

TABLE OF CONTENTS

1.	Introduction		7
	1.1.	Motivation	8
	1.2.	Problem Definition	8
	1.3.	Relevance of the Project	9
	1.4.	Methodology used	9
	1.4.1.	Android	9
	1.4.2.	PHP	10
	1.4.3.	PHP vs HTML	10
	1.4.4.	Android Application Framework	11
2.	Literature Survey		15
	2.1.	Research Papers	16
3.	Requirement		18
	3.1.	Functional Requirement	19
	3.2.	Non-Functional Requirement	19
	3.3.	Constraints	20
	3.4.	Hardware and Software Requirements	20
	3.5.	System Block Diagram	20
4.	Proposed Design		21
	4.1.	System Design /Conceptual Design (Architectural)	22
	4.2.	Detailed Design	23
	4.2.1.	Activity Diagram	23
	4.2.2.	DFD	25
	4.2.3.	Sequence Diagram	27
	4.3.	Plan of Work	28
	4.4.	Project Scheduling &Tracking using Timeline	29
5.	Results and Discussions		30
6.	Plan of Action for the Next Semester		32
	6.1.	Website Development	33
	6.2.	Prototype Development	33
	6.3.	Testing	33
	6.4.	Delivery and Feedback	33
7.	Conclusion		34
8.	References		36
9.	Appendix		37

CHAPTER -1

INTRODUCTION

1. Introduction

1.1 Motivation

Most of the organizations these days provide many resources to their employees in a hope that it will help the organization to grow and it will make the employees happy. One of such resource which the organization provides is a mobile device. But there have been cases of employees misusing the Android Smartphone provided to them. This misuse of the devices for any personal use can bring losses to the organization thereby defying the purpose of providing the smartphone. Hence it is necessary to monitor the devices with/without the employees knowing about it. So that any necessary action can be taken against such employees. MackDroid aims at achieving this goal by providing an application which works in a stealth mode and collects and transfers the usage data such as call logs, messages, WhatsApp to the server which the respective employees can access anytime.

1.2 Problem Definition

MackDroid is an android application which will collect all the information from the user's device as soon as it is installed properly with all the permissions granted by the user. This data will be stored on a remotely located server for monitoring and analysis of the activities performed on the device by the respective people from the organization who have to keep a watch on any misuse of company resources.

Information Collected includes: Location History, Contacts, and Call logs, Text Messages, Photos, Video Files, Browser History and Emails. The permissions requests include permissions to access location history, Call records, network communication, SMS records, and notifications access etc. This information will be accessed only by the system administrator and the licensed authority who has the privilege to monitor the data collected from the device where the Android application is installed. The application will have two modes of installation namely stealth mode and without stealth mode. When the application is installed with Stealth mode, the targeted device will have no trace of the application installed. The processes will be run in multithreading system where multiple processes will function according to the task specified to it.

1.3 Relevance of the project

This application will be mainly used by the organizations who want to track the mobile device activity of the employees in the organization. As the statistical analysis suggests that Android shares the major part of smartphone market, this application focuses on making it possible for the employer to prevent misuse of the device in any form of criminal or personal activity. Apart from this, the application can be used by crime investigators and police department for surveillance over suspected criminals. This would help them keep a track over any criminal activity as well as advantageous to them in getting evidences against criminals. In another way, the application might also help in preventing crimes since the Police and surveillance team would be well aware in advance about any plans of disturbing human peace. The application will not be available for commercial use as it can be misused to cause identity theft and intellectual property theft. Therefore, it will only be made available through licensing to renowned organizations.

1.4 Methodology Used

1.4.1 Android:

Android is a mobile operating system developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices. Android has the largest installing base of all operating systems of any kind. Android is an open-source OS. Android's default user interface is mainly based on direct manipulation using touch inputs that loosely correspond to real-world actions such as tapping, pinching or reverse pinching in order to manipulate on-screen objects. The response to the user input is designed to be immediate and provides fluid touch interface, often using the vibration capabilities of the device to provide full haptic feedback to the user. Android applications (apps) which extend the functionality of devices, are written in Android Software Development (SDK) kit. The SDK includes a comprehensive set of development tools, including a debugger, software libraries, a handset emulator, documentation, samples and tutorials. Android APK will be developed for this application which will be linked to a database stored on a server where the data collected from the devices that have the Android APK installed will be stored and managed by the system administrator.

1.4.2 PHP:

PHP (Hypertext Preprocessor) is a server side scripting language designed primarily for web development. PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by PHP interpreter and is implemented as a module in the web server. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

1.4.3 PHP vs HTML:

HTML is a language used to describe to a browser how to display text and other objects in a browser window. It is not a programming language. HTML works on a client computer (the system on which the page is being viewed).

PHP is a scripting language, and can be used to create web pages written in HTML. PHP runs on the server (the system from which the page comes), and is a full-fledged programming language. PHP is a server side scripting language and hence it executes on the server and only the output is downloaded on the client's computer. PHP can generate a lot more than HTML. It can do things like upload files, generate images (with the GD library), send email, write to your server's filesystem and execute code, among other things. PHP really comes in useful when you're working with a database. For example, you can store all of your user accounts in a database and when a member enters the correct username and password your PHP program/script will fetch the relevant user details from the database and display them in the user's browser. With html it's only really possible for everyone to see the same page, the page won't look different or display different information depending who's looking at it, everyone pretty much gets the same. PHP is a server-side scripting language for web development. After the PHP code is interpreted and executed, the web server sends resulting output to its client, usually in form of a part of the generated web page; for example, PHP code can generate a web page's HTML code, an image, or some other data. You can see the HTML design code of any webpage you want (For Google Chrome, just right click on web page and click on "View page source"). But, you cannot see the PHP code!! (Because PHP is a Server side language).

1.4.4 Android Application Framework

Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram.

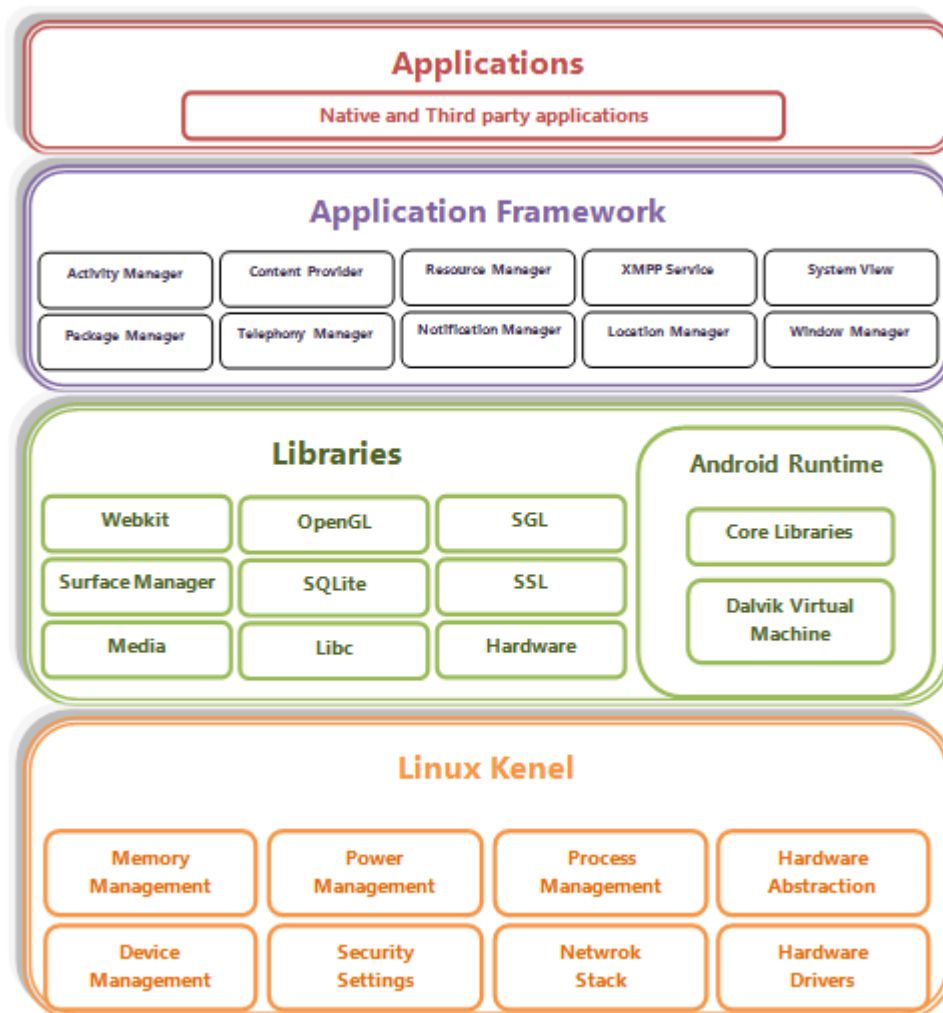


Figure : Android Architecture Stack

Figure 1: Android Framework

i. **Linux Kernel:**

At the bottom of the layers is Linux - Linux 3.6 with approximately 115 patches. This provides a level of abstraction between the device hardware and it contains all the essential hardware drivers like camera, keypad, display etc. Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

ii. Libraries:

On top of Linux kernel there is a set of libraries including open-source Web browser engine Web Kit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security etc.

iii. Android Libraries:

This category encompasses those Java-based libraries that are specific to Android development. Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing and database access. A summary of some key core Android libraries available to the Android developer is as follows –

android.app – Provides access to the application model and is the cornerstone of all Android applications.

android.content – Facilitates content access, publishing and messaging between applications and application components.

android.database – Used to access data published by content providers and includes SQLite database management classes.

android.opengl – A Java interface to the OpenGL ES 3D graphics rendering API.

android.os – Provides applications with access to standard operating system services including messages, system services and inter-process communication.

android.text – Used to render and manipulate text on a device display.

android.view – the fundamental building blocks of application user interfaces.

android.widget – A rich collection of pre-built user interface components such as buttons, labels, list views, layout managers, radio buttons etc.

android.webkit – A set of classes intended to allow web-browsing capabilities to be built into applications.

iv. Android Runtime:

This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called **Dalvik Virtual Machine** which is a kind of Java Virtual Machine specially designed and optimized for Android. The Dalvik VM makes use of Linux core features like memory management and multi-threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine. The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

v. Application Framework:

The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.

The Android framework includes the following key services –

Activity Manager – Controls all aspects of the application lifecycle and activity stack.

Content Providers – Allows applications to publish and share data with other applications.

Resource Manager – Provides access to non-code embedded resources such as strings, colour settings and user interface layouts.

Notifications Manager – Allows applications to display alerts and notifications to the user.

View System – an extensible set of views used to create application user interfaces.

vi. Applications:

One can find all the Android application at the top layer. One can write an application to be installed on this layer only. Examples of such applications are Contacts Books, Browser, and Games etc.

CHAPTER -2

LITERATURE

SURVEY

2. Literature Survey

2.1 Research Papers

a. Android Applications and Security Breach

Previous work: (<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7529383>) emphasizes on breaching the loopholes in Android which is based on Linux kernel. The vulnerabilities in the kernel are very much universal in terms of exploitation by the attackers. It proposes to develop an application to exploit this vulnerability by developing an application containing a malware which runs legally on the device using permissions. The application will ask for permissions and by seeking all the permissions. This application runs a Trojan known as Remote Access Trojan aka RAT which steals the information such as call records, photos, videos and SMS. The Trojan will be a normal APK file that can be installed on Android device using Package Installer. In the source code of this application, there is a section where the attacker configures the URL of the C&C server, it is encoded with base64, the password is for the database that will be uploaded when the server receive information of the victim. The backup URL is in case the server were unavailable. Hence, this android application can potentially collect private information from the devices.

Drawback: Since, this application asks for all the permissions that are provided by the Android architecture, the detection of such malwares in Android device can be easily found out by checking the list of permissions and categorizing them into categories provided by the (Parvez Faruki, Ammar Bharmal, Vijay Laxmi, Vijay Ganmoor, Manoj Singh Gaur, Mauro Conti and Muttukrishnan Rajarajan, «Android Security: A Survey of Issues, Malware Penetration and Defenses», IEEE Communications Surveys & Tutorials 2015, Volume: 17, Issue: 2 pp. 998 – 1022) in the paper. The 4 categories given are Normal, Dangerous, Signature and System permissions.

Proposed Solution: The detection of this malware can be verily made tougher by using and seeking only the set of permissions that are required for the application to perform a specific task. At no given point of time, the application will seek all permissions as the processes running for different tasks will be different. Thereby, no process will have full permissions.

- b. Mobile Activity Monitoring System Using Android Spy (International Journal of Advanced Research in Computer and Communication Engineering):

Mobile activity monitoring system uses android as front end and MySQL as back end. This system sends all the call details and SMS details to the administrator as an alert as soon as the activity is performed by the android user. Location of the user can be checked anytime. And if the user crosses any geographical area, the administrator is alerted.

Drawbacks: The system alerts the administrator on each and every activity of the user, this can result in many redundant data and this data will be waste as it is of no use to the administrator if the user is not doing anything unethical. The system sends the message alerts from the user's device itself, this will lead to the user coming to know about the application and hence authentic data about the activity will not be available. Another shortcoming of this system is that it will be available for use to everyone which can lead to the unethical use of this system. The system discussed in this paper tracks the calls, SMS and location, but nowadays with growing popularity of IMs and Internet voice and video call services, this system might fail.

Proposed Solution: To overcome the redundancy of the unwanted data, the administrator can be given an option to set certain triggers. When these triggers are executed, the administrator will be alerted.

Our proposed system works in a stealth mode and hence no alerts are sent from the user's device itself. The administrator is alerted through emails from the server if need be.

To prevent the unethical distribution and usage of this system, our solution can be installed with a valid purchased license only.

Our proposed system not only tracks the Calls and SMS, it also tracks the WhatsApp messages.

- c. Employee Monitoring System Using Android Smart Phone (International Journal on Recent and Innovation Trends in Computing and Communication).

An employee monitoring system already exists that is used to track the activities of the employees in the organization. It stores calls, messages, pictures, videos, browser cache etc. on a central database which can be accessed by the manager. This application forms a secure connection between the devices using AES method. In this system, the device of the employee is connected to a 3g terminal so that the data is transferred to the

manager application. The manager can therefore check the logs and activities to know if the employee has misused the device provided by the employer.

Drawback: This requires the two devices of the manager and employee to be paired in order to exchange data. In order to overcome the limitations imposed by Bluetooth and Wi-Fi connections, this works on pairing between two devices on a web server. Also, whenever the data is to be transferred from one application to another, employee's device is connected by a 3g terminal by the employer and only then the data can be stored and transferred on to the server.

Solution: Since, this can alert the employee about the tracking of the device data, employee can use various root applications in order to manipulate the data that is stored on the centralized database. Therefore, we propose to develop a solution which needs no authorization or pairing from the employee's device. Infact, our application will have a stealth mode feature that prevents the employee from having any knowledge about the malware installed on the device. This helps to get the actual results and data from the employee's data.

CHAPTER -3

REQUIREMENT

3. Requirements

3.1 Functional Requirements

- A stand-alone android application to collect data from the user's device and store remotely.
- The above client application can be used on android version 4.4 and above and should run fine.
- This application will be licensed application and will be provided to the Organizations as per their need. Every organization will have one unique profile assigned which will have all the information of the employees of that Organization. (Licensing is a method by which only the authorized users will be licensed to use the application. These authorized personals will have to purchase these licenses from time to time. Licensing is done by the Google Play Licensing server).
- The stored data should be platform independent to be sent to the remote server.
- The remote server should be able to collect data from various application simultaneously and store in appropriate folder of each user.
- When connected to network, the client app should authenticate into a central server automatically and submit all the collected information.
- The central server should verify the user and the information header before uploading the data to the database.
- The information of each data upload like the user, the time of upload etc. should be captured.

3.2 Non-Functional Requirements

- The data stored can be encrypted to protect from other Organisation profiles or by server theft.
- Validating the platform independent format before inserting it into the centralized server (database).
- This application will have another mode of operation (Stealth Mode) where the application will not be visible in the application manager of the device and will show no RAM Usage.
- The performance with the server website will be high enough to handle simultaneous transactions by the users.

3.3 Constraints

- The mobile device will be unable to send the data to the server in case of no internet connection.
- The application will not work properly if it is not granted all the permissions.
- The application will automatically get deleted once phone storage is erased using Factory Data Reset.

3.4 Hardware and Software Requirements

3.4.1 Hardware:

- a. Mobile Device
- b. Server (RAM at least 8 GB)
- c. Backup Server

3.4.2 Software:

- a. Android OS
- b. Web Browser

3.5 System Block Diagram

As seen in the figure below MackDroid has a connection with remote server and local database which resides inside the application. Users are connected to the server. All of the users will have a unique ID and password for authentication. When a user successfully logs in the web portal, he will be able to see a list of employees, which when selected will show the activities of the employee.

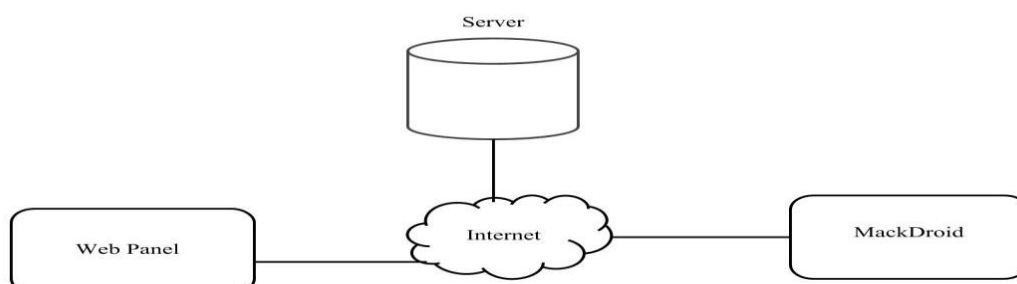


Fig.2 System Block Diagram

CHAPTER - 4

PROPOSED

DESIGN

4. Proposed Design

4.1 System Design

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

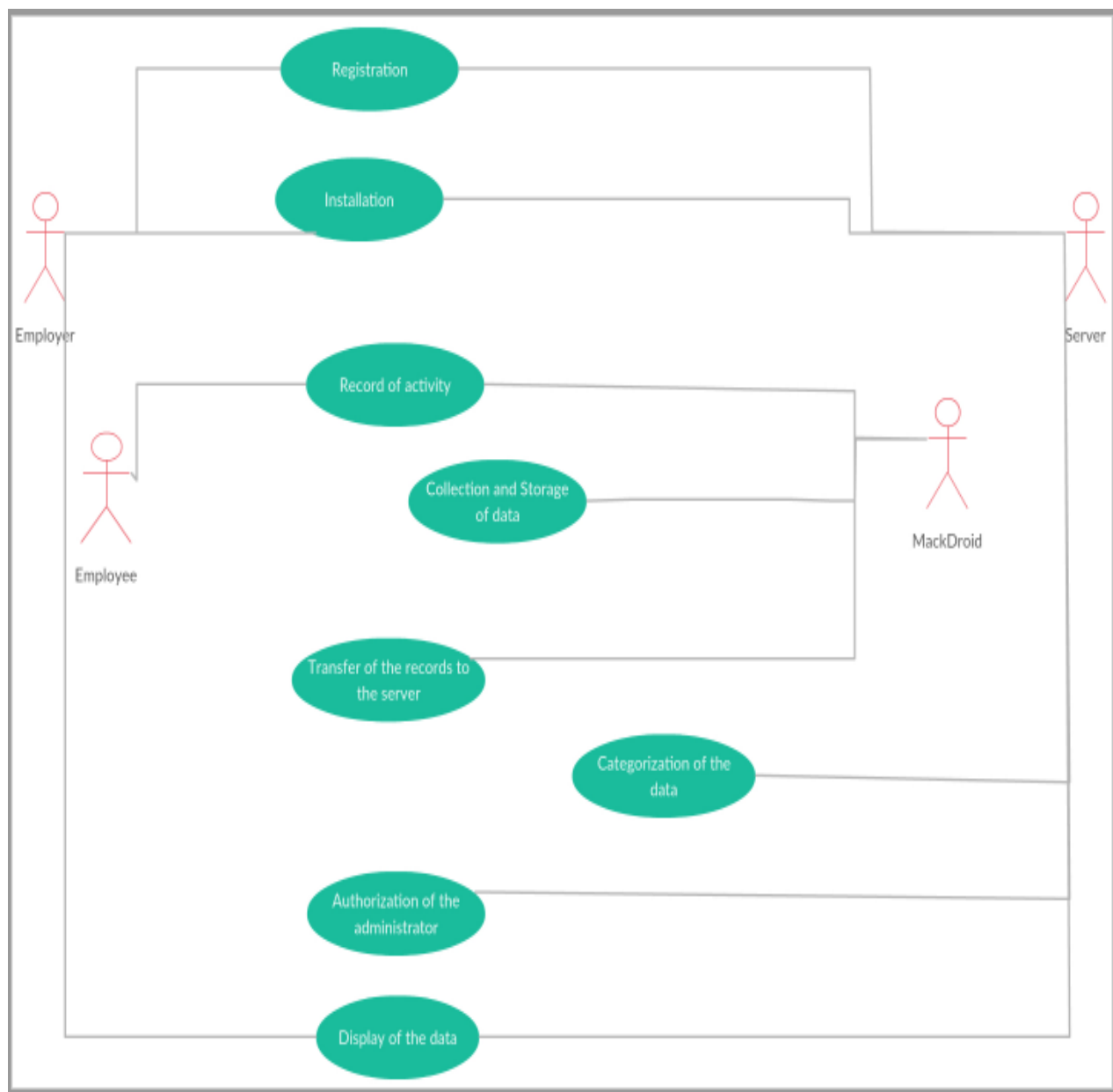


Fig.3 Use Case

4.2 Detailed Design

4.2.1 Activity Diagram

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow control by using different elements like fork, join etc.

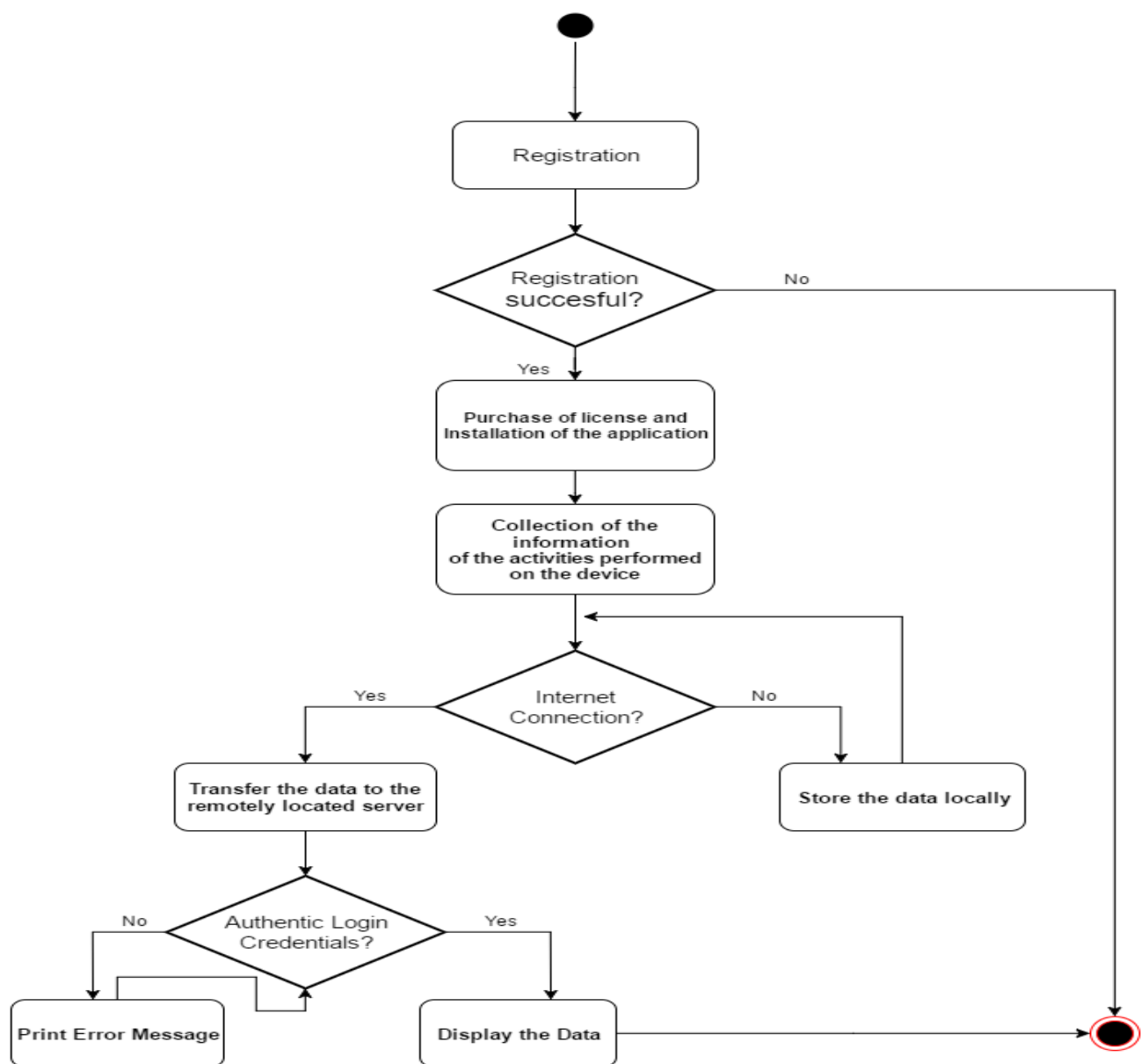


Fig.4 Activity Diagram

4.2.2 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. Following is DFD's for our System; it has 2 levels, level 0 and level 1.

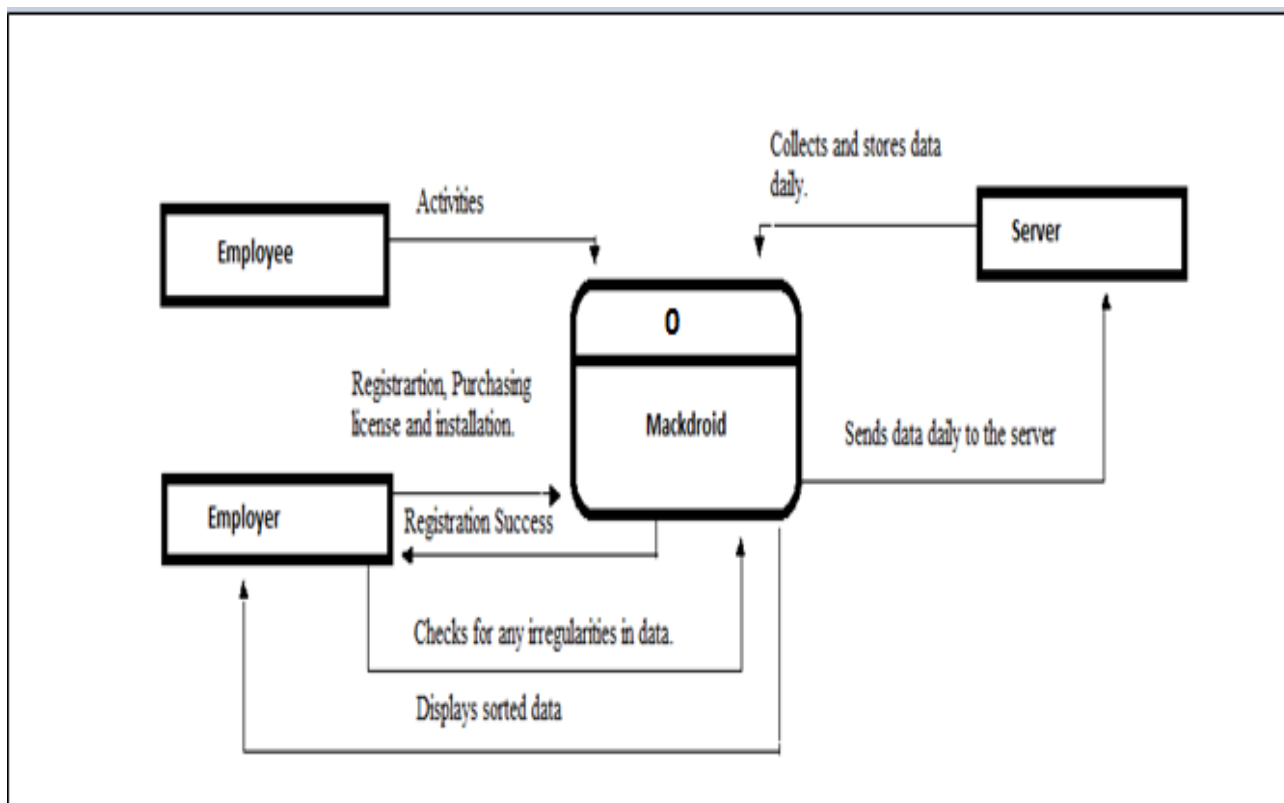


Fig.5 Data Flow Diagram (Level-0)

It consists of a single process called MackDroid and there are 3 external agents, the employee, the employer and the server. The employee gives input in form of records of activities.

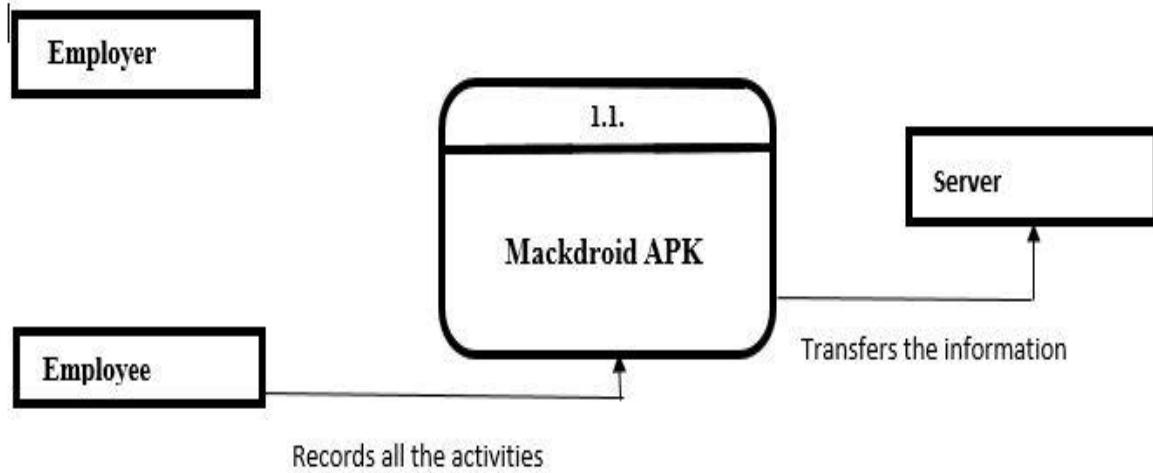


Fig.6 Data Flow Diagram (Level-1.1)

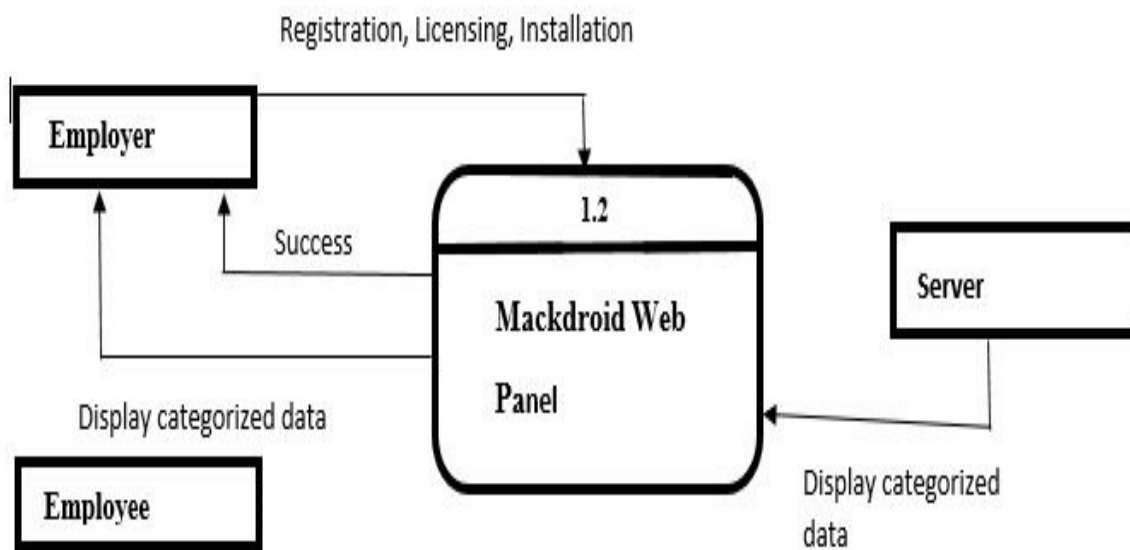


Fig.7 Data Flow Diagram (Level-1.2)

4.2.3 Sequence Diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Following is sequence diagram for our system: -

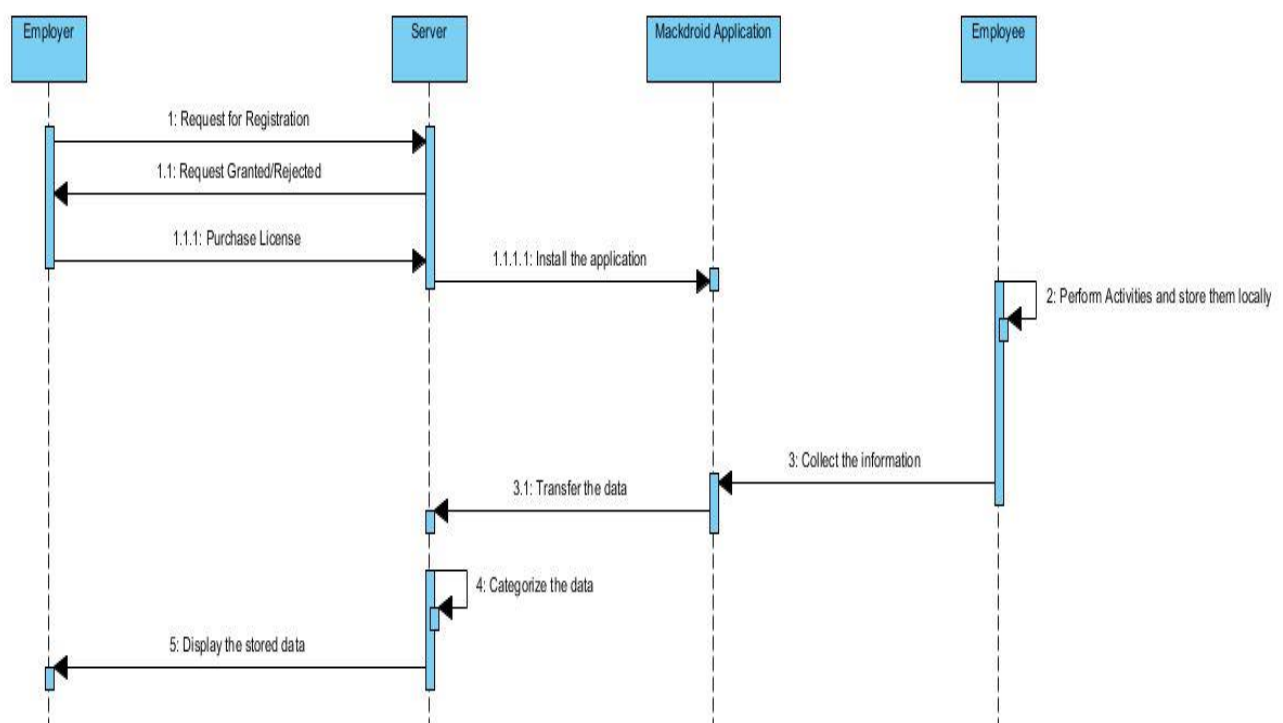


Fig. 8 Sequence Diagram

4.3 Plan of Work:

Task Name	Start	Finish
1 MACKDROID	Tue 19-07-16	Thu 02-03-17
1.1 Requirement Analysis	Tue 19-07-16	Wed 10-08-16
1.1.1 Requirement Gathering	Tue 19-07-16	Wed 27-07-16
1.1.2 Determine Objective and Scope	Thu 28-07-16	Tue 02-08-16
1.1.3 Requirement Review	Wed 03-08-16	Wed 10-08-16
1.2 Planning & Analysis	Thu 11-08-16	Wed 07-09-16
1.2.1 Task Identification	Thu 11-08-16	Mon 22-08-16
1.2.2 Setting Deadlines and Milestones	Tue 23-08-16	Wed 31-08-16
1.2.3 Budget Estimate	Thu 01-09-16	Tue 06-09-16
1.2.4 Task Assignment	Thu 01-09-16	Wed 07-09-16
1.3 Designing	Thu 03-11-16	Tue 29-11-16
1.3.1 GUI Design	Thu 03-11-16	Fri 18-11-16
1.3.2 Design Review	Mon 21-11-16	Mon 28-11-16
1.3.3 Design Finalised	Tue 29-11-16	Tue 29-11-16
1.4 Implementation	Thu 08-09-16	Fri 13-01-17
1.4.1 Coding	Thu 08-09-16	Fri 13-01-17
1.4.1.1 Android	Thu 08-09-16	Wed 12-10-16
1.4.1.2 APK completed	Thu 13-10-16	Thu 13-10-16
1.4.1.3 PHP	Tue 29-11-16	Thu 12-01-17
1.4.1.4 Website Developed	Fri 13-01-17	Fri 13-01-17
1.5 Database Implementation	Thu 13-10-16	Wed 02-11-16
1.6 Integration and Prototype	Fri 13-01-17	Mon 30-01-17
1.7 Testing	Thu 08-09-16	Mon 20-02-17
1.7.1 Test Plan and Procedure	Thu 08-09-16	Wed 21-09-16
1.7.2 Unit Testing	Thu 22-09-16	Thu 12-01-17
1.7.3 Integration Testing	Thu 22-09-16	Tue 17-01-17
1.7.4 System Testing	Wed 18-01-17	Mon 30-01-17
1.7.5 Alpha Testing	Tue 31-01-17	Thu 02-02-17
1.7.6 Beta Testing	Fri 03-02-17	Mon 20-02-17
1.8 Documentation	Thu 08-09-16	Wed 01-03-17
1.9 Delivery & Feedback	Thu 02-03-17	Thu 02-03-17

4.4 Project Scheduling and Tracking using Time Line

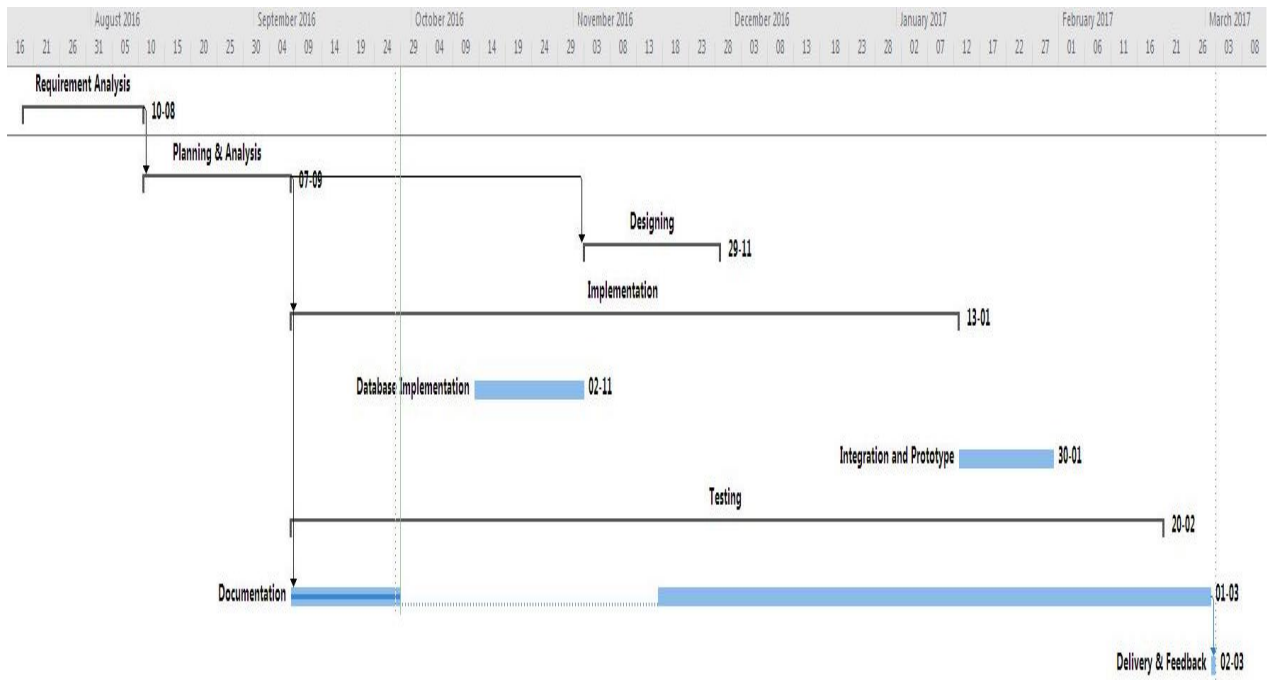


Fig.9 TimeLine Chart

CHAPTER -5

RESULTS &

DISCUSSIONS

5. Results and Discussion

The MackDroid web portal can be implemented using various other languages like HTML or .NET. By weighing all the options, we found out that it is best to use PHP for our server as PHP is advantageous as compared to other languages for implementation of database.

Also as Android is more popular and widely used than Apple OS, it is efficient to launch the Android app first.

CHAPTER -6

PLAN OF ACTION

FOR NEXT

SEMESTER

6. Plan of Action for Next Semester

6.1. Website Development

The main development of website will start in this phase. The website will be developed using PHP scripting language.

6.2. Prototype Development

Prototype Development phase will include a development of prototype for the system. This prototype will help to understand the system better and will help to find out the bugs and errors in the system.

6.3. Testing

This phase will include testing of the system using various methods. The results yielded by this phase will help to improve the product before it hits the market.

6.4. Delivery and Feedback

The final product when developed will be given to the intended users. Feedback will be collected from the users for further improvement of the system.

CHAPTER -7

CONCLUSION

7. Conclusion

7.1. Limitations

1. The application has to be installed on the device directly or in stealth mode using a binder.
2. The system administrator will not be able to access the data until and unless the device is connected to the internet and the data is stored on the server.
3. Factory Data Reset will erase the application from the phone automatically and therefore no further data will be stored on the server.
4. The application will run only on Android Platform with all permissions granted to various processes.

7.2. Future Scope

1. System administrator should be able to install this application on various other platforms such as Windows, iOS etc.
2. Remote access to the device which enables system administrator to handle and modify the device settings and functions.
3. Front camera access through to the handler allowing him to capture images at any point of time.
4. Ability to create a chat box between the handler and the mobile device without any access to the mobile device user. Therefore, mobile user will be only able to reply and read the messages.

7.3. Inference

After thorough analysis and implementation, it can be understood that such an application will be very useful in the fields on investigation, surveillance, monitoring and gaining remote access to other android devices. Since, the application can be misused to attempt identity threat and intellectual property theft, it can only be provided to licensed users. This application can prevent various crimes taking place and can prove to be very handy in maintaining peace and security of the nation.

References

Journal Papers:

1. Diana Gabriela Noemí Benítez-Mejía, Gabriel Sánchez-Pérez, Linda Karina Toscano-Medina “Android Applications and Security Breach” Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC), 2016 Third International Conference
2. Nitin P. Jagtap, Kanchan A. Patil, Shaziya Sayyed Shakil, Nitin S. Ingle “Mobile Activity Monitoring System Using Android Spy” International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 2, February 2015.
3. Ashwini Jaybhaye , Prajakta Kokare , Bhakti Toradmal, Tanmay Kulkarni “Employee Monitoring System Using Android Smartphone” International Engineering Research Journal (IERJ) Volume 1 Issue 2 Page 32-35, 2015.

Books:

1. Fei Hu “Big Data: Storage, Sharing and Security” CRC Press, 27-Apr-2016.
2. Mirosław Kutylowski, Jaideep Vaidya “Computer Security - ESORICS 2014” 19th European Symposium on Research in Computer Security, Wrocław, Poland, September 7-11, 2014

9. Appendix

a. List of Figures

Figure Number	Heading	Page no.
1	Android Framework	09
2	System Block Diagram	17
3	Use Case Diagram	19
4	Activity Diagram	20
5	Data Flow Diagram: Level 0	21
6	Data Flow Diagram: Level 1.1	22
7	Data Flow Diagram: Level 1.2	22
8	Sequence Diagram	23
9	Timeline Chart	25