**CONTENTS**

**PAGE INDEX**

**No. Topic**  **Page No.**

**PAGE INDEX I-II**

**TABLE INDEX III**

**FIGURE INDEX IV-V**

**ABSTRACT VI**

1. **INTRODUCTION 1-4**

**1.1 Existing Solutions 2**

**1.2 Proposed System 3**

**1.3 Scope 4**

**2. MOBILE PHONE APPLICATION DEVELOPMENT 5-13**

**2.1 Why Android? 6**

**2.1.1 Android Architecture 7**

**2.1.2 Application Framework 8**

**2.1.3 Libraries 9**

**2.1.4 Android Runtime 10**

**2.1.5 Linux Kernel 10**

**2.2 Android Development 11**

**2.2.1 Android SDK and Eclipse 12**

**2.2.2 Android Debug Bridge 12**

**3. SYSTEM SPECIFICATION 14-21**

**3.1 System Requirements Elicitation 15**

**3.1.1 Software Requirements 15**

**3.1.2 Hardware Requirements 15**

**3.1.3 Application Requirements 16**

**3.1.4 Non-Functional Requirements 16**

**PAGE INDEX**

**No. Topic**  **Page No.**

**3.2 System Components Overview 17**

**3.2.1 Asset Management 17**

**3.2.2 Location Management 17**

**3.2.3 Data Management 18**

**3.2.4 Tracker Management 19**

**3.3 System Architecture 21**

**4. SYSTEM DESIGN 22-29**

**4.1 Design Overview 23**

**4.2 System Design Model 24**

**4.2.1 Use case Diagram 24**

**4.2.2 Class Diagram** **25**

**4.2.3 Sequence Diagram 25**

**4.2.4 Collaboration Diagram 26**

**4.2.5 Activity Diagram 27**

**4.2.6 State Chart Diagram 28**

**4.2.7 Deployment Diagram 29**

**5. SYSTEM IMPLEMENTATION 30-43**

**5.1 Sample Code 31**

**5.2 User Interface (Screenshots) 32**

**6. TESTING 44-47**

**6.1 Unit Testing 46**

**6.2 Integration Testing 46**

**6.3 JUnit Testing 47**

**6.4 Tracker Communication Testing 47**

**7. CONCLUSION 48-49**

**8. BIBLIOGRAPHY**

**TABLE INDEX**

**No. Table Page No.**

**2.1 Mobile OS Comparison 8**

**2.2 Commonly Used abd Commands 13**

**3.1 NMEA Fields 20**

**6.1 Unit Testing 46**

**FIGURE INDEX**

**No. Figure Page No.**

**1.1 Existing System 2**

**1.2 Proposed System 3**

**2.1 Android Architecture 7**

**2.2 Activity Life Cycle 11**

**3.1 Mock-up of Application 18**

**3.2 System Architecture 21**

**4.1 Use Case Diagram 24**

**4.2 Class Diagram 25**

**4.3 Sequence Diagram 26**

**4.4 Collaboration Diagram 26**

**4.5 Activity Diagram 27**

**4.6 State Chart Diagram 28**

**4.7 Deployment Diagram 29**

**5.1 Service Enabled 32**

**5.2 Service Disabled 33**

**5.3 Application Enabled with Pass code 34**

**5.4 SMS with Location of Asset Sent to User 35**

**5.5 Location of Asset 36**

**5.6 Application Disabled 37**

**5.7 App Settings Turned OFF 38**

**FIGURE INDEX**

**No. Figure Page No.**

**5.8 Error SMS Sent to User 39**

**5.9 Application Enabled with New Pass code 40**

**5.10 App Settings Turned ON 41**

**5.11 New Location Link Sent to User 42**

**5.12 Location of Asset on Google Maps 43**

**ABSTRACT**

*As security increasingly becomes not just the concern of security institutions but also that of small companies and individuals, there is a need to offer a solution for everyday common problems of monitoring and tracking valuable assets such as cars, pets, motor-bikes and other valuables. This solution has to be cheap, convenient and inexpensive to be adopted by the general mass of people.*

*This project mainly focuses on developing an Android app which facilitates monitoring and tracking of valuable assets such as cars, pets, motor-bikes and other valuables. In this project GPS tracker (GPS supporting Android Device) uses GPS to determine the precise location information (latitude and longitude) and transmit it to the receiver (Android phone) using SMS service, then the receiver android phone extracts the longitude and latitude from the last incoming SMS message and display that location in Google maps, pinpointing the exact location. This allows the asset's location to be displayed against a map backdrop in real time. Purpose of this simple application avoids manual extraction of latitude; longitude coordinates from the SMS and typing "maps.google.com" by user and then textual typing the longitude and latitude to get visual location on the map.*

*KEYWORDS****:*** *Android, GPS, Google Maps, SMS*