**INDUSTRIAL TRAINING REPORT**

**DOCKET: THE TIMETABLE APP**

Submitted in partial fulfillment of the

Requirements for the award of

**Degree of Bachelor of Technology in Computer Science Engineering**



Submitted by

Name: YASH JAISWAL

University Roll No. 1901430100243

**SUBMITTED TO:**

**Department of Computer Science & Engineering (16 size)**

IMS Engineering college

GHAZIABAD (UP).

**CERTIFICATE**



**DECLARATION**

I hereby declare that the Industrial Training Report entitled ("DOCKET: THE ORGANISER") is an authentic record of my own work as requirements of Industrial Training during the period from August 5 to September 5 for the award of degree of B.Tech. (Computer Science & Engineering), IMS ENGINEERING COLLEGE, GZB, under the guidance of (Tim Buchalka & Jean-Paul Roberts).

**(Signature of student)**

**(YASH JAISWAL)**

**(1901430100243)**

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Head of Department**

**(Signature and Seal)**

**ACKNOWLEDGEMENT**

To list who all have helped me is difficult because they are so numerous and the depth is so enormous.

I would like to acknowledge the following as being idealistic channels and fresh dimensions in the completion of this project.

I take this opportunity to thank IMS ENGINEERING COLLEGE for giving me chance to do this project.

I would like to thank my HOD, Dr. Pankaj Agarwal for providing the necessary facilities required for completion of this project.

I take this opportunity to thank our Coordinator Ms.C Pushpanjali for her moral support and guidance.

I would also like to express my sincere gratitude towards my project guide Ms. Megha Gupta whose guidance and care made the project successful.

**ABOUT THE COMPANY**

Udemy, Inc. is an American massive open online course (MOOC) provider aimed at professional adults and students. It was founded in May 2010 by Eren Bali, Gagan Biyani, and Oktay Caglar.

As of January 2020, the platform has more than 35 million students and 57,000 instructors teaching courses in over 65 languages. There have been over 400 million course enrollments. Students and instructors come from 180+ countries and 2/3 of the students are located outside of the U.S.

Students take courses largely as a means of improving job-related skills. Some courses generate credit toward technical certification. Udemy has made a special effort to attract corporate trainers seeking to create coursework for employees of their company. As of 2020, there are more than 130,000 courses on the website.

The headquarters of Udemy is located in San Francisco, California, with offices in Denver, Colorado; Dublin, Ireland; Ankara, Turkey; Sao Paulo, Brazil; and Gurugram, India.

In 2007, Udemy (you-de-mee, portmanteau of you + academy) [9] founder Eren Bali built a software for a live virtual classroom while living in Turkey. He saw potential in making the product free for everyone, and moved to Silicon Valley to found a company two years later. The site was launched by Bali, Oktay Caglar and Gagan Biyani in early 2010.

In February 2010, the founders tried to raise venture capital funding, but the idea failed to impress investors and they were rejected 30 times, according to Gagan Biyani. In response to this, they bootstrapped the development of the product and launched Udemy—"The Academy of You"—in May 2010.

Within a few months, 1,000 instructors had created about 2,000 courses, and Udemy had nearly 10,000 registered users. Based on this favorable market reaction, they decided to attempt another round of financing, and raised $1 million in venture funding by August.

In October 2011, the company raised an additional $3 million in Series A funding led by Groupon investors Eric Lefkofsky and Brad Keywell, as well as 500 Startups and MHS Capital.

In December 2012, the company raised $12 million in Series B funding led by Insight Venture Partners, as well as Light bank Capital, MHS Capital and Learn Capital, bringing Udemy's total funding to $16 million.

On April 22, 2014, the Wall Street Journal's digital edition reported that Dennis Yang, Chief Operating Officer of Udemy was named CEO, replacing Eren Bali.

In May 2014, Udemy raised another $32 million in a Series C funding, led by Norwest Venture Partners, as well as Insight Venture Partners and MHS Capital.

In June 2015, Udemy raised a $65 million Series D financing round, led by Stripes Group. Now Udemy joined another online learning house Skillsdox Inc. of Canada to open up School of Skills in India.

**TABLE OF CONTENTS**

1. **INTRODUCTION**

a.Introductionof Project **9**

b. Project Overview  **10**

c. Purpose  **10**

d. Limitation of Project  **10**

e. Objective of Project **10**

1. **SYSTEM ANALYSIS**

a. Feasible study **11**

b. Introduction **11**

c. Summary and Constraint **11**

d. Types of Feasible Study **12**

e. Hardware and Software Requirements **13**

**3. DEVELPOMENT ENVIRONMENT**

a. Introduction to SQL Database **14**

b. Introduction to Android Studio **16**

c. Introduction to Java  **18**

**4.SYSTEM DESIGN**

a. Modular Description **22**

**5.IMPLEMENTATION AND TESTING**

a. Testing  **23**

b. Implementation and Screenshots **26**

**6. CONCLUSION 29**

**7. BIBLIOGRAPHY 30**

**LIST OF TABLE**

1. Basic system requirements for Android Studio pg - 17

**LIST OF FIGURES / IMAGES**

1. Landing page pg – 26

2. Setting pg - 26

3. Adding the timetable pg - 27

4. Adding the task pg - 27

**ABSTRACT**

This project “**DOCKET**” aims to provide a simple platform to the students to manage their timetable. This software will help students in saving their time and enhancing their capabilities. The app will help the user to work in a highly effective and efficient environment. Students only an android device to use this app. This software also stores the details of the user in the database which

can be accessed further in future. This app enables user to construct a timetable without any collision with other timetable in a simple and creative way.

This provide the students with the scope of improvement not only in their carrier but in their life as well. The project “**DOCKET**” is developed with the objective of making the system reliable, easier, fast, and more productive.

**INTRODUCTION**

**A. INTRODUCTION OF PROJECT**

Timetabling concerns all activities with regard to producing a schedule that must be subjective to different constraints. Timetable can be defined as the optimization of given activities, actions or events to a set of omits in space-time matrix to satisfy a set of desirable constraints.

A key factor in running an educational center or basically an academic environment is the need for a well-planned, well-throughout and clash-free timetable. Back in the days when technology was not in wide use, (lecture) timetables were manually created by the academic institution.

Every school year, tertiary institutions are faced with the tedious task of drawing up academic timetables that satisfies the various courses and the respective. amination being offered by the different department.

Timetable development process starts when each Head of Department provide the following information to be used for timetable scheduling. The information provides the modules with dates, time and venues suitable in a particular semester:

• Examinable courses in a particular semester.

• Dates for lectures to be held

• Specified time for lectures i.e. Between 8am and 4pm)

• The venue of the scheduled lectures.

A timetabling problem consists of four parameters and they are:

• T (set of time),

• R (set of available resources),

• M (set of scheduled contacts) and

• C (set of constraints).

This problem assigns time and resources to the contacts on such a way that the constraints will be satisfied. In various timetabling problems, educational timetabling has been generally examined from practical standpoint. The quality of the timetable determines the quality of time dedicated by lecturers, students and administrators to academic activities. Various academic timetabling includes:

• School timetable

• Lecture timetable

• Examination timetable and

• Assignment timetable.

**B. PROJECT OVERVIEW**

Timetabling concerns all activities with regard to producing a schedule that must be subjective to different constraints. Docket helps in scheduling and organizing all the activities for all the 7 days of the week.

**C. PURPOSE**

Planning timetables is one of the most complex and error-prone applications. The available system currently builds or generates a set of timetables manually, but most times have issues with generating a clash-free and complete timetable. Most educational institutions have resorted to manual generation of their timetables which according to statistics takes much time to get completed and optimal. Even at the optimal stage of the manually generated timetable, there are still a few clashes and it is the lecturer that takes a clashing course that works out the logistics of the course so as to avoid the clash. Therefore, there is a great requirement for an application distributing the course evenly and without collisions. My aim here is to develop a simple, easily understandable, efficient and portable application, which could generate good quality timetables within minutes.

**D. LIMITATIONS OF PROJECT**

The app only works in android devices not on iOS devices.

**E. OBJECTIVE OF PROJECT**

The objective of Docket is to enable a student to make timetable, evaluate his /her performance and enhances the scope of improvement.

**SYSTEM ANALYSIS**

**A. FEASIBILITY STUDY**

An initial investigation in a proposal that determines whether an alternative system is feasible. A proposal summarizing the thinking of the analyst is presented to the user for review. When approved, the proposal initiates feasibility study that describes and evaluates candidate systems and provides for the selection of best system that meets system performance requirements.

To do a feasibility study, we need to consider the economic, technical factors in system development. First a project team is formed. The team develops system flowcharts that identify the characteristics of candidate systems, evaluate the performance of each system, weigh system performance and cost data and select the best candidate system for the job. The study culminates in a final report to the management.

**B. INTRODUCTION**

**1.** Describe and identify characteristics of candidate systems.

**2.** Determine and evaluate performance and cost effectiveness of each candidate system.

**3.** Weigh system performance and cost data.

**4.** Select the best candidate system.

**C. SUMMARY**

1. A feasibility study is conducted to select the best system that meets performance requirements. This entails an identification description, an evaluation of candidate systems, and the selection of the best system for the job.

2. A statement of constraints, the identification of specific system objectives and a description of outputs define a system’s required performance. The analyst is then ready to evaluate the feasibility of candidate systems to produce these outputs.

3. Three key considerations are involved in feasibility analysis: economic, technical and behavioral.

4. There are few steps in feasibility study:

**i. STATEMENT OF CONSTRAINTS: -** Constraints are factors that limit the solution of a problem. Some constraints are identified during the initial investigation

**ii. IDENTIFICATION OF SPECIFIC SYSTEM OBJECTIVES:**

**-** Once the constraints are spelled out, the analyst proceeds to identify the system’s specific performance objectives. They are derived from the general objectives specified in the project directive at the end of the initial investigation. The steps are to state the system’s benefits and then

translate them into measurable objectives.

**iii. DESCRIPTION OF OUTPUTS: -** A final step in system performance definition is describing the output required by the user. An actual sketch of the format and contents of the reports as well as a specification of the media used, their frequency, size and numbers of

copies required are prepared at this point.

**D. TYPES OF FEASIBLE STUDY**

**i. Legal Feasibility**: - Determines whether the proposed system conflicts with legal requirements, e.g. a data processing system must comply with the local Data Protection Acts.

ii. **Operational Feasibility: -**Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development

schedule, delivery date, corporate culture, and existing business processes.

To ensure success, desired operational outcomes must be imparted during design and development. These include such design dependent parameters such as reliability, maintainability, supportability, usability, predictability, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviors are to be realized. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters.

A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of

systems engineering that needs to be an integral part of the early design phases

iii. **Economic Feasibility: -**The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will

provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/ benefits analysis.

iv. **Technical Feasibility: -**The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the

expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

**E. HARDWARE AND SOFTWARE REQUIREMENTS**

**1. HARDWARE REQUIREMENTS**

* Processor: Intel atom Processor Z2520 1.2 GHz or above
* Ram: 512 Mega Byte (MB) or Greater
* Hard disk: 1.2 Giga Byte (GB) or Greater

**2. SOFTWARE REQUIREMENTS**

* Operating System: Android 5.0 and above

**DEVELPOMENT ENVIRONMENT**

**A. INTRODUCTION TO SQL DATABASE**

SQL, Structured Query Language is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). It is particularly useful in handling structured data, i.e. data incorporating relations among entities and variables.

SQL offers two main advantages over older read–write APIs such as ISAM or VSAM. Firstly, it introduced the concept of accessing many records with one single command. Secondly, it eliminates the need to specify how to reach a record, e.g. with or without an index.

Originally based upon relational algebra and tuple relational calculus, SQL consists of many types of statements, which may be informally classed as sublanguages, commonly: a data query language (DQL), a data definition language (DDL), a data control language (DCL), and a data manipulation language (DML). The scope of SQL includes data query, data manipulation (insert, update and delete), data definition (schema creation and modification), and data access control. Although SQL is essentially a declarative language (4GL), it also includes procedural elements.

SQL was one of the first commercial languages to utilize Edgar F. Codd’s relational model. The model was described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks”. Despite not entirely adhering to the relational model as described by Codd, it became the most widely used database language.

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987. Since then the standard has been revised to include a larger set of features. Despite the existence of standards, most SQL code requires at least some changes before being ported to different database systems.

**HISTORY**

SQL was initially developed at IBM by Donald D. Chamberlin and Raymond F. Boyce after learning about the relational model from Edgar F. Codd in the early 1970s. This version, initially called SEQUEL (Structured English Query Language), was designed to manipulate and retrieve data stored in IBM's original quasi-relational database management system, System R, which a group at IBM San Jose Research Laboratory had developed during the 1970s.

Chamberlin and Boyce's first attempt at a relational database language was Square, but it was difficult to use due to subscript notation. After moving to the San Jose Research Laboratory in 1973, they began work on SEQUEL. The acronym SEQUEL was later changed to SQL because "SEQUEL" was a trademark of the UK-based Hawker Siddeley Dynamics Engineering Limited company.

After testing SQL at customer test sites to determine the usefulness and practicality of the system, IBM began developing commercial products based on their System R prototype including System/38, SQL/DS, and DB2, which were commercially available in 1979, 1981, and 1983, respectively.

In the late 1970s, Relational Software, Inc. (now Oracle Corporation) saw the potential of the concepts described by Codd, Chamberlin, and Boyce, and developed their own SQL-based RDBMS with aspirations of selling it to the U.S. Navy, Central Intelligence Agency, and other U.S. government agencies. In June 1979, Relational Software, Inc. introduced the first commercially available implementation of SQL, Oracle V2 (Version2) for VAX computers.

By 1986, ANSI and ISO standard groups officially adopted the standard "Database Language SQL" language definition. New versions of the standard were published in 1989, 1992, 1996, 1999, 2003, 2006, 2008, 2011 and, most recently, 2016.

**INTEROPERABILITY AND STANDARDIZATION**

**Overview**

SQL implementations are incompatible between vendors and do not necessarily completely follow standards. In particular date and time syntax, string concatenation, NULLs, and comparison case sensitivity vary from vendor to vendor. Particular exceptions are PostgreSQL and Mimer SQL which strive for standards compliance, though PostgreSQL does not adhere to the standard in how folding of unquoted names is done. The folding of unquoted names to lower case in PostgreSQL is incompatible with the SQL standard, which says that unquoted names should be folded to upper case. Thus, Foo should be equivalent to FOO not foo according to the standard.

Popular implementations of SQL commonly omit support for basic features of Standard SQL, such as the DATE or TIME data types. The most obvious such examples, and incidentally the most popular commercial and proprietary SQL DBMSs, are Oracle (whose DATE behaves as DATETIME, and lacks a TIME type) and MS SQL Server (before the 2008 version). As a result, SQL code can rarely be ported between database systems without modifications.

**Reasons for incompatibility**

There are several reasons for this lack of portability between database systems:

* The complexity and size of the SQL standard means that most implementers do not support the entire standard.
* The standard does not specify database behavior in several important areas (e.g. indexes, file storage...), leaving implementations to decide how to behave.
* The SQL standard precisely specifies the syntax that a conforming database system must implement. However, the standard's specification of the semantics of language constructs is less well-defined, leading to ambiguity.
* Many database vendors have large existing customer bases; where the newer version of the SQL standard conflicts with the prior behavior of the vendor's database, the vendor may be unwilling to break backward compatibility.
* There is little commercial incentive for vendors to make it easier for users to change database suppliers (see vendor lock-in).
* Users evaluating database software tend to place other factors such as performance higher in their priorities than standards conformance.
* Standardization history
* SQL was adopted as a standard by the American National Standards Institute (ANSI) in 1986 as SQL-86[29] and the International Organization for Standardization (ISO) in 1987. It is maintained by ISO/IEC JTC 1, Information technology, Subcommittee SC 32, Data management and interchange.
* Until 1996, the National Institute of Standards and Technology (NIST) data management standards program certified SQL DBMS compliance with the SQL standard. Vendors now self-certify the compliance of their products.

The original standard declared that the official pronunciation for "SQL" was an initialism: /ˌɛsˌkjuːˈɛl/ ("ess cue el"). Regardless, many English-speaking database professionals (including Donald Chamberlin himself) use the acronym-like pronunciation of /ˈsiːkwəl/ ("sequel"), mirroring the language's pre-release development name, "SEQUEL".

**B. INTRODUCTION TO ANDROID STUDIO**

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on Jet Brains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020 It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.

On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development. Java is still supported, as is C++.

Features

The following features are provided in the current stable version:

* Gradle-based build support
* Android-specific refactoring and quick fixe
* Lint tools to catch performance, usability, version compatibility and other problems
* ProGuard integration and app-signing capabilities
* Template-based wizards to create common Android designs and components
* A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations
* Support for building Android Wear apps
* Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine[[18]](https://en.wikipedia.org/wiki/Android_Studio#cite_note-18)
* Android Virtual Device (Emulator) to run and debug apps in the Android studio.

Android Studio supports all the same programming languages of IntelliJ (and [CLion](https://en.wikipedia.org/wiki/CLion)) e.g. Java, C++, and more with extensions, such as [Go](https://en.wikipedia.org/wiki/Go_(programming_language));[[19]](https://en.wikipedia.org/wiki/Android_Studio#cite_note-19) and Android Studio 3.0 or later supports Kotlin and "all Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects [backport](https://en.wikipedia.org/wiki/Backporting) some Java 9 features. While IntelliJ states that Android Studio supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.

Once an app has been compiled with Android Studio, it can be published on the Google Play Store. The application has to be in line with the Google Play Store developer content policy

## System requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Basic system requirements for Android Studio | | | |
|  | Microsoft Windows | Mac | Linux |
| Operating System Version | Microsoft® Windows® 7/8/10 (32- or 64-bit)  The Android Emulator only supports 64-bit Windows. | Mac® OS X® 10.10 (Yosemite) or higher, up to 10.14 (macOS Mojave) | GNOME or KDE desktop  Tested on gLinux based on Debian (4.19.67-2rodete2). |
| Random Access Memory (RAM) | 4 GB RAM minimum; 8 GB RAM recommended. | | |
| Free disk space | 2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image). | | |
| Minimum required JDK version | Java Development Kit 8 | | |
| Minimum screen resolution | 1280 x 800 | | |

The Android Emulator has additional requirements beyond the basic system requirements for Android Studio, which are described below:

* SDK Tools 26.1.1 or higher;
* 64-bit processor;
* Windows: CPU with UG (unrestricted guest) support;
* Intel Hardware Accelerated Execution Manager (HAXM) 6.2.1 or later (HAXM 7.2.0 or later recommended).

The use of hardware acceleration has additional requirements on Windows and Linux:

* Intel processor on Windows or Linux: Intel processor with support for Intel VT-x, Intel EM64T (Intel 64), and Execute Disable (XD) Bit functionality;
* AMD processor on Linux: AMD processor with support for AMD Virtualization (AMD-V) and Supplemental Streaming SIMD Extensions 3 (SSSE3);
* AMD processor on Windows: Android Studio 3.2 or higher and Windows 10 April 2018 release or higher for Windows Hypervisor Platform (WHPX) functionality.

To work with Android 8.1 (API level 27) and higher system images, an attached webcam must have the capability to capture 720p frames.

**C. INTRODUCTION TO JAVA**

Java is an object oriented programming language originally developed by Sun Microsystems and released in 1995.

Java was originally developed by James Gosling at Sun Microsystems (which has since merge into Oracle Corporation).

Java programs are platform independent which means they can be run on any operating system with any type of processor as long as the Java interpreter is available on that system.

Java code that runs on one platform does not need to be recompiled to run on another platform, it’s called “write once, run anywhere” (WORA).

Java virtual machine (JVM) executes Java code, but is written in platform specific languages such as C/C++/ASM etc. JVM is not written in Java and hence cannot be platform independent and Java interpreter is actually a part of JVM.

1. **USES OF JAVA**

Earlier, java was only used to design and program small computing devices but later adopted as one of the platform independent programming language and now according to Sun, 3 billion devices run java. Java is one of the most important programming language in today’s IT industries.

JSP – Java is used to create web applications like PHP and ASP, JSP (Java Server Pages) used with normal HTML tags, which helps to create dynamic web pages.

Applets – This is another type of Java program that used within a web page to add many new features to a web browser.

J2EE – The software Java 2 Enterprise Edition are used by various companies to transfer data based on XML structured documents between one another.

JavaBeans – This is something like Visual Basic, a reusable software component that can be easily assemble to create some new and advanced application.

Mobile – Besides the above technology, Java is also used in mobile devices, many kinds of games and services built in Java. Today, all leading mobile service provider like Nokia, Siemens, Vodafone are using Java technology.

1. **Types of Java Applications**

Web Application Java is used to create server side web applications. Currently, servlet, jsp, struts, jsfetc technologies are used.

Standalone Application It is also known as desktop application or window-based application. An application that we need to install on every machine or server such as media player, antivirus etc. AWT and Swing are used in java for creating standalone applications.

Enterprise Application an application that is distributed in nature, such as banking applications etc. It has the advantage of high level security, load balancing and clustering. In java, EJB is used for creating enterprise applications.

Mobile Application Java is used to create application software for mobile devices. Currently Java ME is used for creating applications for small devices, and also Java is programming language for Google Android application development.

1. **Facts about Java**

Object Oriented – In java everything is an Object. Java can be easily expanded since it is based on the Object model.

Platform independent – C and C++ are platform dependency languages hence the application programs written in one Operating system cannot run in any other Operating system, but in platform independence language like Java application programs written in one Operating system can able

to run on any Operating system.

Simple – Java is designed to be easy to learn. If you understand the basic concept of OOP java would be easy to master.

Secure – With Java’s secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public key encryption.

Architectural neutral – Java compiler generates an architecture neutral object file format which makes the compiled code to be executable on many processors, with the presence Java runtime system. Portable – being architectural neutral and having no implementation dependent aspects of

the specification makes Java portable. Compiler and Java is written in ANSI C with a clean portability boundary which is a POSIX subset.

Robust – Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.

Multithreaded – With Java’s multithreaded feature it is possible to write programs that can do many tasks simultaneously. This design feature allows developers to construct smoothly running interactive applications.

Interpreted – Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light weight process.

High Performance – With the use of Just-In-Time compilers Java enables high performance.

Distributed – Java is designed for the distributed environment of the internet.

Dynamic – Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry an extensive amount of runtime information that can be used to verify and resolve accesses to objects on runtime.

1. **Different Editions of Java Technology**

Java SE – Java SE or Java Standard Edition provides tools and API’s that you can use to create server applications, desktop applications, and even applets. These programs developed using Java SE can be run on almost every popular operating system, including Linux, Macintosh,

Solaris, and Windows.

JEE – Based on the foundation framework of the standard edition, Java Enterprise Edition helps in web application service, component model and enterprise class service oriented architecture (SOA).

JME – Java Micro Edition or JME for short is an accumulation of Java APIs that are used for the development of software for devices like mobile phones, PDAs, TV set top boxes, game programming. The platform of micro edition generally consists of an easy user interface, a robust security model and a wide variety of built-in networks for running Java based application.

1. **Popular Java Editors**

To write your java programs you will need a text editor. There are even more sophisticated IDE available in the market. But for now, you can consider one of the following:

Notepad – On Windows machine you can use any simple text editor like Notepad (Recommended for this tutorial), Text Pad.

NetBeans – is a Java IDE that is open source and free which can be downloaded from www.netbeans.org/index.html

Eclipse – is also a java IDE developed by the eclipse open source community and can be downloaded from <http://www.eclipse.org>

**SYSTEM DESIGN**

**MODULAR DESCRIPTION**

**1. FRONT PAGE**

In this module new student or user can see the timetable he/she has made for a particular day and time.

All these details gets stored in the database, creating record for each student also

these details can be accessed further

**2. ADDING TIMETABLE**

Here the students are add the timetable by providing the Title of the timetable , start ti me and end time. And by selecting a particular color to the timetable made and then save.

.

**3. SETTING PAGE**

i. Here the user can change the Number of Weeks according to the need

ii. Can turn on and off the timetable for the weekends.

iii. Can turn on and off the notifications

iv. Add a home screen of the timetable made.

v. Reset the timetable.

**4. TASK PAGE**

Here the user can see the number of task that he/ she has made. If the user is new then this area is blank and the user can tap on the “+” icon to add a new task.

**5. ADDING TASK**

Here the user can add the task by simply adding the Title of the task, Date till which he/ she has to complete the task and category(optional).

**IMPLEMENTATION AND TESTING**

**A. TESTING**

Testing is the process of exercising software with the intent of finding errors and ultimately correcting them. The following testing techniques have been used to make this project free of errors.

**i. Content Review**

The whole content of the project has been reviewed thoroughly to uncover typographical errors, grammatical error and ambiguous sentences.

**ii. Navigation Errors**

Different users were allowed to navigate through the project to uncover the navigation errors. The views of the user regarding the navigation flexibility and user friendliness were taken into account and implemented in the project.

**iii. Unit Testing**

Focuses on individual software units, groups of related units.

Unit – smallest testable piece of software.

A unit can be compiled /assembled / linked/loaded; and put under a test

harness.

Unit testing done to show that the unit does not satisfy the application and /or its implemented software does not match the intended designed structure.

**iv. Integration Testing**

Focuses on combining units to evaluate the interaction among them

Integration is the process of aggregating components to create larger components.

Integration testing done to show that even though components were individually satisfactory, the combination is incorrect and inconsistent.

**v. System testing**

Focuses on a complete integrated system to evaluate compliance with specified requirements (test characteristics that are only present when entire system is run)

A system is a big component.

System testing is aimed at revealing bugs that cannot be attributed to a

component as such, to inconsistencies between components or planned interactions between components.

Concern: issues, behaviors that can only be exposed by testing the entire integrated system (e.g., performance, security, recovery) each form encapsulates (labels, texts, grid etc.). Hence in case of project in V.B. form are the basic units. Each form is tested thoroughly in term of calculation, display etc.

**vi. Regression Testing**

Each time a new form is added to the project the whole project is tested thoroughly to rectify any side effects. That might have occurred due to the addition of the new form. Thus regression testing has been performed.

**vii. White-Box testing**

White-box testing (also known as clear box testing, glass box testing, transparent box testing and structural testing) tests internal structures or workings of a program, as opposed to the functionality exposed to the end-user. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT). While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test

paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

Techniques used in white-box testing include:

**API testing (application programming interface**) – testing of the application using public and private APIs

**Code coverage** – creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)

**Fault injection methods** – intentionally introducing faults to gauge the efficacy of testing strategies

Code coverage tools can evaluate the completeness of a test suite that was created with any method, including black-box testing. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested. Code coverage as a software metric can be reported as a percentage for:

Function coverage, which reports on functions executed

Statement coverage, which reports on the number of lines executed to complete

the test

100% statement coverage ensures that all code paths, or branches (in terms of control flow) are executed at least once. This is helpful in ensuring correct functionality, but not sufficient since the same code may process different inputs correctly or incorrectly.

**viii. Black-box testing**

Black-box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation. The tester is only aware of what the software is supposed to do, not how it does it. Black-box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, state transition tables, decision table testing, fuzz testing, model-based

testing, use case testing, exploratory testing and specification-based testing.

Specification-based testing aims to test the functionality of software according to the applicable requirements. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case. Test cases are built around

specifications and requirements, i.e., what the application is supposed to do. It uses external descriptions of the software, including specifications, requirements, and designs to derive test cases. These tests can be functional or non-functional, though usually functional.

Specification-based testing may be necessary to assure correct functionality, but it is insufficient to guard against complex or high-risk situations.

One advantage of the black box technique is that no programming knowledge is required. Whatever biases the programmers may have had, the tester likely has a different set and may emphasize different areas of functionality. On the other hand, black-box testing has been said to be "like a walk in a dark labyrinth without a flashlight." Because they do not examine the source code, there are

situations when a tester writes many test cases to check something that could have been tested by only one test case, or leaves some parts of the program untested.

This method of test can be applied to all levels of software testing: unit, integration, system and acceptance. It typically comprises most if not all testing at higher levels, but can also dominate unit testing as well.

**ix. Alpha Testing**

Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.

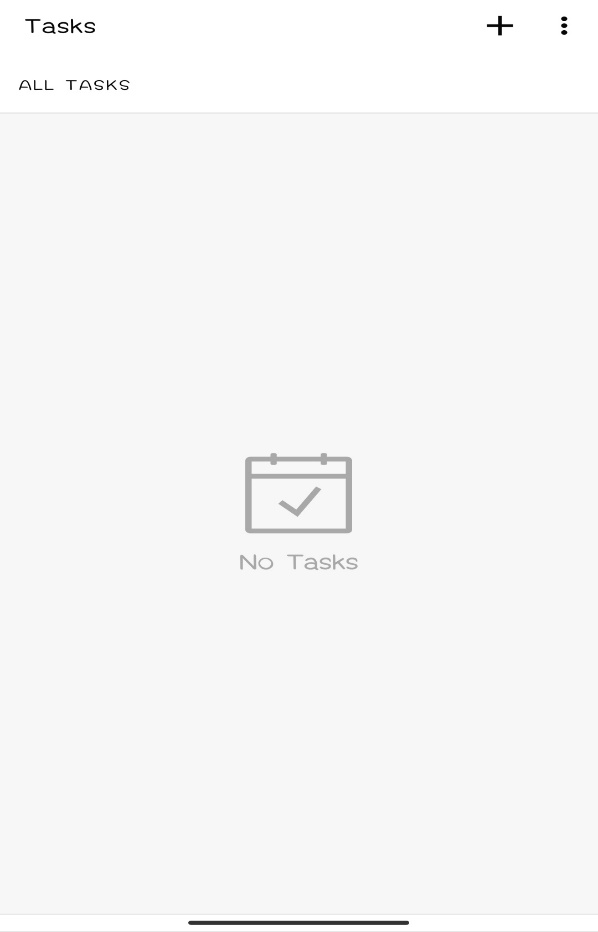
**x. Beta Testing**

Beta testing comes after alpha testing and can be considered a form of external user acceptance testing. Versions of the software, known as beta versions, are released to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or bugs. Sometimes, beta versions are made available to the open

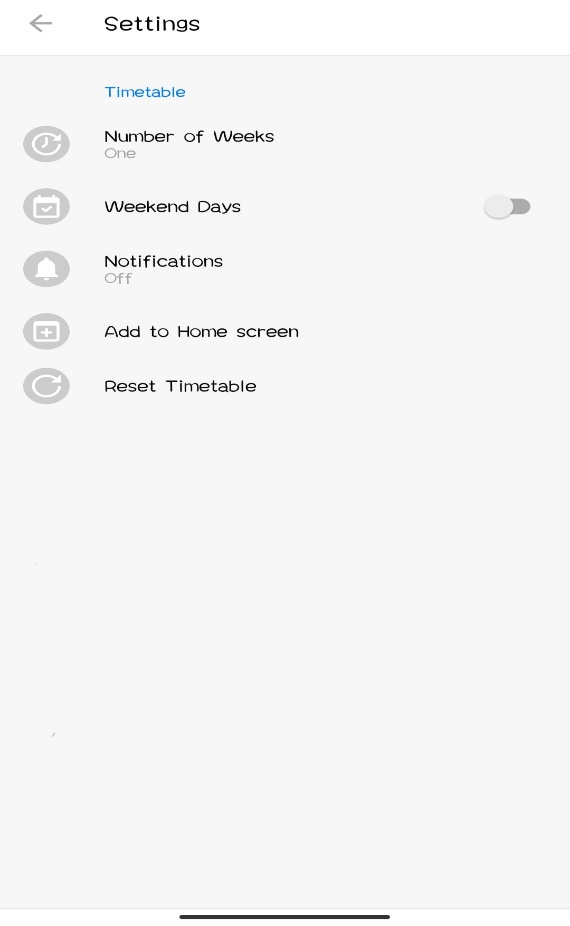
public to increase the feedback field to a maximal number of future users.

**B. IMPLEMENTATION**

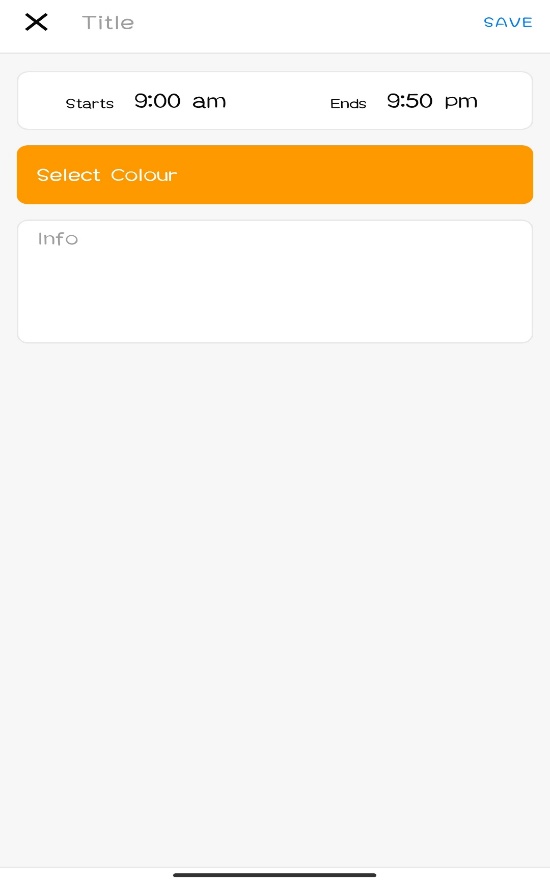
**i. LANDING PAGE**

****

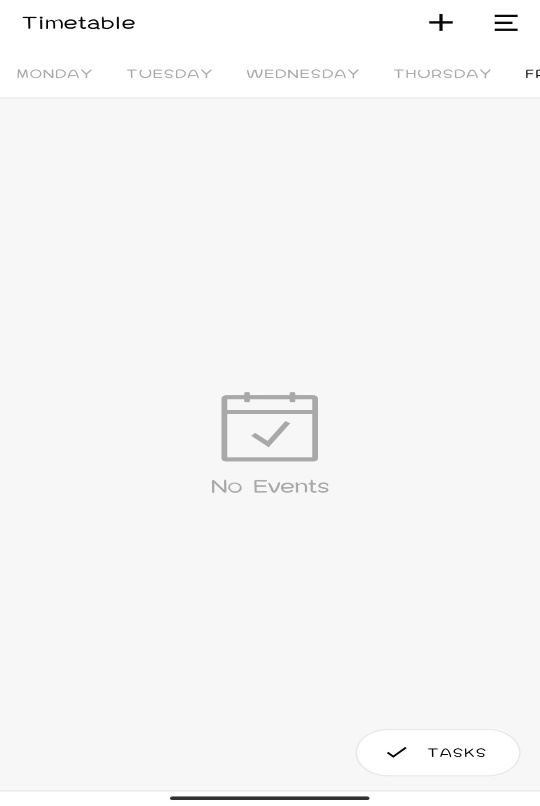
**ii. SETTINGS PAGE**

****

**iii. ADDING TIMETABLE PAGE**

****

**iv. ADDING TASK PAGE**

****

**CONCLUSION**

The project “**DOCKET**” aims to simplify the process of organizing or creating

the student’s timetable by making it user friendly.

This project makes the whole process automated as user just need to enter few

details in this to get started and then he/she can choose the subject of his choice just by tapping in the color which he/she has generated earlier. Result is generated

automatically

This project cover very much every function needed by a student in a college or school.

**BIBLIOGRAPHY**

**1.** The complete reference, Java 2, 5th edition by Herbert Schildt.

**2.** www.javatpoint.com/java

**3.** www.google.co.in

**4.** Oracle: A Beginner’s Guide by Michael Abbey and Michael J. Korey.

**5**. www.youtube.co.in

**6**. www.udemy.co.in