Webkiosk Android Application

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

> in Computer Science and Engineering By Rajat Kumar (141371) Rishabh Verma (141353) Akhil Kumar (141378)

> > Under the supervision of Amit Jakhar

to



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report entitled "Webkiosk Android Application" in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering/submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2017 to December 2017 under the supervision of Mr Amit Jakhar.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Rajat Kumar (141371) Rishabh Verma (141353) Akhil Kumar (141378)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Mr.Amit Jakhar Professor in Department of Computer Science and Engineering

Dated: 15th May 2018

ACKNOWLEDGEMENT

I express my sincere thanks and indebtedness to my esteemed institution **Jaypee University Of Information Technology, Waknaghat** which has provided me with an opportunity to fulfill my desire and reach my goal.

First, I would like to thank **Mr. Amit Jakhar** Faculty member at **Jaypee University Of Information Technology** for his help in providing all the facilities and guidance required for project work.

On a moral personal note, my deepest appreciation and gratitude to my beloved parents, who have been a fountain of inspiration and have provided unrelenting encouragement and support. I heartily thank all those who have helped me directly or indirectly in the successful completion of this project.

Rajat kumar(141371) Rishabh Verma(141353) Akhil Kumar(141378)

Table Of Contents

1.0 INTRODUCTION

| 1.1 Abstract | 1 |
|-------------------|---|
| 1.2Objective | 1 |
| 1.3 Methodologies | 2 |

2.0 Literature Survey

| 2.1 Android | 3 |
|--------------------------|-----|
| 2.2 Android Architecture | 4 |
| 2.3 JSON | 10 |
| 2.4 PHP | 13 |
| 2.5 SQLite Database | .18 |
| 2.6 Android Studio | .19 |
| 2.7 API | .19 |
| 2.8 Amazon API Gateway | 22 |

3.0 System Development

| 3.1 Analysis | 23 |
|--------------|----|
| 3.2 Design. | |

| 4.0 Performance Analysis | 31 |
|--------------------------|----|
| 5.0 Conclusion | 41 |
| 6.0 References | 42 |

LIST OF FIGURES

| S. NO. | FIG NO | FIGURE NAME | PAGE NO. |
|--------|--------|--------------------------------|----------|
| 1 | 2.1 | Android Architecture | 6 |
| 2 | 2.2 | Android Framework | 7 |
| 3 | 2.3 | JSON Object | 10 |
| 4 | 2.4 | JSON Array | 11 |
| 5 | 2.5 | JSON Value | 11 |
| 6 | 2.6 | JSON String | 12 |
| 7 | 2.7 | JSON Number | 12 |
| 8 | 2.8 | PHP Flowchart | 16 |
| 9 | 2.9 | PHP Websites | 18 |
| 10 | 2.10 | API Calls | 23 |
| 11 | 3.1 | Design Flowchart | 26 |
| 12 | 3.1.1 | Flow Diagram | 27 |
| 13 | 3.1.1 | Use case diagram | 28 |
| 14 | 3.1.2 | ER model | 29 |
| 15 | 3.1.3 | Activity Diagram | 30 |
| 16 | 4.1 | P.A. of Inputs | 31 |
| 17 | 4.2 | P.A. of Attendance View | 31 |
| 18 | 4.3 | P.A. of Detail Attendance View | 32 |
| 19 | 4.4 | P.A. of Attendance Predictor | 32 |
| 20 | 4.5 | P.A. of Student Info | 33 |
| 21 | 4.6 | P.A. of Time Table | 33 |
| 22 | 4.7 | P.A. of Fragment switch | 34 |

| 23 | 4.8 | P.A. of Date Sheet activity switch | 34 |
|----|------|------------------------------------|----|
| 24 | 4.9 | P.A. of select year | 35 |
| 25 | 4.10 | P.A. of view date sheet | 35 |
| 26 | 4.11 | P.A. of subject reg. activity | 36 |
| 27 | 4.12 | P.A. of logout | 36 |
| 28 | 4.13 | Login | 37 |
| 29 | 4.14 | Overview attendance | 37 |
| 30 | 4.15 | Open navigation drawer | 37 |
| 31 | 4.16 | Detail attendance | 38 |
| 32 | 4.17 | Attendance predictor | 38 |
| 33 | 4.18 | Time table | 39 |
| 34 | 4.19 | Student Information | 39 |
| 35 | 4.20 | Date Sheet | 40 |
| 36 | 4.21 | Subject Registered | 40 |
| 37 | 4.22 | Logout | 41 |

1.0 INTRODUCTION

1.1 ABSTRACT

Now a days, it is highly possible to adapt mobile computing in various applications. The portability, open source nature of smart phones and android development platform has made the development of application software for various environments as handy. Smartphone applications are resulted in paper less work, easy to use and time saving in nature. The wireless communication technology of smart phone enables the information transfer from the current client to remote database server, where ever the network range is available.

The Webkiosk android application is a mobile computing software application, which provides easy access to the attendance record of a student in every registered subject. In this work, some additional functionalities are given to the user which will allow the user to manage the attendance more effectively. The additional functionalities will be display of exam grades of candidates , an attendance predictor which predicts the attendance after next class is missed or attended.

1.2 OBJECTIVE

The objective of this project is to build an online system app for students of college/ Universities to view their attendance, time table on their mobile smart phones directly rather than using any web browser application.

There already is an online browser based system named "Webkiosk" for viewing the attendance record of a student, but in this project will build an application for doing so. Which will be a faster and easier approach to this action.

In addition to the basic attendance display functionality of the application, there will be some additional functionalities to improve the overall experience of the user and provide more options to make attendance management easier than before.

1.3 METHODOLOGIES

In Webkiosk Android Application software used for creating the user interface of the application is android studio with SDK version. A software development methodology is a technique that is used to structure, plan and control the process of developing a new system. There are several software development methodologies available now-a- days . Each one have its own strengths and weaknesses. The selection of the most efficient and appropriate methodology is required to develop a new software based on the nature of new system. Each iteration is considered as a stand-alone project composed of activities. So, the phases of life cycle such as requirements analysis, design and implementations are carried out at each iteration in turn. The results are accommodated into the overall system which grows incrementally within the system until the iteration is fully completed. Although the waterfall model is widely used for software development, yet it is not considered as the ideal way. The waterfall model has some risks at each stage, despite the fact that it is easy to define at each stage. While moving to the next stage, the previous stage should be fully implemented. However, at any point during the life cycle of software, we might have to go back to make readjustments and refinements. So it is not a safe approach. The extreme programming cannot be used as it requires working in pairs while developing. Conversely, in the iterative and incremental approach, a new system is develop using smaller incremental chunks. In this, each chunk is provided with some activities and a specific deadline is given, within which it has to be completed. The chunk is combined to the system, once it is fully completed. This cycle is to be executed until the system is fully developed .In case of any failure during any iteration, the process will not be carried out until we get a fully functional outcome. Using the iterative and incremental approach, the minimum requirements will be provided to the users. Based on the advantages described above the iterative and incremental approach will be followed here, that is the app to be developed will be divide into smaller chunks. Once a chunk is fully functional, a paradigm of the app has to be distributed among the students for evaluation. Finally, if there are no errors, the new chunk will be added to the existing code. The process will be carried out until application is fully completed. Priority will be given to the requirements, in order to secure a functional delivery before the deadline. More focus will be on implementation and delivery of a functional chunk of minimum

requirements at the first iteration of application development. Other iterations will focus on capabilities of application.

2.0 LITERATURE SURVEY

-Reetesh V. Golhar, Prasann A. Vyawahare, PavanH.Borehare, AshwiniManusmare [1] developed an android application for an Institute which connected all the departments of the Institute like administrator, accounts, Students section and many more modules. In this application they got all the notifications which were given by the institute on the notice board to be displayed on this android application. This saved a lot of time and the long time consuming process got simpler by this application.

-Abhinav kathuria, Anu Gupta in the department of Chandigarh of Punjab University gave a research paper named " [2] Challenges in Android Application Development: A case study" which discussed the design and development of an Android based app named as ' iquiz' that can help students in the preparation of competitive exams like UGC-NET, GATE etc while they are on the move.

-Li Ma, Lei Gu and Jin Wang [3] gave a research paper on development of android applications with minimum redundancies and pop ads. They developed 3 kinds of applications based on Android SDK and Java i.e. Audio Player, Weibo Client and video player .Their main focus was to develop android applications with minimum interference which results in smooth functioning.

2.1 ANDROID

Android is a portable working framework created by Google, in view of a changed form of the Linux bit and other open source programming and composed fundamentally for touchscreen cell phones, for example, cell phones and tablets. Furthermore, Google is also working very hard to implement many more features in to the already present gadgets.. Variations of Android are additionally utilized on diversion supports, computerized cameras, PCs and different hardware.

At first created by Android Inc., which Google purchased in 2005, Android was revealed in 2007, with the primary business Android gadget propelled in September 2008. The working framework has since experienced various real discharges, with the present rendition being 8.1 "Oreo", discharged in December 2017. The centre Android source code is known as Android Open Source Project (AOSP), and is basically authorized under the Apache License.

Android is likewise connected with a suite of restrictive programming created by Google, including centre applications for administrations, for example, Gmail and Google Search, and also the application store and computerized dissemination stage Google Play, and related advancement stage. These applications are authorized by producers of Android gadgets ensured under guidelines forced by Google, yet AOSP has been utilized as the premise of contending Android biological communities, for example, Amazon.com's Fire OS, which use its own counterparts to these Google Mobile Services.

Android has been the smash hit OS worldwide on cell phones since 2011 and on tablets since 2013. As of May 2017, it has more than two billion month to month dynamic clients, the biggest introduced base of any working framework, and starting at 2017, the Google Play store includes more than 3.5 million apps.

2.2 ANDROID ARCHITECTURE

2.2.1 ANDROID FUNDAMENTALS

While android application are being written in standard Java programming language, they are not executed in the same way as the standard java codes that is via Java virtual machine (JVM). In place of that Google has offered a custom virtual machine which is responsible for the conversion and execution of the code. This virtual machine created by Google is known as

Dalvik. Here all the custom Java classes are to be converted into a Dalvik compatible instruction set before they are executed in the Operating system. In Dalvik Virtual machine the generated Java class files are taken , then they are merged into one or many Dalvik compatible format (.dex). One unique advantage of using Dalvik is that it reuses the duplicate information from multiple class files and in doing that it reduces the file size by half of the traditional .jar file. This advantage of Dalvik comes in handy considering the size of the android devices which is not very large, therefore the small size of these executables is preferred. This is more convenient for all kinds of devices as small size is preferred everywhere. Dalvik executable files are smaller in size and are faster to execute than traditional java executables (.jar) , therefore this method gives better performance while using less space on the device.

2.2.2 ANDROID PLATFORM

Android is a product stack for mobiles that integrates a working framework, middleware and key applications. Android applications are built by the tools and APIs provided by the Android SDK using Java programming language. Linux controls the Android framework administrations, security, and process administration.



fig 2.1 Android Architecture

2.2.3 APPLICATION FRAMEWORK

Developers can create applications, as it gives an open improvement stage. Anything from getting to area data to exploiting gadget equipment is given to the developer by Android.

Developer has full access to the API systems used by the core applications. Easiness in reusing the segments is kept in thought while outlining the application design, any application can distribute its capacities and some other application may then make utilization of those abilities (subject to security limitations implemented by the structure). Client can supplant the parts by a similar system. Hidden all applications is an arrangement of administrations and systems, including:

-Lists, grids, text boxes etc, can be used to build an application.

-Data sharing among applications can be done by Content Providers.

- Layout files, graphics and localized strings are managed by the Resource Manager.

-A Notification Manager that enables all applications to display custom alerts in the status bar

- A common navigation back stack is provided by the Activity Manager which takes care of the application lifecycle.



fig 2.2 Android framework

2.2.4 ANDROID RUNTIME

Android Runtime (ART) is an application runtime condition utilized by the Android working framework. Substituting Dalvik, the process virtual machine initially utilized by Android, ART plays out the interpretation of the application's bytecode into local directions that are later executed by the gadget's runtime condition.

Android 2.2 "Froyo" brought trace-based just-in-time (JIT) assemblage into Dalvik, enhancing the execution of utilizations by constantly profiling applications each time they run and progressively accumulating much of the time executed short sections of their bytecode into local machine code. While Dalvik deciphers whatever remains of application bytecode, local execution of those short bytecode sections, called "traces", gives noteworthy execution changes.

Dissimilar to Dalvik, ART presents the utilization of ahead-of-time (AOT) arrangement by assembling whole applications into native machine code upon their establishment. By wiping out Dalvik's interpretation and trace based JIT aggregation, ART enhances the general execution productivity and lessens power utilization, which brings about enhanced battery self-rule on cell phones. In the meantime, ART brings speedier execution of utilizations, enhanced memory allocation and trash gathering systems, new applications investigating highlights, and more precise abnormal state profiling of uses.

To keep up in backward compatibility, ART utilizes a similar info bytecode as Dalvik, provided through standard .dex records as a component of APK documents, while the .odex documents are supplanted with Executable and Linkable Format (ELF) executable.

2.2.5CURRENT VERSION

Today android is in its 8th version, Android 8.1 and the codenamedis Oreo. The lack of APIs, enhanced UI and added functionalities are corrected by every instalment of Android. This latest version of android adds some important things such elements as multitasking, switching between the running apps, also the ability to use 2 or more languages at the same time. High-performance 3-D graphics are managed by the revolutionised Vulkan API. Also using this new version of android on supported devices we will see apps grow to life because of sharper graphics and eye catching effects.

Why Android is better?

• Applications

- Google applications

Google applications like Gmail, YouTube or Maps are generally pre-installed on Android.

-Widgets

Small tools on the main window called widgets can be used to get information about the particular application.

-Android Market

Applications built by the developers are available on the play store, which is an online store either for a price or free, which can be downloaded by the user directly on their smartphone.

Multitasking

Android allows multitasking in the sense that multiple applications can run simultaneously. With Task Manager it is possible view all running tasks and to switch from one to another easily.

• SDK

A development kit has been put at disposal of everybody. Accordingly, any developer can

create their own applications, or change the android platform. This kit contains a set of libraries, powerful tools for debugging and development, a phone emulator, thorough documentation, FAQs and tutorials.

2.3 JSON

JSON stands for JavaScript Object Notation and it is used as a data exchange format. JSON can be used in various projects which are in need of objects, arrays or parsing. In this application the parsing of data from the Webkiosk is done using JSON parsing.

The following entities has been used in this project-

(a) Object

An object is collection or set of name/value pairs, an object always begins with '{' and always ends with '}'. Each name is followed by a colon ':', these pairs are separated with ','.



(b) Array

An array is a set or collection of values, an array always begin with '[' and always ends with a ']'. An array can also hold JSON objects.



(c) Value

A value can be many things, it can be a string in double quotes or a Boolean value or a number, an array, an object.



fig 2.5 JSON Value

(d) String



fig 2.6 JSON String

(e) Number



fig 2.7 JSON Number

2.4 PHP

PHP means hypertext pre-processor. It is a server-side scripting language. The static websites or dynamic websites or web applications are mainly developed using language. PHP means hypertext preprocessor, which previously meant personal initial pages.

PHP scripts can only be interpreted on a server on which PHP is installed.

Client that access PHP scripts require only a web browser.

A PHP file consists of PHP tags and always ends with the extension".php".

What is a scripting language?

A script generally is a set or collection of programming instructions which are interpreted at runtime of the script.

So we can say that a scripting language is essentially a language that interprets scripts at runtime. These scripts are often integrated into other software environments.

The purpose of scripts is generally to improve performance or perform routine tasks for an application.

In scripting languages the scripts which are on the server side are interpreted on the server while the scripts which are on the client side are interpreted by the client application.

There is difference between PHP and javascript, javascript is client side script interpreted by the client's browser while PHP is a server-side script interpreted on the server. The similarity between the two is that Both PHP and JavaScript can be integrated into HTML pages.

Programming Language Vs Scripting Language

| Programming language | Scripting language |
|---|--|
| Contains all the functions for building complete applications | Utilized in routine tasks |
| Before execution code has to be compiled | Code generally executed without compiling beforehand |
| No need to embed these in other languages | Are generally embedded into other languages |

What does PHP stand for?

PHP used to mean **Personal Home Page**, but today it refers to Hypertext Preprocessor.

PHP code is used in many areas of web development, it can be used in web content management systems. One of the most popular use of PHP is with the HTML code, when it is used with an HTML code it is most useful. PHP can also be used with different web template systems.

Php Syntax

```
<?php
echo 'Hello World';
?>
```

A PHP file, that is the file with .php extension can also contain tags such as HTML and client side scripts such as JavaScript.

- HTML is an added advantage when learning PHP Language. PHP can be learned without knowing HTML but you can work with PHP at best when you at least are familier with the basics of the HTML language.
- **Database management systems** or DBMS is used in applications which are database powered.
- JavaScript and XML are mostly being used in more advanced topics some of which are interactive applications and web services, you will need.

Below is a Flowchart diagram which shows the basic architecture of a PHP application and how the requests are handled by the server.



fig 2.8 PHP flowchart

Why use PHP?

Obviously, you are in charge of a certain number of programming languages; One might wonder why we would like to use PHP as our web programming poison. Below are some of the valid reasons to use PHP

- PHP unlike many other languages is an open source and is free.
- Unlike many other languages PHP has a short learning curve.

• Most of the today's web hosting servers are supporting PHP, while other languages such as ASP that require IIS. This is a huge advantage which makes PHP a profitable choice.

•Today PHP is updated periodically to keep up with the latest user or client needs and technology trends.

• PHP being a server side scripting language adds one more advantage to already big list what this means is that it needs only to be installed on the server side and the clients which require server resources need not have PHP installed, only use of a simple web browser would be enough tp keep things running.

• PHP also goes well with database languages like MySQL. Thats not all you can use PHP with every dabase management systems. You can still use PHP with

or Postgres

or Oracle

or MS SQL Server

or ODBC, etc.

• PHP is multiplatform; This means that you can implement your application on different operating systems, such as Windows, Linux, Mac OS, etc.

When we talk about market share, PHP is not behind any other scripting language here there are more than 20 million websites and applications developed on the Internet using the PHP scripting language.

This can be attributed to the points mentioned above;

The diagram below shows some of the popular sites that use PHP



fig 2.9 PHP Website

2.5 SQLite Database

SQLite is a very popular database management system. It is being used in many of the big firms and is actually preferred by many experts. There are many unique features which SQLite carries.

a) Zero configuration-

There is no need of installation or configuration if anyone wants to use SQLite

b) Serverless-

Unlike most system which needs to be executed as separate servers, SQLite directly enables any process or thread to write or read from the database.

- c) Single database file
- d) Compact

2.6 Android studio

Android Studio has been developed for developing android applications by Google official and it is officially integrated development environment (IDE) for Google's Android operating system, developed with JetBrains Intelli IDEA J software and this software is only build for creating applications for android. This software can be downloaded from many known websites and is available for many platforms such as Linux, MacOS, Windows. Since its release, android studio has been the substitute for app development in place of eclipse. This development tool is becoming very popular in android market and is preferred by many developers.

At a Google conference held on April 15, 2013, Android Studio was launched. In the preview given at the conference version 1.0 was displayed this was after the beta version was developed and fully tested by Google team, this happened in June 2014. After the successful results shown by the beta version the stable version was released and can with a storm in the market. The current version which is stable and has alot of new features is 3.2.0 which was launched in 23 October 2017.

2.7 Use of API (Application Programming Interface)

API stands for Application Programming Interface. This API acts as an intermediate connection between two communicating applications, the use of this tool enables two applications two communicate with each other. We are using API's in our day today life without even realising it, whenever we use our day today apps like Facebook or any other app to send a message we use API to communicate with the server.

Lets see an example of API

Lets think about is, we all use different apps to connect to the internet and send or receive data. After making the connection with the server, it gets the necessary information then it decodes what is written and then forwards the data to the phone. The application then interprets this data and gives the desired information in a readable way. This is what an API does, this is the main function of API, all of this is possible through the API.

To understand this concept better, we are going to take an example which will be easier to understand. Suppose you are a football player who is need of a transfer to a certain club, here the club you want to go to is the 'system' and will give you the offer or the contract. Here only one thing is missing that your link with the club, that is where the agent comes in to play here the agent is the API. The agent tells the club what you need and then comes back with the answer. The agent is acting as an API here.

Lets take another example to understand it better. Suppose you want to go see a movie, what do you do? You open up the website select things like cinema hall, time, movie, seat etc. To make a reservation for the movie you interact with the website and run through its database and check whether any seats are available and also check the cost of each seat.

Now let us think what will happen if we use any other website to book our ticket like bookmyshow which adds up the data from different sites instead of the direct site of the cinema hall. What will happen if we use any other tools which doesn't have direct access to the databases?

In this scenario of using other site like bookmyshow, it communicates with the cinema hall an here it is acting like an API. So API is the service which will request the booking system to get information from the cinema hall's databases. The API then gives back the response for the

cinema's request, after that it delivers the data to bookmyshow. These results display the most recent updates for the cinema halls databases.

What an API Also Provides Is a Layer of Security

The data on your phone is never completely exposed to the server and in the same way the server is never completely exposed to your phone. Instead, everyone communicates with small data packets, sharing only what is needed, like ordering take-out food. You tell the restaurant what you'd like to eat, they tell you what they need in return, and then, in the end, get your food.

The APIs have become so valuable that they make up much of the revenue of many companies. Huge MNC companies like Salesforce, Google, eBay, Amazon and Expedia are just some of the companies that make profits with their APIs. What "API economy" refers to this API market.

The Modern API

Over the years, what an "API" has often described any type of generic connectivity interface for an application. More recently, however, the modern API has acquired some features that make it extraordinarily valuable and useful:

• Modern APIs have many advantages like they conform to standards (typically HTTP and REST), which are user-friendly moreover thee are also easily accessible and widely understood.

• They are treated more like products than codes. They are designed for consumption of specific audiences (for example, mobile device developers), are documented and are submitted to a version so that users can have certain maintenance and lifecycle expectations.

• The modern API's are much more standardized and because of this they have a much stronger discipline for security and governance, as well as being monitored and managed for performance and size.

• Like any other production software, the modern API has its own software development cycle (SDLC) for design, testing, construction, administration and version control. Furthermore, modern APIs are well documented for consumption and version control.

2.8 Amazon API Gateway

Amazon API Gateway is a fully managed gateway service which is provided by amazon to allow the developers to create, publish, maintain, monitor and protect APIs at any level which is required. By using Amazon API and few clicks in the AWS Management Console any one can create an API that acts as a "gateway" for applications to access data, business logic, or the functionality of their back-end services, such as workloads that are running on Amazon Elastic Compute Cloud. (Amazon EC2), code that runs on AWS Lambda or any web application.

Amazon API Gateway does the management of all the activities which are involved in the acceptance and processing of hundreds of thousands of simultaneous API calls, including traffic management, authorization and access control, and monitoring and administration of the API version. One more advantage of Amazon API Gateway is that it does not have anyminimum fees or start-up costs. You will only pay for the API calls that you receive and the amount of data which is transferred.





3.0 SYSTEM DEVELOPMENT

3.1ANALYSIS

3.1.1 EXISTING SYSTEM

There is already an android application available on Google play store for displaying the attendance of any student for any subject. It provides only basic features like after logging in to your account from the home page, you can see your current semester attendance record for every registered subject.

3.1.2 ADVANTAGES OF OUR APPLICATION OVER EXISTING ONE

The new webkiosk application will provide more features in addition to the basic Log In and attendance display which are the following-

- Log in
- Attendance display of every subject
- Attendance predictor (allows user to see what will be the attendance if next class is missed or attended)
- Display exam results.

3.1.3 REQUIREMENTS

Operating system-

- Windows 7 or above
- Java and XML Development used
- Android Studio v 2.3.0, API level 26
- Database: SQL + database
- Client/Server system
- Platform: Java/PHP

Hardware/ Software requirements-

The workstation android devices should at least have following configurations

- 800 MHz processor
- 128 MB RAM
- 256 MB HDD

• Android 2.2.0 and above

Design and Implementation constraint-

- The time allocated for this project will be till the end of this semester.
- The language will be java and SQL, development environment will be Android Studio 2.2.3.

User documentation-

- The system app will provide an online attendance view system on smart phones that describes the functionality and options available to users.
- The system will provide a hard copy of user manual which is identical to that is viewed in HTML.

Assumptions and Dependencies-

- The database mentioned in this SRS document is precisely administered with correct information needed by webkiosk.
- The system assumes that detail regarding each student would be correct.
- Class data being used for setup and student recognition is dependent on informatics.
- Statistics on student attendance is dependent on faculty's consistent update of attendance.

External Interface requirements-

User Interface-

- The system shall provide details of attendance and time table of respective students.
- All modifications will be taken through updated database.
- Application will be accessed through a mobile user interface.
- The program will provide a page that produces current stats on class attendance and time table.

Software Interfaces-

• To allow a user to view attendance

- To allow a user to view time table
- To allow a user to calculate the lectures needed to be attended to maintain particular attendance as prescribed by authorities.

Communication Interfaces-

- The system will communicate to database through internet.
- Security of a user must be consistent through use of password.

3.2 DESIGN:



fig 3.1 Design Flowchart

FLOW-DIAGRAM



fig 3.1.1 Flow Diagram

This is a flowchart of "Webkiosk Android application", here upon entering the student credentials the user is logged into the home page of the application. After that the functionalities of the app are ready to be used.

This flowchart correctly shows hoe the decision is taken and upon doing what, where the app will take you.

USE-CASE DIAGRAM:

WEBKIOSK ANDROID APPLICATION



fig 3.1.2 Use Case Diagram

This use case diagram tells us about the functionalities that can be accessed by the student.

The actor can access the following activities from the system-

- 1. Login activity
- 2. View Attendance activity

- 3. View Date sheet activity
- 4. View Subjects registered activity
- 5. Attendance Calculator activity
- 6. Logout activity

ER MODEL:



fig 3.1.3 ER Model

From this diagram we can see that enrolment number is used a a primary key when identifying a user. This is the ER model of Webkiosk being used in JUIT Waknaghat.

-courseid is one of the candidate keys

Activity Diagram:





Working of the system can be explained from the above diagram, as the user enters their credentials to login. The credentials are authenticated at the JUIT web servers.

If the credentials are authenticated, Login is granted, and if wrong the user is asked to re-enter their credentials.

After logging in the user enters the home screen with an overview of their attendance.

From there the user can chose from multiple options which are

Get detailed attendance, get their time table, get date sheet, use the attendance calculator, view registered subjects and finally logout.

4.0 PERFORMANCE ANALYSIS

Following screenshots gives the performance of the Android application when different options were choosen :

| Webk | losk | 2 - [~/StudioProje | cts/Webkiosk2] - [app] | - ~/StudioPr | ojects/Webkiosk2/app | o/src/main/java/com/exam | ple/akhil/webkiosk/D | etailedAtten | dance.) 🤶 | En 🕴 👞 ୶)) | 2:14 PM | ₩ |
|------------|-------------|---|--|---------------------|--|---------------------------|----------------------|--------------|-------------|--|-----------|-------------|
| 6 | Eil | e <u>E</u> dit <u>V</u> iew <u>N</u> av | igate <u>C</u> ode Analy <u>z</u> e <u>F</u> | Refactor <u>B</u> u | ld R <u>u</u> n <u>T</u> ools VC <u>S</u> <u>V</u> | ⊻indow <u>H</u> elp | | | | | | |
| | | 🖶 💋 🛩 🄶 💥 | 5 D D Q Q 4 → | 🔨 🗔 app | - 📐 🔸 🕸 🕩 🖬 🔳 | I 🖫 🧕 🖬 🚣 ? | | | | | Q, | . 🖪 |
| | D | Webkiosk2 े 🛅 ap | p 🔁 src 🔁 main 👌 🗖 j | ava 📄 com | 🗈 example े 🛅 akhil 🔪 | 🗈 webkiosk 🛇 🖸 DetailedA | ttendance | | | | | |
| | t | Android Monitor | | | | | | | | | \$r• ±. | |
| R | Proje | 📗 LENOVO Le | novo A7020a48 Android | 6.0, API 23 | com.example.akhil. | webkiosk (10571) 🔻 | | | | | | Grad |
| P - | #1 © | Monitors | | | | Verbose 🔻 🔍 | | 🛛 🗹 Regex | Show only s | selected applicatio | n 🔻 🎫 | e |
| | g | Memory | | | | | | | | | | |
| | 📢 Z: Struct | 50.39 MB | | | | | | | | Free [3.63 MB] Allocated [18.11 M | 18] | |
| | s | 0.00 MB | | | | | Os | 5s | 10s | - | | |
| 2 | pture | M CPU | | | | | | | | | | |
| | 0 () | 100.00 % 80.00 % | | | | | | | | User [0.00 %] | | |
| | | 40.00 % | | | | | | | | Kernel [1.59 %] | | |
| 9 | | 0.00 % | | | | | 0s | 5s | 10s | - | | |
| | | ڬ Network | | | | | | | | | | |
| | ۲ | 19.46 KB/s | | | | | | | 1 | _ | | |
| | arian | 8.00 KB/s | | | | | | | | Tx [0.00 KB/s] Bx [0.00 KB/s] | | |
| 1 | Build V | 0.00 KB/s 3.86 KB/s | | | | | 0s | 5s | 10s | _ | | |
| | ÷ | M GPU | | | | | | | | | | |
| | : Favorites | | | | | This monitor is disabled. | | | | | | Gradle Cons |
| | * | | | | | | | | | | | ole |
| | | 📐 <u>4</u> : Run 🛛 🔮 TODO | <u>6</u> : Android Monitor | Terminal | 📕 <u>0</u> : Messages | | | | Q | Event Log 🛛 🏺 Andr | oid Model | |
| | | Gradle build finish | ed in 1s 268ms (moment | s ago) | | | | 363:76 L | F‡ UTF-8‡ C | ontext: <no contex<="" th=""><th>(t> 1a</th><th>0</th></no> | (t> 1a | 0 |

Figure 4.1: Screenshot 1 of performance analysis (Inputing information for login)

| Webk | iosk File | 2 - [~/StudioProjec Edit View Navio | ts/Webkiosk2] - [app] gate Code Analyze | -~/StudioPr Refactor Bui | ojects/Webkioska ld Run Tools V | 2/app/src/main/java CS Window Help | a/com/examp | le/akhil/webk | ciosk/Det | ailedAtten | dance.j 蒂 🖪 | n 🕴 📧 🕪) 🕻 | 2:15 PM · | ψ |
|----------|----------------|---|--|-----------------------------|------------------------------------|---------------------------------------|----------------|---------------|-----------|------------|----------------|---|-----------|----------------|
| 0 | | H 🖸 🛩 \Rightarrow 🐰 | D D Q Q 4 + | 🔸 🔼 app | - + 🎄 🖡 [| i 🔳 🖳 🧕 🖬 | ₩ ? | | | | | | Q, | |
| | - | Webkiosk2 👌 📬 app |) 🖿 src) 🖿 main) 🖿 j | java 🖻 🖿 com | 🗈 example 🔪 🗈 al | khil) 🗈 webkiosk) (| C Detailed Att | endance | | | | | | |
| | sct | Android Monitor | | | | | | | | | | | ⊕- ±. | ۲ |
| <u>R</u> | Proj | 🔲 LENOVO Ler | ovo A7020a48 Android | 6.0, API 23 | com.example.a | akhil.webkiosk (115 | 46) 🔻 | | | | | | | Grad |
| • P | • | Monitors | | | | Verbose 🔻 | Q.* | | | 🗹 Regex | Show only se | ected application | - 112 | ſe |
| | are | Memory | | | | | | | | | | | | |
| | 📢 Z: Struct | 50.39 MB | | | | | | | | | | Free [0.53 MB] Allocated [21.20 MB | 1] | |
| | se | 0.00 MB | | | | | 0s | 5s | 10s | 15s | 20s | | | |
| 2 | aptui | M CPU | | | | | | | | | | | | |
| | 3 | 100.00 % 80.00 % 40.00 % | | | | | 0s | 5s | 10s | 155 | 20s | User [1.33 %] Kernel [0.00 %] | | |
| 1 | | Metwork | | | | | | | | | | | | |
| | Build Variants | 19.46 KB/s 8.00 KB/s 0.00 KB/s 3.86 KB/s | | | | | 0s | | 10s 🔻 | 155 | 205 | Tx [0.00 KB/s] Rx [0.00 KB/s] | | |
| | • | Magen GPU | | | | | | | | | | | | |
| a | 🗰 2: Favorites | 0.00 ms | | | | This monitor is | s disabled. | | | | | | | Gradle Console |
| | _ | 📐 4: Run 🛛 🔮 TODO | 🏺 <u>6</u> : Android Monitor | Terminal | 👜 <u>0</u> : Messages | | | | | | Q Eve | ent Log 🛛 🏺 Andro | id Model | 0 |
| 100 | | Gradle build finishe | d in 1s 246ms (moment | ts ago) | | | | | | 363:76 LI | F‡ UTF-8 ‡ Cor | itext: <no contex<="" th=""><th>> °a</th><th>曼</th></no> | > °a | 曼 |

Figure 4.2: Screenshot 2 of performance analysis (Attendance view)



Figure 4.3: Screenshot 3 of performance analysis (Detail Attendance View)



Figure 4.4: Screenshot 4 of performance analysis (Attendance Predictor)



Figure 4.5: Screenshot 5 of performance analysis (Student Info)



Figure 4.6: Screenshot 6 of performance analysis (Time Table)



Figure 4.7: Screenshot 7 of performance analysis (Time Table Fragment Switching)



Figure 4.8: Screenshot 8 of performance analysis (Switching To DateSheet Activity)



Figure 4.9: Screenshot 9 of performance analysis (Selecting Year In Datesheet)



Figure 4.10: Screenshot 10 of performance analysis (View Datesheet)



Figure 4.11: Screenshot 11 of performance analysis (Subject Registered Activity)



Figure 4.12: Screenshot 12 of performance analysis (Logout)

LOGIN:



Fig4.13(a)



OVERIEW ATENDANCE:

| | ATTENDANCE | |
|------------|-------------|-----------|
| | | |
| ADVANCED A | LGORITHMS | |
| 10M11Cl211 | | |
| Lect: 75 | Tut: 75 | Total: 75 |
| | CODITUMS | |
| ADVANCED F | LGORITHMS | |
| | 7.4.75 | T-4-1-75 |
| DESIGN AND | ANALYSIS OF | REAL |
| | DRITHMS | |
| 17B1WCI814 | | |
| Laste 02 | Tut: 83 | Total: 83 |







DETAILED ATTENDANCE:



Fig 4.16

ATTENDANCE PREDICTOR:



Fig 4.17

TIME TABLE:



Fig 4.18

STUDENT INFO:



Fig 4.19

DATESHEET:



SUBJECT REGISTERED:



STRATEGIC MANAGEMENT

11B1WPD832

Fig 4.21

LOGOUT:

| = | Home | Settings | |
|------------------------|-----------|----------------------|-----------|
| | AT | T Logout | |
| ADVAN | ICED ALG | ORITHMS | |
| 10M110 | CI211 | | |
| Lect: 75 | Т | Tut: 75 | Total: 75 |
| ADVAN 10M110 | ICED ALGO | ORITHMS | |
| Lect: 75 | Т | Tut: 75 | Total: 75 |
| DESIGI | N AND AN | ALYSIS OF R | EAL |
| WORL |) ALGORIT | THMS | |
| 17B1W | CI814 | | |
| Lect: 83 | Т | Tut: <mark>83</mark> | Total: 83 |

MOBILE AND DISTRIBUTED

Fig 4.22

5.0 CONCLUSION

The android application was created successfully. All the features which were proposed at the start of the project were added to the application.

This application is now ready to be used by the students of JUIT Waknaghat and they don't need to use the time consuming browser for accessing the functionalities of Webkiosk.

This app created is easier and faster to use than the accessing the functionalities of the webkiosk through browser.

5.1 Future Scope

Some additional functionalities which were left out in this version of application can be added in the future. Also there is scope of optimizing the performance with latest technologies which will be available in the coming versions of android studio and JSON.

6.0 REFERENCES

[1] Abhinav Kathuria and Anu Gupta " iquiz" application by, Department of Computer Science, Punjab university.

[2] Reetesh V. Golhar, Prasann A. Vyawahare, PavanH.Borehare, Ashwini Manusmare on "application which connects all the departments of an Institutes."

[3] Li Ma, Lei Gu and Jin Wang "Research and Development of mobile applications"