers may use any text editor to edit Java and XML files then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely).[81]

Android applications are packaged in .apk format and stored under /data/app folder on the Android OS (the folder is accessible to root user only for security reasons). APK package contains .dex files (compiled byte code files called Dalvik executables), resource files, etc.

Android Operation System

Android is an operating system based on Linux with a Java programming interface. It provides tools, e.g., a compiler, debugger and a device emulator as well as its own Java Virtual machine (Dalvik Virtual Machine - DVM). Android is created by the Open Handset Alliance which is led by Google.

Android uses a special virtual machine, e.g., the Dalvik Virtual Machine. Dalvik uses special bytecode. Therefore, you cannot run standard Java bytecode on Android. Android provides a tool "dx" which allows to convert Java Class files into "dex" (Dalvik Executable) files. Android applications are packed into an .apk (Android Package) file by the program "aapt" (Android Asset Packaging Tool) To simplify development Google provides the Android Development Tools (ADT) for Eclipse . The ADT performs automatically the conversion from class to dex files and creates the apk during deployment.

Android supports 2-D and 3-D graphics using the OpenGL libraries and supports data storage in a SQLite database.

Every Android applications run in its own process and under its own user id which is generated automatically by the Android system during deployment. Therefore, the application is isolated from other running applications and a misbehaving application cannot easily harm other Android applications.

Important Android components

An Android application consists out of the following parts:

- Activity - Represents the presentation layer of an Android application, e.g., a screen which the user sees. An Android application can have several activities and it can be switched between them during runtime of the application.
- Views - The User interface of an Activities is built with widgets classes which inherent from "android. view. View". The layout of the views is managed by "android. view. View Groups".
- Services - perform background tasks without providing an UI. They can notify the user via the notification framework in Android.
- Content Provider - provides data to applications, via a content provider your application can share data with other applications. Android contains a SQLite DB which can serve as data provider
- Intents are asynchronous messages which allow the application to request functionality from other services or activities. An application can call directly a service or activity (explicit intent) or asked the Android system for registered services and applications for an intent (implicit intents). For

application could ask via an intent for a contact application. Application registers itself to an intent via an Intent Filter. Intents are a powerful concept as they allow to create loosely coupled applications.

- Broadcast Receiver - receives system messages and implicit intents, can be used to react to changed conditions in the system. An application can register as a broadcast receiver for certain events and can be started if such an event occurs.
- A Java Virtual Machine (JVM) enables a set of computer software programs and data structures to use a virtual machine model for the execution of other computer programs and scripts. The model used by a JVM accepts a form of computer intermediate language commonly referred to as Java bytecode. This language conceptually represents the instruction set of a stack-oriented, capability architecture. Sun Microsystems states there are over 4.5 billion JVM-enabled devices
- A JVM can also execute bytecode compiled from programming languages other than Java. For example, Ada source code can be compiled to execute on a JVM. JVMs can also be released by other companies besides Oracle (the developer of Java) — JVMs using the "Java" trademark may be developed by other companies as long as they adhere to the JVM specification published by Oracle and to related contractual obligations.
- Java was conceived with the concept of WORA: "write once, run anywhere". This is done using the Java Virtual Machine. The JVM is the environment in which java programs execute. It is software that is implemented on non-virtual hardware and on standard operating systems.
- JVM is a crucial component of the Java platform, and because JVMs are available for many hardware and software platforms, Java can be both middleware and a platform in its own right, [clarification needed] hence the trademark write once, run anywhere. The use of the same bytecode for all platforms allows Java to be described as "compile once, run anywhere", as opposed to "write once, compile anywhere", which describes cross-platform compiled languages. A JVM also enables such features as automated exception handling, which provides "root-cause" debugging information for every software error (exception), independent of the source code.

- A JVM is distributed along with a set of standard class libraries that implement the Java application programming interface (API). Appropriate APIs bundled together form the Java Runtime Environment (JRE).
- Java's execution environment is termed the Java Runtime Environment, or JRE.
- Programs intended to run on a JVM must be compiled into a standardized portable binary format, which typically comes in the form of .class files. A program may consist of many classes in different files. For easier distribution of large programs, multiple class files may be packaged together in a .jar file (short for Java archive).
- The Java application launcher, java, offers a standard way of executing Java code. Compare javaw.^[2]
- The JVM runtime executes .class or .jar files, emulating the JVM instruction set by interpreting it, or using a just-in-time compiler (JIT) such as Oracle's Hotspot. JIT compiling, not interpreting, is used in most JVMs today to achieve greater speed. There are also ahead-of-time compilers that enable developers to precompile class files into native code for particular platforms.
- Like most virtual machines, the Java Virtual Machine has a stack-based architecture akin to a microcontroller/microprocessor. However, the JVM also has low-level support for Java-like classes and methods, which amounts to a highly idiosyncratic [clarification needed] memory model and capability-based architecture.

Download the Android SDK

Welcome Developers! If you are new to the Android SDK, please read the steps below, for an overview of how to set up the SDK.

If you're already using the Android SDK, you should update to the latest tools or platform using the *Android SDK and AVD Manager*, rather than downloading a new SDK starter package. See Adding SDK Components.

Here an overview of the steps you must follow to set up the Android SDK:

- Prepare your development computer and ensure it meets the system requirements.
- Install the SDK starter package from the table above. (If you're on Windows, download the installer for help with the initial setup.)
- Install the ADT Plugin for Eclipse (if you'll be developing in Eclipse).
- Add Android platforms and other components to your SDK.
- Explore the contents of the Android SDK (optional).

To get started, download the appropriate package from the table above, then read the guide to Installing the SDK. Installing the SDK

Step 1. Preparing Your Development Computer

Before getting started with the Android SDK, take a moment to confirm that your development computer meets the System Requirements. In particular, you might need to install the JDK, if you don't have it already.

If you will be developing in Eclipse with the Android Development Tools (ADT) Plugin—the recommended path if you are new to Android—make sure that you have a suitable version of Eclipse installed on your computer as described in the System Requirements document. If you need to install Eclipse, you can download it from this location:

The "Eclipse Classic" version is recommended. Otherwise, a Java or RCP version of Eclipse is recommended.

Use the Eclipse update manager to install all available plugins for the Android Development Tools (ADT) from the URL <https://dl-ssl.google.com/android/eclipse/>.
Configuration

In Eclipse open the Preferences dialog via Windows -> Preferences. Select Android and maintain the installation path of the Android SDK.

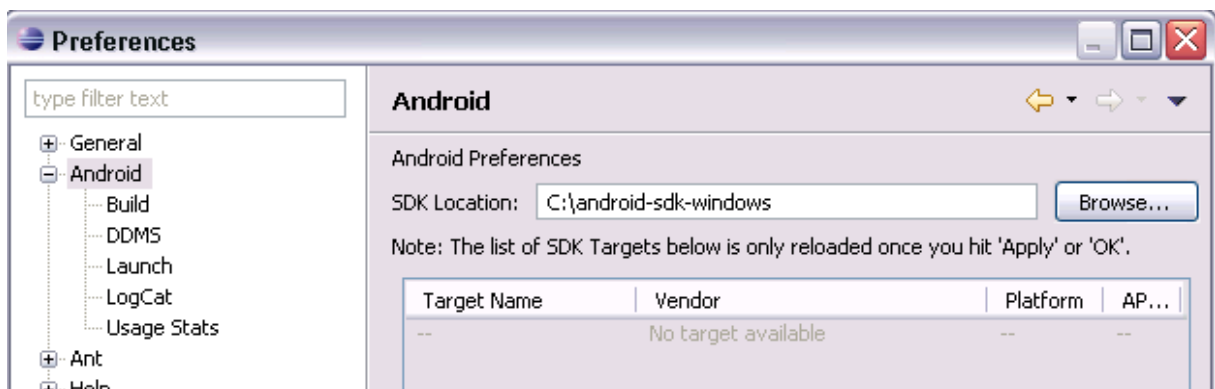


Figure:6.3.3: Preference

Select Window -> Android SDK and AVD Manager from the menu.

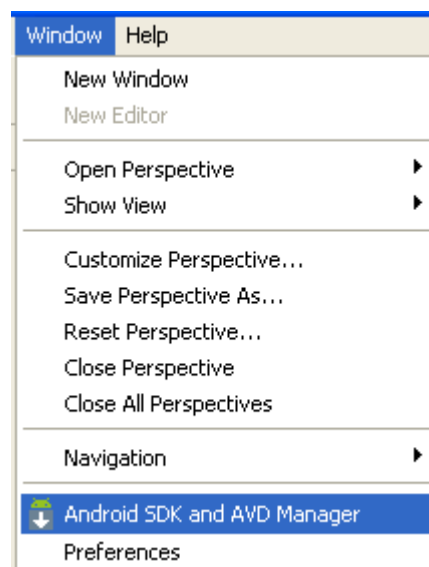


Figure:6.3.4: Preference

Select available packages and select the latest version of the SDK. Step 2.

Downloading the SDK Starter Package

The SDK starter package is not a full development environment—it includes only the core SDK Tools, which you can use to download the rest of the SDK components (such as the latest Android platform).

If you haven't already, get the latest version of the SDK starter package from the SDK download page.

If you downloaded a .zip or .tgz package (instead of the SDK installer), unpack it to a safe location on your machine. By default, the SDK files are unpacked into a directory named android-sdk-<machine-platform>.

If you downloaded the Windows installer (.exe file), run it now and it will check whether the proper Java SE Development Kit (JDK) is installed (installing it, if necessary), then install the SDK Tools into a default location (which you can modify).

Make a note of the name and location of the SDK directory on your system—you will need to refer to the SDK directory later, when setting up the ADT plugin and when using the SDK tools from the command line.

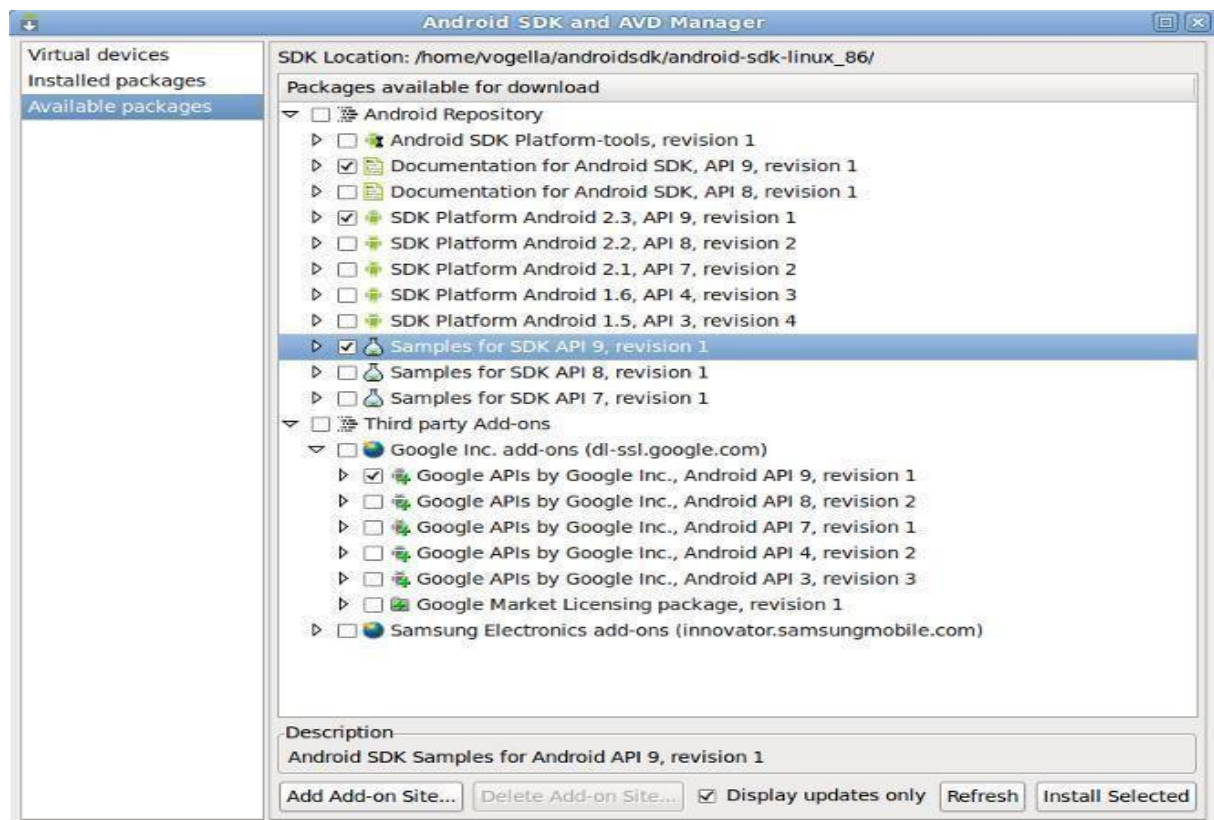


Figure:6.3.4: Packages

Step 3. Installing the ADT Plugin for Eclipse

Android offers a custom plugin for the Eclipse IDE, called Android Development Tools (ADT), that is designed to give you a powerful, integrated environment in which to build Android applications. It extends the capabilities of Eclipse to let you quickly set up new Android projects, create an application UI, debug your applications using the Android SDK tools, and even export signed (or unsigned) APKs in order to distribute your application. In general, developing in Eclipse with ADT is a highly recommended approach and is the fastest way to get started with Android.

If you'd like to use ADT for developing Android applications, install it now. Read Installing the ADT Plugin for step-by-step installation instructions, then return here to continue the last step in setting up your Android SDK.

If you prefer to work in a different IDE, you do not need to install Eclipse or ADT. Instead, you can directly use the SDK tools to build and debug your application. The Introduction to Android application development outlines the major steps that you need to complete when developing in Eclipse or other IDEs.

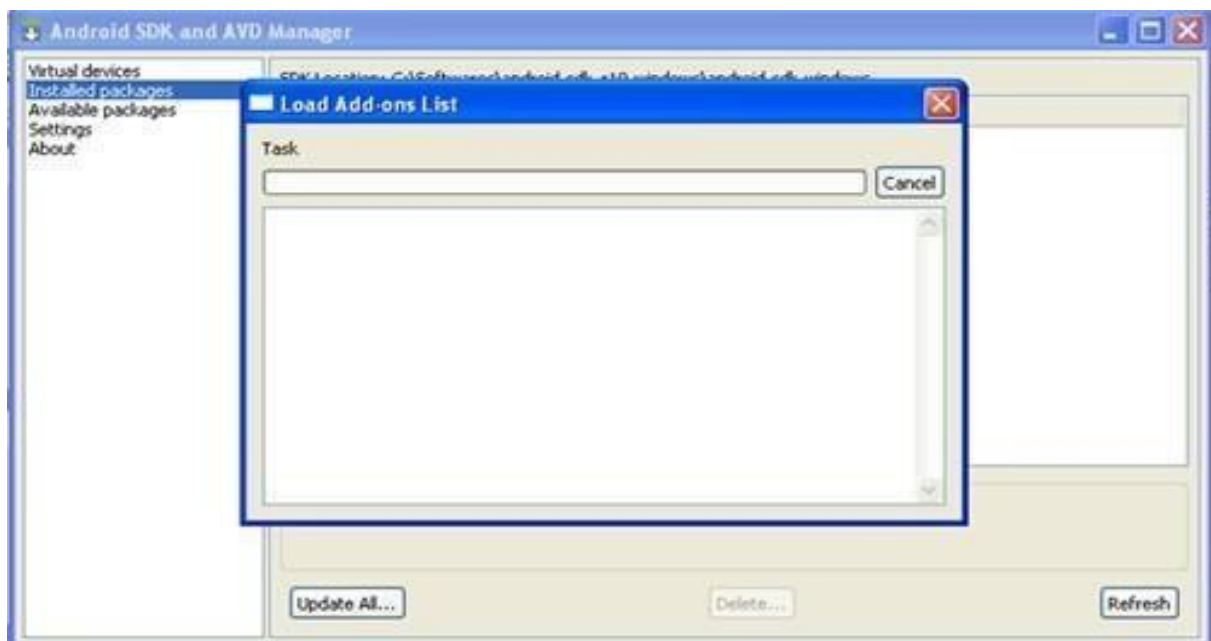


Figure:6.3.5:ADT Plugin

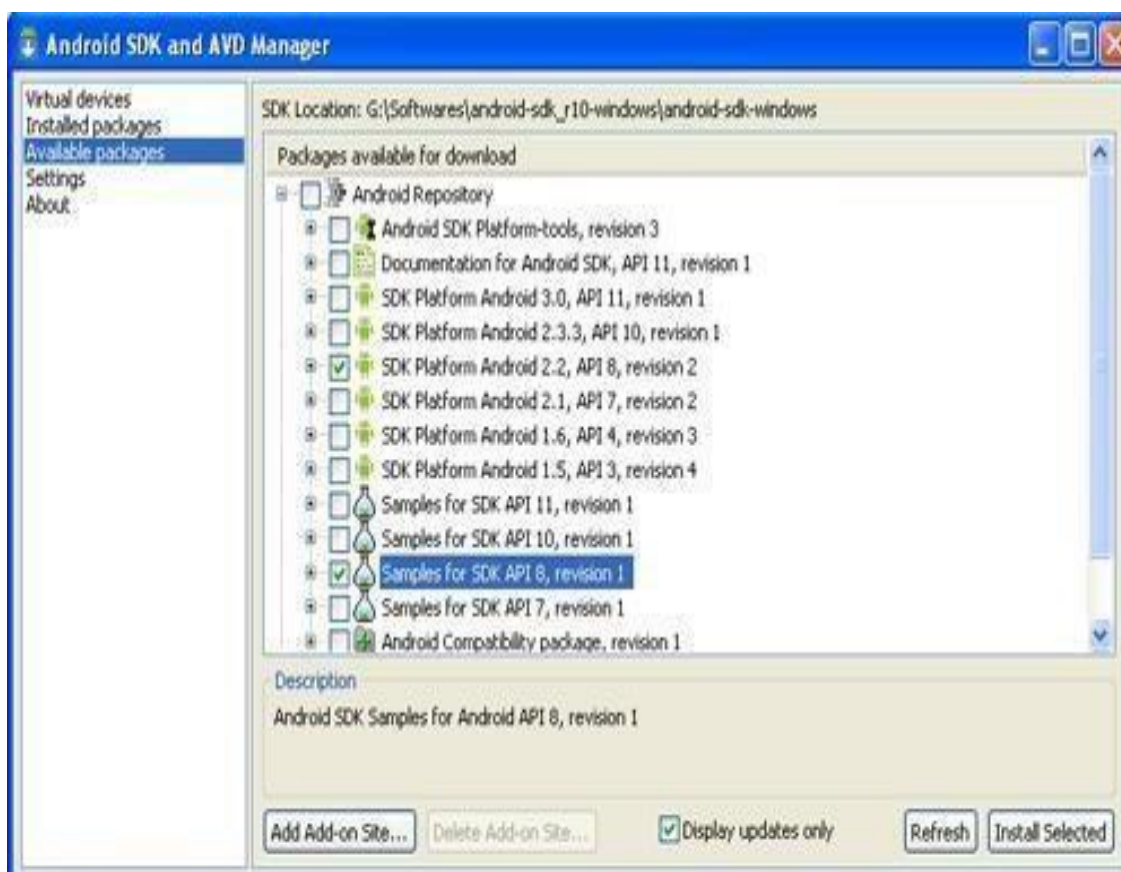


Figure 6.3.5: Select the Packages

Step 4. Adding Platforms and Other Components

The last step in setting up your SDK is using the Android SDK and AVD Manager (a tool included in the SDK starter package) to download essential SDK components into your development environment.

The SDK uses a modular structure that separates the major parts of the SDK—Android platform versions, add-ons, tools, samples, and documentation—into a set of separately installable components. The SDK starter package, which you've already downloaded, includes only a single component: the latest version of the SDK Tools. To develop an Android application, you also need to download at least one Android platform and the associated platform tools. You can add other components and platforms as well, which is highly recommended.

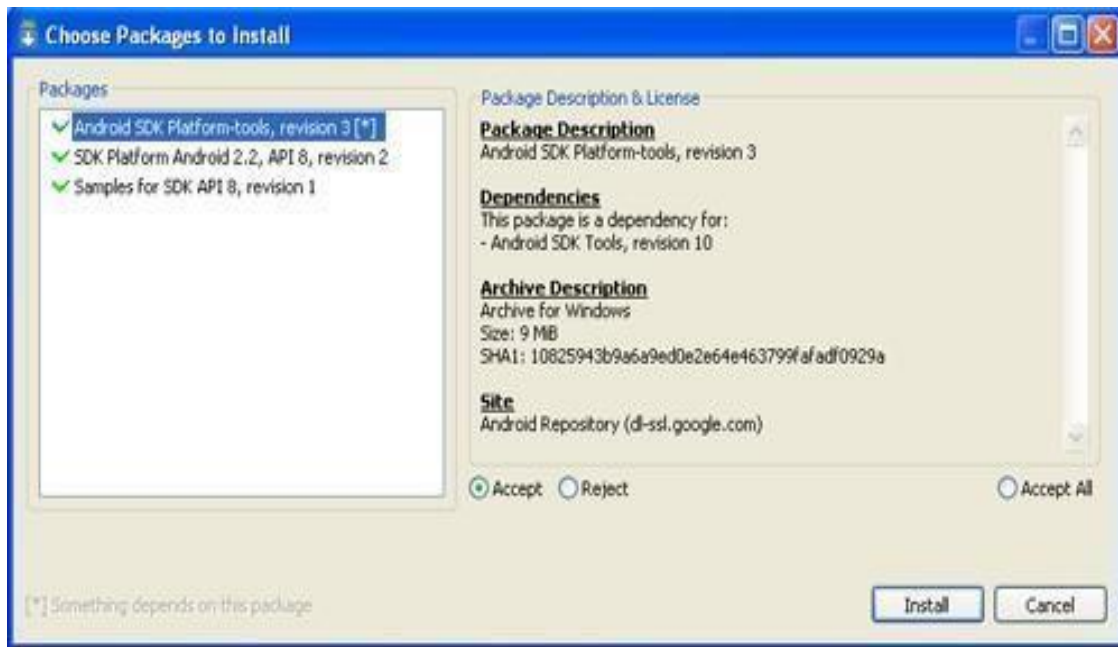


Figure:6.3.6:Choose the packages to install

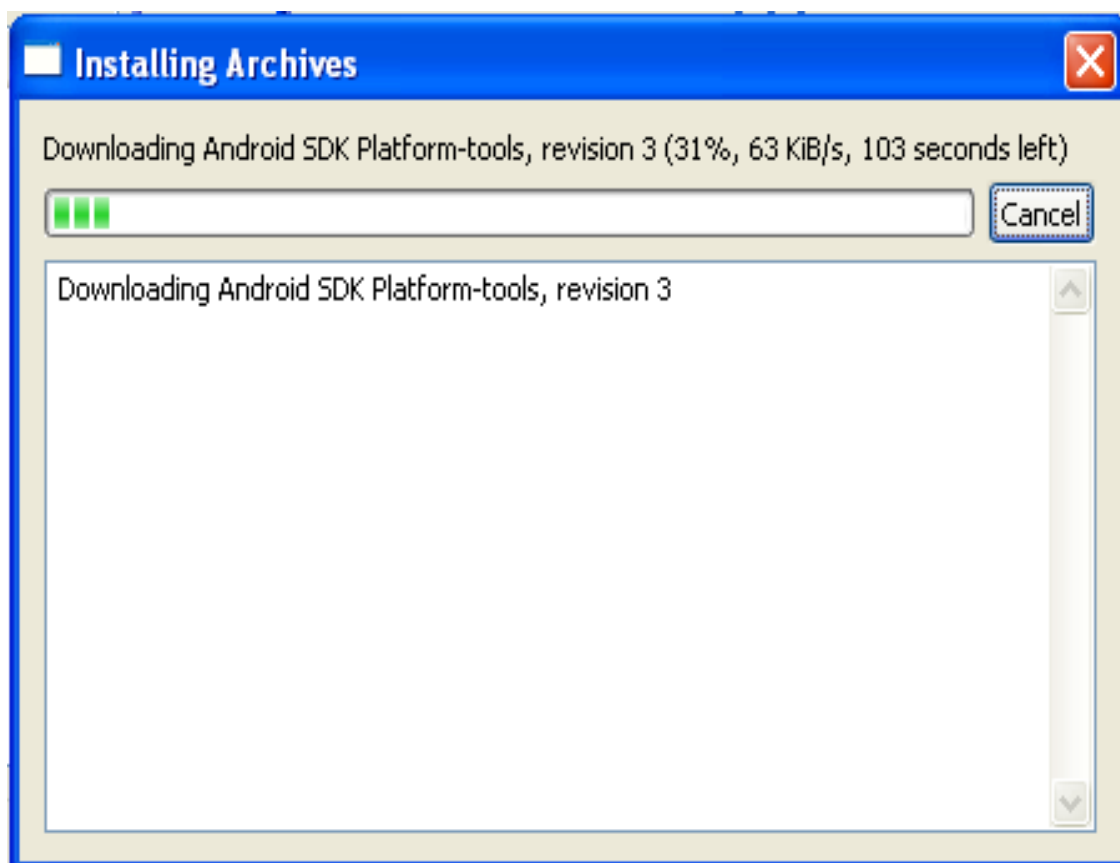


Figure:6.3.7: Install the packages

If you used the Windows installer, when you complete the installation wizard, it will launch the Android SDK and AVD Manager with a default set of platforms and other

components selected for you to install. Simply click **Install** to accept the recommended set of components and install them. You can then skip to Step 5, but we recommend you first read the section about the Available Components to better understand the components available from the Android SDK and AVD Manager.

You can launch the Android SDK and AVD Manager in one of the following ways:

- From within Eclipse, select **Window > Android SDK and AVD Manager**.
- On Windows, double-click the SDK Manager.exe file at the root of the Android SDK directory.
- On Mac or Linux, open a terminal and navigate to the tools/ directory in the Android SDK, then execute:

To download components, use the graphical UI of the Android SDK and AVD Manager to browse the SDK repository and select new or updated components (see figure 1). The Android SDK and AVD Manager installs the selected components in your SDK environment. For information about which components you should download, see Recommended Components.

The *Android Repository* offers these types of components:

- **SDK Tools** — Contains tools for debugging and testing your application and other utility tools. These tools are installed with the Android SDK starter package and receive periodic updates. You can access these tools in the
- <sdk>/tools/ directory of your SDK. To learn more about them, see SDK Tools in the developer guide.
- **SDK Platform-tools** — Contains platform-dependent tools for developing and debugging your application. These tools support the latest features of the Android and platform are typically updated only when a new platform becomes available. You can access these tools in the <sdk>/platform-tools/ directory. To learn more about them, see Platform Tools in the developer.

- **Android platforms** — An SDK platform is available for every production Android platform deployable to Android-powered devices. Each SDK platform component includes a fully compliant Android library, system image, sample code, and emulator skins. To learn more about a specific platform, see the list of platforms that appears under the section "Downloadable SDK Components" on the left part of this page.
- **USB Driver for Windows** (Windows only) — Contains driver files that you can install on your windows computer, so that you can run and debug your applications on an actual device. You *do not* need the USB driver unless you plan to debug your application on an actual Android-powered device. If you develop on Mac OS X or Linux, you do not need a special driver to debug your application on an Android-powered device. See Using Hardware Devices for more information about developing on a real device.
- **Samples** — Contains the sample code and apps available for each Android development platform. If you are just getting started with Android development, make sure to download the samples to your SDK.
- **Documentation** — Contains a local copy of the latest multi-version documentation for the Android framework API.

The *Third-party Add-ons* provide components that allow you to create a development environment using a specific Android external library (such as the Google Maps library) or a customized (but fully compliant) Android system image. You can add additional Add-on repositories by clicking **Add Add-on Site**.

ECLIPSE:

Eclipse is an open-source community whose projects are focused on building an extensible development platform, runtimes and application frameworks for building, deploying and managing software across the entire software lifecycle. Many people know us, and hopefully love us, as a Java IDE but Eclipse is much more than a Java IDE.

The Eclipse open-source community has over 60 open-source projects. These projects can be conceptually organized into seven different "pillars" or categories:

1. Enterprise Development
2. Embedded and Device Development
3. Rich Client Platform
4. Rich Internet Applications
5. Application Frameworks
6. Application Lifecycle Management (ALM)
7. Service Oriented Architecture (SOA)

The Eclipse community is also supported by a large and vibrant ecosystem of major IT solution providers, innovative start-ups, universities and research institutions and individuals that extend, support and complement the Eclipse Platform.

The exciting thing about Eclipse is many people are using Eclipse in ways that we have never imagined. The common thread is that they are building innovative, industrial strength software and want to use great tools, frameworks and runtimes to make their job easier.

Eclipse is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system. It is written mostly in Java and can be used to develop applications in Java and, by means of various plug-ins, other programming languages including Ada, C, C++, COBOL, Perl, PHP, Python, Ruby (including Ruby on Rails framework), Scala, Clojure, and Scheme. The IDE is often called Eclipse ADT for Ada, Eclipse CDT for C/C++, Eclipse JDT for Java, and Eclipse PDT for PHP.

ARCHITECTURE:

Eclipse employs plug-ins in order to provide all of its functionality on top of (and including) the runtime system, in contrast to some other applications where functionality is typically hard coded. The runtime system of Eclipse is based on Equinox, an OSGi standard compliant implementation.

This plug-in mechanism is a lightweight software componentry framework. In addition to allowing Eclipse to be extended using other programming languages such as C and Python, the plug-in framework allows Eclipse to work with typesetting languages like LaTeX,[2] networking applications such as telnet, and database management systems. The plug-in architecture supports writing any desired extension to the environment, such as for configuration management. Java and CVS support is provided in the Eclipse SDK, with Subversion support provided by third-party plugins.

With the exception of a small run-time kernel, everything in Eclipse is a plug-in. This means that every plug-in developed integrates with Eclipse in exactly the same way as other plug-ins; in this respect, all features are "created equal". Eclipse provides plug-ins for a wide variety of features, some of which are through third parties using both free and commercial models. Examples of plug-ins include a UML plug-in for Sequence and other UML diagrams, a plug-in for DB Explorer, and many others.

The Eclipse SDK includes the Eclipse Java Development Tools (JDT), offering an IDE with a built-in incremental Java compiler and a full model of the Java source files. This allows for advanced refactoring techniques and code analysis. The IDE also makes use of a workspace, in this case a set of metadata over a flat file space allowing external file modifications as long as the corresponding workspace "resource" is refreshed afterwards

The Eclipse SDK includes the Eclipse Java Development Tools (JDT), offering an IDE with a built-in incremental Java compiler and a full model of the Java source files. This allows for advanced refactoring techniques and code analysis. The IDE also makes use of a workspace, in this case a set of metadata over a flat file space allowing external file modifications as long as the corresponding workspace "resource" is refreshed afterwards.

Eclipse implements widgets through a widget toolkit for Java called SWT, unlike most Java applications, which use the Java standard Abstract Window Toolkit (AWT) or Swing. Eclipse's user interface also uses an intermediate GUI layer called JFace, which simplifies the construction of applications based on SWT.

Rich Client Platform

- Equinox OSGi – a standard bundling framework.
- Core platform – boot Eclipse, run plug-ins
- Standard Widget Toolkit (SWT) – a portable widget toolkit
- JFace – viewer classes to bring model view controller programming to SWT, file buffers, text handling, text editors
- Eclipse Workbench – views, editors, perspectives, wizards

History

- Eclipse began as an IBM Canada project. It was developed by Object Technology International (OTI) as a Java-based replacement for the Smalltalk based Visual Age family of IDE products,[4] which itself had been developed by OTI.[1] In November 2001, a consortium was formed to further the development of Eclipse as open source. In January 2004, the Eclipse Foundation was created.[5]

Eclipse 3.0 (released on 21 June 2004) selected the OSGi Service Platform specifications as the runtime architecture.[6]

Eclipse was originally released under the Common Public License, but was later relicensed under the Eclipse Public License. The Free Software Foundation has said that both licenses are free software licenses, but are incompatible with the GNU General Public License (GPL).[7] Mike Milinkovich, of the Eclipse Foundation commented that moving to the GPL would be considered when version 3 of the GPL was released.

According to Lee Nackman, Chief Technology Officer of IBM's Rational division at that time and later head of Rational software development and support, the name "Eclipse" was chosen to target Microsoft's Visual Studio product, and not Sun Microsystems. Ironically, Nackman is now himself a Microsoft employee.

Eclipse (SDK)

Eclipse Software Development Kit (SDK) is a Java based open-source integrated development environment (IDE) which combines a number of different Eclipse projects including Platform, Java Development Tools (JDT) and the Plug-in Development Environment (PDE).

Eclipse can be used to create a large array of software applications using languages ranging from PHP, C++ programs, to Java. It is one of the most popular development tools in both the open-source and commercial worlds.

It provides Java editing with validation, incremental compilation, cross-referencing, code assist; an XML Editor; Mylyn; and much more.

Eclipse is released under the Eclipse Foundation, a commercially friendly license that allows organizations to include Eclipse software in their commercial products, while at the same time asking those who create derivative works of EPL code to contribute back to the community.

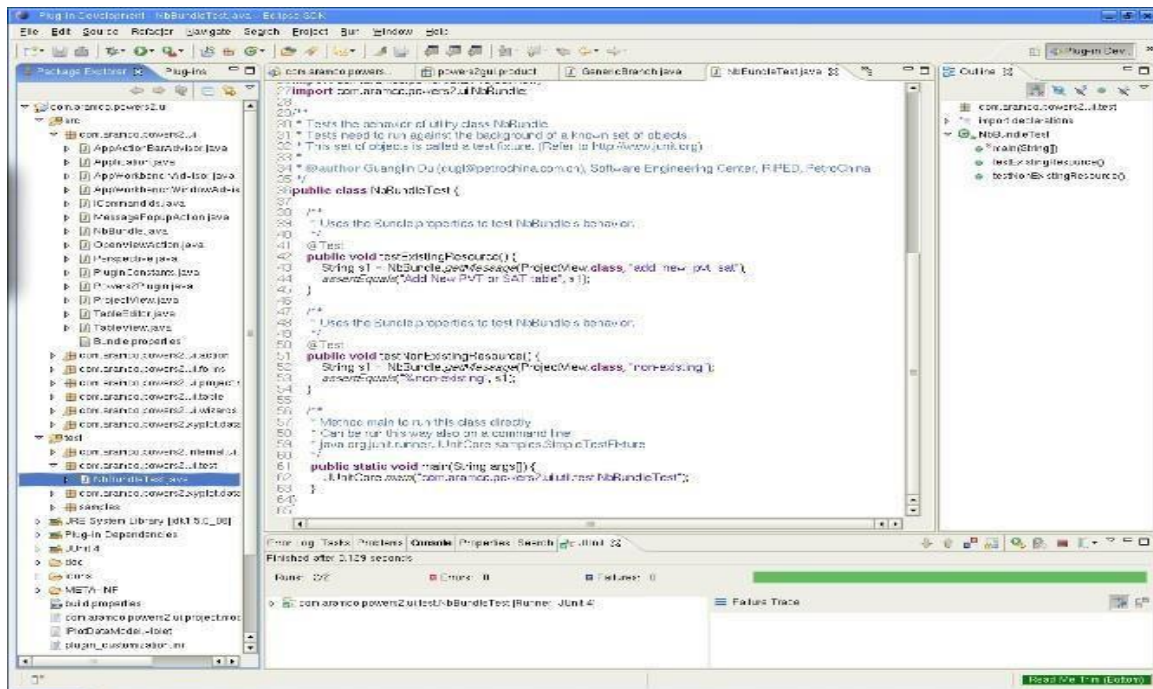


Figure:6.3.8:Execute the Code

Eclipse Platform

The Eclipse Platform provides the core frameworks and services upon which all plug-in extensions are created. It also provides the runtime in which plug-ins are loaded, integrated, and executed. The primary purpose of the Platform is to enable other tool developers to easily build and deliver integrated tools.

Features include:

- Supports the construction of a variety of tools for application development
- Supports an unrestricted set of tool providers, including independent software vendors (ISVs)
- Supports tools to manipulate arbitrary content types (e.g., HTML, Java, C, JSP, EJB, XML, and GIF)
- Facilitates seamless integration of tools within and across different content types and tool providers

Supports both GUI and non-GUI-based application development environments

Java Development Tools (JDT)

The JDT project provides the tool plug-ins that implement a Java IDE supporting the development of any Java application, including Eclipse plug-ins. It adds a Java project nature and Java perspective to the Eclipse Workbench as well as a number

of views, editors, wizards, builders, and code merging and refactoring tools. The JDT project allows Eclipse to be a development environment for itself.

Features include:

- Java projects with source files arranged in package directories
- Editing with keyword and syntax coloring, outline showing declaration structure
- Code formatter
- Refactoring
- Search
- Compare
- Compile - JCK-compliant Java compiler
- Run Java programs in a separate target Java virtual machine
- Debug programs with JPDA-compliant Java virtual machine

Android Source Code

The following step is optional.

During Android development it is very useful to have the Android source code available as Android uses a lot of defaults.

Haris Peco maintains plugins with provides access to the Android Source code.

Use the Eclipse update manager to install two of his plugins.

Create an Android Emulator Device

The Android tools include an emulator. This emulator behaves like a real Android device in most cases and allow you to test your application without having a real device. You can emulate one or several devices with different configurations. Each configuration is defined via an "Android Virtual Device" (AVD).

To define an AVD press the device manager button, press "New" and maintain the following.

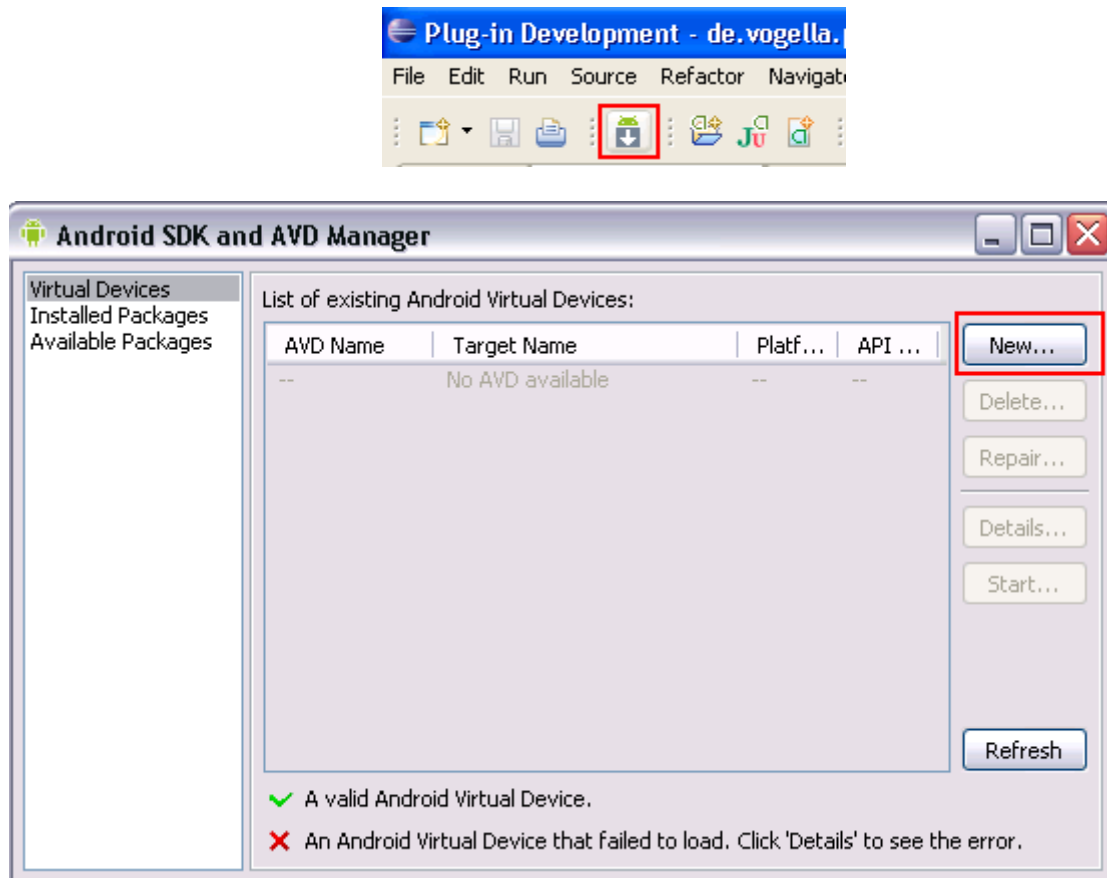


Figure:6.3.9:Android SDK and AVD Manager

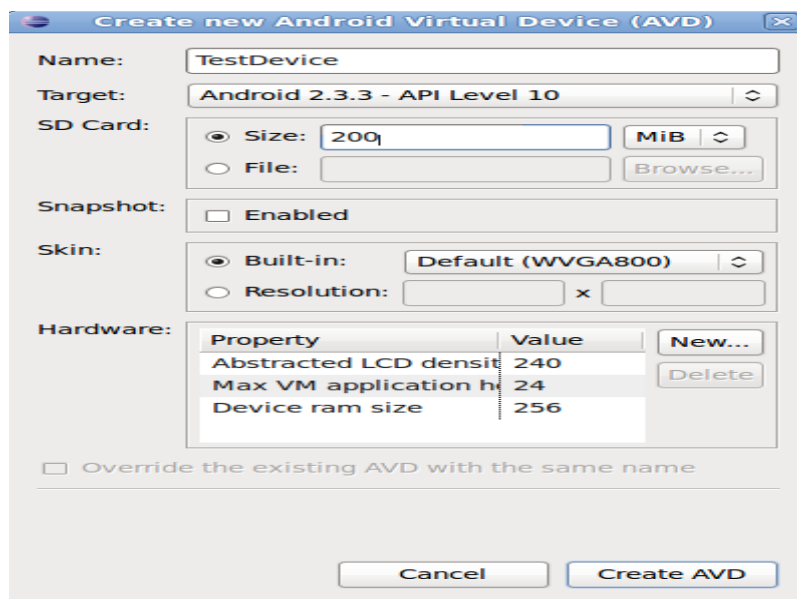


Figure:6.3.10:Create New AVD

Press "Create AVD". This will create the device and display it under the "Virtual devices". To test if your setup is correct, select your device and press "Start".



Figure:6.3.11:Out line Of Application

Error handling

Things are not always working as they should be. Several users report that get the following errors:

- Project ... is missing required source folder: 'gen'
- The project could not be built until build path errors are resolved.
- Unable to open class file R.java.

To solve this error, select from the menu Project -> Clean.

If you having problems with your own code, you can use the Log Cat viewer as described in Log Cat Viewer .

Out puts:

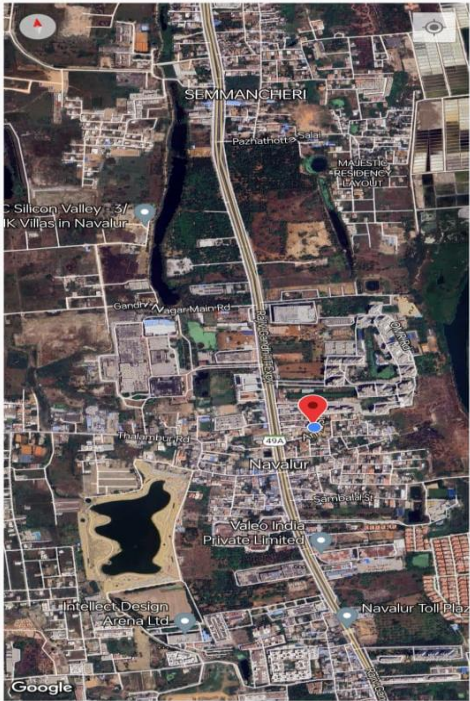


Figure:6.3.11: Location

Date :Wed Dec 27 12:20:08 GMT+05:30 2023 Number :AX-SBYONO Message :Activation Code for YONO Registration is 007675. This code is valid for 30 minutes. Do not share it
Date :Wed Dec 27 12:17:07 GMT+05:30 2023 Number :JK-SBYONO Message :The Reference Number to activate YONO is rAX5Je. Please visit SBI NARAYANGUDA branch within 7 days f
Date :Wed Dec 27 12:26:08 GMT+05:30 2023 Number :JK-SBYONO Message :Alert: Dear Customer, you have successfully set your MPIN for YONO SBI mobile banking app. If not do
Date :Wed Dec 27 12:26:02 GMT+05:30 2023 Number :AX-SBYONO Message :621233 is your OTP for registering YONO SBI mobile banking app. Do not share it with anyone. If not
Date :Wed Dec 27 12:21:59 GMT+05:30 2023 Number :AD-SBYONO Message :Dear Customer, YONO SBI app successfully enabled for mobile banking. If not done by you please call
Date :Wed Dec 27 12:13:48 GMT+05:30 2023 Number :+918976967382 Message :YONOREG 1d47303978c542b533df 2e77c68bedcdf2f90cb8eeb1037689c1fbb39a 8273bb57b9be9b91d5b70153140160939d
Date :Wed Dec 27 12:26:09 GMT+05:30 2023 Number :AD-SBYONO Message :Dear Customer, YONO SBI app successfully enabled for mobile banking. If not done by you please call
Date :Wed Dec 27 12:15:42 GMT+05:30 2023 Number :JK-SBYONO Message :454431 is your OTP for registering YONO SBI mobile banking app. Do not share it

Figure:6.3.11:Messages



Figure:6.3.11:Contacts



Figure:6.3.12:Geo Fencing Messages

CHAPTER-7

SYSTEM STUDY AND TESTING

7.1 Feasibility study

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

ECONOMICAL FEASIBILITY □ TECHNICAL FEASIBILITY □ SOCIAL
FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some

constructive criticism, which is welcomed, as he is the final user of the system.

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

7.2.Types of test & Test Cases UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

Valid Input: identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected. Functions : identified functions must be exercised.

Output: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

WHITE BOX TESTING

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

UNIT TESTING:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

INTEGRATION TESTINGSS

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g., components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER-8

CONCLUSION

In conclusion, this application is designed for locating missing children. The solution represented in this paper takes the advantages of smart phones which offers rich features like Google maps, GPS, SMS etc. Some of the best works implemented in past relies on SMS based tracking which is not helpful to get an accurate location in our proposed system we have provided real time tracking. We have added Geo-fencing and Emergency messaging services to enhance the system. In this system the parent can view the children's call's history and sms history Future enhancements for this child safety system could include integration with advanced artificial intelligence and machine learning algorithms to provide predictive threat detection, allowing the system to proactively identify potential risks to children. Additionally, incorporating wearable devices or IoT (Internet of Things) technology.

REFERENCES

- [1] Geo-Fencing, available at: <http://www.fieldtechnologiesonline.com>
- [2] Aditi Gupta; Vibhor Harit; Child Safety & Tracking Management System by Using GPS, Geo-Fencing & Android Application: An Analysis; 12-13 February 2016
- [3] Child Safety & Tracking Management System
- [4] Harshal Chaudhary; Dr. Ranjana Zinjore; Dr. Varsha Pathak ;Parent-Hook: A Child Tracking System based on Cloud URL; 30-31 October 2020
- [5] N. Senthamilarasi, N.Divya Bharathi, D.Ezhilarasi, R.B.Sangavi; Child Safety Monitoring System Based on IoT; 23/09/2023
- [6] Rita H. Pawade, Dr. Arun N. Gaikwad, "Android Based Children Tracking System", International Journal of Science, Engineering and Technology Research (IJSETR), Volume 4, Issue 6, June 2015, pg. 2088-2092
- [7] A.Al-Mazloun, E. Omer, M. F. A. Abdullah—GPS and SMS based children tracking system using smart phone. International Journal of Electrical, Robotics, Electronics and Communication Engineering Vol:7 No.2, 2013.
- [8] Ghaith Bader Al-Suwaidi, Mohamed Jamal Zemerly, "Locating friends and family using mobile phones with global positioning system (GPS)," IEEE/ACS International Conference on Computer Systems and Applications, 2009.
- [9] Anderson, Ruth E., et al., "Building a transportation information system using only GPS and basic SMS infrastructure," 2009 International Conference on Information and Communication Technologies and Development (ICTD), IEEE, 2009.
- [10] Almomani, I.M., Alkhalil, N.Y., Ahmad, E.M., Jodeh, R.M., "Ubiquitous GPS vehicle tracking and management system," 2011 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), pp.1-6, 6- 8 Dec. 2011.
- [11] CogniToys, "Cognitoys: Internet-connected smart toys that learn and grow," 2020. <https://www.kickstarter.com/projects/cognitoys/cognitoysinternet-connected-smart-toys-that-learn>.
- [12] P. C. K. Hung, L. Rafferty, and M. Fantinato, "Toy computing," in Encyc. of

Comp. Graphics and Games (N. Lee, ed.), Springer, 2019.

[13] G. Disterer and C. Kleiner, "BYOD – Bring your own device," *Procedia Technology*, vol. 9, pp. 43–53, 2013.

[14] W. Stallings and L. Brown, *Computer Security: Principles and Practice*. Pearson, 4th ed., 2008.

[15] L. Liu and M. T. Özsu, eds., *Encyclopedia of Database Systems*, vol. 6. Springer, 2009.

[16] D. A. Hertz, "Don't talk to strangers: An analysis of government and industry efforts to protect a child's privacy online," *Federal Communications Law J.*, vol. 52, no. 2, pp. 429–451, 2000.

[17] D. Holloway and L. Green, "The internet of toys," *Comm. Res. and Prac.*, pp. 1–14, 2016.

[18] Canada, "PIPEDA – Personal Information Protection and Electronic Documents Act," 2018. Minister of Justice, Canada, <https://lawslois.justice.gc.ca/PDF/P-8.6.pdf>.

[19] EU, "GDPR – General Data Protection Regulation," 2016. Eur. Parliament and Council of the Eur. Union, <https://eugdpr.org>.

[20] N. Vollmer, "Article 12 EU General Data Protection Regulation (EUGDPR). privacy/privazy according to plan," 2019. <http://www.privacyregulation.eu/en/article-12-transparent-information-communication-and-modalities-for-the-exercise-of-the-rights-of-the-data-subject-GDPR.htm>.

[21] P. C. K. Hung, M. Fantinato, and J. Roa, "Children privacy protection," in *Encyc. of Comp. Graphics and Games* (N. Lee, ed.), Springer, 2019.

[22] D. Boyd, E. Hargittai, J. Schultz, and J. Palfrey, "Why parents help their children lie to facebook about age: Unintended consequences of the 'children's online privacy protection act'," *Fir. Mond.*, vol. 16, 2011.

CONFERENCE FULL PAPER

