**TIME TABLE MANAGEMENT SYSTEM**

*An Application Development – 1 (Project) Report Submitted*

*In partial fulfillment of the requirement for the award of the degree of*

***Bachelor of Technology in***

***Computer Science and Engineering (Cyber Security)***

# by

## A.JOSEPH 22N35A6201

**S.RAMYA 21N31A6256**

## M.MANISAGAR 21N31A6243

***Under the Guidance of***

## Dr P. DILEEP

**Professor**

**Department of CSE (Emerging Technologies) MRCET (Autonomous Institution, UGC Govt. of India)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**(EMERGING TECHNOLOGIES)**

**MALLA REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(Autonomous Institution – UGC, Govt. of India)**

**(Affiliated to JNTU, Hyderabad, Approved by AICTE, Accredited by NBA & NAAC – ‘A’ Grade, ISO 9001:2015**

**Certified)**

Maisammaguda (v), Near Dullapally, Via: Kompally, Hyderabad – 500 100, Telangana State, India

**2023-2024**

# DECLARATION

We hereby declare that the project entitled **“Time Table Management System”** submitted to **Malla Reddy College of Engineering and Technology,** affiliated t**o** Jawaharlal Nehru Technological University Hyderabad (JNTUH) as part of III Year B.Tech – I Semester and for the partial fulfillment of the requirement for the award of **Bachelor of Technology** in **Computer Science and Engineering (Cybersecurity)** is a result of original research work done by us.

It is further declared that the project report or any part thereof has not been previously submitted to any University or Institute for the award of degree or diploma.

# A.JOSEPH (22N35A6201) S.RAMYA (21N31A6256) M.MANISAGAR (21N31A6243)



**CERTIFICATE**

This is to certify that this is the Bonafide record of the project titled “**Time Table Management System**”, submitted by **JOSEPH** (22N35A6201) **RAMYA (**21N31A6256) and **MANISAGAR (**21N31A6243) of **B.Tech III YEAR – I Semester** in the partial fulfillment of the requirements for the degree of **Bachelor of Technology** in **Computer Science and Engineering (Cyber Security)**, Dept. of CSE (Emerging Technologies) during the year 2023-2024. The results embodied in this project report have not been submitted to any other university or institute for the award of any degree or diploma.

**Dr. M V Kamal Dr. P Dileep**

Head Of the Department (ET) Project Coordinator

**Department of CSE (ET) Department of CSE (ET)**

**EXTERNAL EXAMINER Dr. M V Kamal**

**Professor &**

**Head of the Department**

**ACKNOWLEDGEMENT**

We feel ourself honored and privileged to place our warm salutation to our college “Malla Reddy College of Engineering and Technology (Autonomous Institution – UGC Govt. of India) and our Principal **Dr. S Srinivasa Rao,** Professor who gave us the opportunity to do the Application Development -1 (Project) during our III Year B.Tech and profound the technical skills.

We express our heartiest thanks to our Director **Dr. V S K Reddy,** Professor for encouraging us in every aspect of our project and helping us realize our full potential.

We are also thankful to our Head of the Department **Dr. M V Kamal,** Professor for providing training and guidance, excellent infrastructure, and a nice atmosphere for completing this project successfully.

We would like to express our sincere gratitude and indebtedness to our project supervisor

**Dr. I. Nagaraju ,** Professor for his valuable suggestions and interest throughout the course of this project.

We convey our heartfelt thanks to our Project Coordinator **Dr. P Dileep,** Professor for allowing for their regular guidance and constant encouragement during our dissertation work.

We would like to thank all our staff of the Department of CSE (Emerging Technologies) and even all other department who have been helpful directly and in-directly in making our project a success.

Finally, we would like to take this opportunity to thank our **family** for their support and blessings for completion of our project that gave me the strength to do my project.

**D.SAIKIRAN (20N31A6714) M.LUCKYDHAR (20N31A6741) MD.REHAN PASHA (20N31A6742)**

**ABSTRACT**

Timetable generation in most colleges and schools is done manually by writing over the data which consumes a lot of time and in addition to that it’s a “hectic” job to do. Again, there is a high probability of getting drawn into an error like clash among the classes or having two particular classes in the same classroom or with the same faculty having more than one lecture at a particular time. Timetable preparation is a complex task and requires patience.

There are many drawbacks in the current system where we enter data manually. So schools are unable to design the timetable in time and run into multiple problems. These are nothing but common human errors which are not easy to prevent in a task like these. Just to overcome such issues we propose a automated system which comes with precision, accuracy and ease. The automatic timetable generator will be taking number of inputs such as subject name, subject credits or frequency and number of timetables to be printed, and so on. Depending upon the inputs that are entered a possible timetable will be generated for you in no time along with accuracy.

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Chapter No.** | | **Contents** | **Page no** |
| 1 |  | Introduction | 1 |
|  | 1.1 | Problem Definition | 2 |
|  | 1.2 | Scope | 2 |
|  | 1.3 | Existing System | 3 |
|  | 1.4 | Proposed System | 4 |
| 2 |  | LITERATURE SURVEY | 6 |
| 3 |  | System Requirements |  |
|  | 3.1 | Software Requirements | 9 |
|  | 3.2 | Hardware Requirements | 9 |
| 4 |  | System Design |  |
|  | 4.1 | Dataflow Diagrams / UML Diagrams | 10 |
|  | 4.2 | ER Diagrams | 11 |
| 5 |  | Implementation | 13 |
| 6 |  | Testing | 22 |
| 7 |  | Results | 26 |
| 8 |  | Conclusion | 30 |
|  |  | References | 32 |

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **S.NO** | **FIGURE TITLE** | **PAGE No** |
| 1 | Time Table Diagram | 5 |
| 2 | Data Flow Diagram | 10 |
| 3 | ER Diagram | 11 |
| 4 | Use Case Diagram | 11 |
| 5 | Sequential Diagram | 12 |
| 6 | Dashboard | 26 |
| 7 | Create Batch | 26 |
| 8 | Create Course | 27 |
| 9 | Create Professor | 27 |
| 10 | Allotment of Batches | 28 |
| 11 | Time table generation | 28 |
| 12 | Deletion of Batches, Courses, professors | 29 |

**CHAPTER 1 INTRODUCTION**

Keeping track of classes and exams may be a significant challenge in the world of education. This is where the Automatic Time Table Management System (ATMS) comes in handy; it functions much like an extremely intelligent assistant when it comes to setting up and managing academic schedules.

Saying goodbye to the hassle of having to manually determine who goes where and when is one thing. ATMS accomplishes this by utilizing sophisticated technologies to streamline the schedule creation process. It's similar to having a wizard to help universities and institutions make efficient use of their time and resources.

Colleges may avoid the turmoil of competing timetables and promptly adjust to any changes using ATMS. It's important to make things simple for everyone, not just scheduling. Even non-techies may easily customize schedules with this system because it is so user-friendly.

We'll look at how ATMS makes life easier for universities and schools in this document. Prepare to enter a world where making schedules is a snap, allowing teachers to concentrate more on what matters most—delivering an excellent education.

. Technology makes life easier by furnishing better support to different systems, better exactness, easy maintenance, etc. Now a day's technology ultimately means “computers”. Currently computers are getting more popular because of ease of literacy, accuracy, ease of work, with the least time consumption, ease of conservation with cost effectiveness.

## Problem Definition

The key to managing any academy or institution is having a well-planned schedule devoid of conflicts. The academic institution creates timetables by hand. Creating academic calendars is a laborious effort that must be completed by every institution. Timetabling is the process of putting together a schedule for all activities that needs to be kept confidential due to various restrictions.

## Scope

Educational institutions can simplify scheduling with the use of the Time Table Management System. It has a user-friendly interface, multi-user accessibility, and automated scheduling. Enhanced adaptability is achieved through alerting systems, resource management, and customization possibilities. In addition to ensuring security and facilitating smooth integration with other technologies, the system offers reporting and analytics. This paper is meant to be a user-friendly manual for creating and managing timetables in an effective and scalable manner.

Academic timetables in educational institutions can be created, viewed, and modified automatically with the help of the Time Table Management System. It allows dynamic updates and has user-friendly interfaces for batch creation, schedule viewing, course creation, and professor management. The system is scalable, integrates images, manages problems, and takes credit hours into account. It offers a complete answer for effective scheduling administration in educational settings.

By including tabs for different functions including examining current schedules, adding new batches, courses, and professors, and controlling deletions, the system encourages user-friendliness. It lets institution logos to be included, which guarantees an aesthetically pleasing interface.

In order to construct and manage well-organized schedules, users can designate professors and courses. When changes take place, the system automatically updates timetables, allowing for real-time adjustments. It places a strong emphasis on error management, warning users of problems such as inadequate resources while adding courses to batches.

## Existing System:

For educational institutions and organisations to effectively organize and manage schedules for varied activities, time table management systems are essential tools. By streamlining the creation, distribution, and updating of timetables, these solutions guarantee the best possible use of both time and resources. In plain English, let's examine the main elements and functionalities of an established timetable management system.

A Time Table Management System is essentially a computerized tool created to manage the scheduling challenges in educational establishments like colleges, universities, and schools. The main objective is to streamline the timetable generation and management process, resulting in a more organized and efficient procedure.

An intuitive user interface that is available to administrators, teachers, and occasionally students makes up the majority of the system. Administrators are frequently the ones in charge possess strong tools at their disposal to help them create timetables that effectively allocate students, instructors, and resources in accordance with predetermined criteria and preferences.

The ability to enter and manage several factors, such as available classes, teacher preferences, and specific subjects, is one of the core advantages. With the use of this data, the system creates a schedule that minimizes conflicts between teachers' availability during specific hours and the scheduling of classes.

These systems frequently permit fast alterations in order to guarantee flexibility. For instance, administrators can quickly modify the programme without affecting the entire schedule if a teacher becomes unavailable due to unanticipated events. In the ever-changing environment of educational institutions, this flexibility is essential.

The system takes into account variables like classroom capacity to aid in resource optimization.as well as accessibility. It makes sure that every class is placed in the right space, preventing crowding and fostering a positive learning atmosphere.

The system might also include functions like automated clash detection, which would notify administrators of any scheduling problems that require attention. This avoids problems like students having incompatible exam schedules or teachers being assigned to different classes at the same time.

Advanced Time Table Management Systems have the potential to interface with various educational software programmes, including learning management systems and student information systems. This integration guarantees smooth data

## Proposed System:

The Time Table Management System is a Java-based application designed to facilitate the creation, viewing, and management of timetables for educational institutions. The system employs a graphical user interface (GUI) using the Swing framework to provide an interactive and user-friendly experience.

**Key Features**:

1. **Home Page**:

- Upon launching the application, users are greeted with a welcoming home page displaying the system name and a relevant image.

- The interface is aesthetically pleasing, with a combination of text and images.

2. **View Timetable**:

- Users can view timetables for batches and professors through a dedicated tab.

- A dynamic table displays the schedule for each day of the week, and users can select specific batches or professors to view their respective timetables.

3. **Create Batch**:

- The system allows users to create batches, defining a group of students with a unique name.

- Users can associate courses and professors with batches, building a comprehensive timetable.

4. **Create Professor**:

- Professors can be added to the system with their respective names.

- Courses can be assigned to professors, providing a linkage between faculty and subjects.

5. **Create Course**:

- Users can create courses by specifying a name and the associated credit hours.

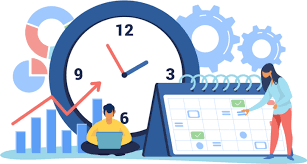
- Courses can be linked to batches, professors, and viewed in the timetable.

6. **Delete Functionality**:

- The system supports the deletion of courses, professors, and batches.

- Users can easily remove unwanted elements from the system, maintaining data integrity.

- The system includes error handling mechanisms, such as alerts, to notify users of invalid selections or insufficient resources when attempting to add courses to batches.



**CHAPTER-2**

# LITERATURE SURVEY

Jaydev Bhatti from Chitkara University published a research paper on timetable handling mechanism using python in journal named as international journal of scientific & technology research in march, 2020. [2]

This research paper created by the Chitkara university shows how timetable was generated for three different categories i.e.. Faculty wise for faculty members, class wise for students, as well as classroom wise timetable. The system was capable of providing a separate schedule timetable every time, we change the hard constraints for the program which is quite fascinating and appealing. [2]

The interface generated by the timetable was impressive as well, easy to understand and easy to use. It solved Many existential problems that existed in the current manual way of timetable generation. Solutions to help problems like overlapping of two subjects, classroom clashes and faculty clashes. This is time saving as well as easy to handle. [2]

Abhinayal along with Sahithi and Akaanksha published a research paper on online application of automatic Timetable generator using python as programming language in August, 2016.[1]

EA (Evolutionary algorithm) is used to solve a large limitations-based university timetable issue. In their approach, context- based reasoning methods are utilized for achieving realistic timetables in minimum time. A combinatorial optimization problem developed to solve the university timetabling problem where a set of measures has to be planed into time periods and established into suitable rooms. Manually generated timetables is maintained, it is universal problem due to lack of suitable methods implemented.[1]

Sheng Xiang Yang, a member of IEEE along with Sadaf Naseem Jat published a research paper on a mimetic algorithm for university course timetabling problem.[3]

The university course timetabling problem (UCTP) is a combinatorial optimization problem, which contains various events scheduled at their appropriate locations. The formulation of timetables for schools and colleges is quite a bit difficult task, as it is nondeterministic polynomial time problem (NP problem). This research paper discuss about generic algorithm along with guided search and local search techniques. The guided search strategy which is used here is to create offspring into the population based on a data structure that stores information extracted from good individuals of previous generations. [3]

The LS techniques use their exploitive search ability to improve the search efficiency of the proposed GAs and the quality of individuals. The proposed GAs is tested on two sets of benchmark problems in comparison with a set of state-of the- art methods from the literature. The experimental results show that the proposed GAs is able to produce promising results for the UCTP. [3]

Vittoria Maniezzo, Alberto colorni, Marco Dorigo, proposed a Genetic Algorithm To Solve The Timetable Problem, Centre for Emergent Computing.[5]

Get your schedules done easily Import data using copy/paste Schedule multiple weeks at flexible times Drag cards in both master and individual views Avoid conflicts using extra views Divide classes into groups to handle electives. Get notified on the progress. [5]

Crafted for 30+ years to solve the most challenging constraints Save weeks of hard mind bending work Pin cards you don't want to be moved by the generator Mark forbidden slots e.g. for part-time teachers Distribute subjects evenly across the week/class Optimize teacher/student load, gaps, room usage, building moves. Find optimal solution even in "mission impossible" cases Share, collaborate, publish and enjoy Students/teachers can access their timetables from any device Personalize timetables with 10+ skins available Print master and individual timetables Export to various formats and Apps Made a mistake?[5]

Shabina Sayed, Ansari Ahmed, Ansari Aamir, and Ansari Zaeem, published by International journal for innovative research in science & Technology in September, 2015. This Research paper addresses the Problem of Timetabling, which covers a very broad range of real problems faced continually in educational institutes, and they describe how Evolutionary Algorithms (EAs) helped them to solve this real problem.[4]

Automated Timetable Generator application is simplify the process of time table generation smoothly and without efforts which may otherwise needed to done manually which takes lots of time and many brains, possibly leading to constraints problems that are diﬃcult to determine while making manual timetable considering each and every constraints is very difficult. So this application helps to provide a optimal solution.[4]

## CHAPTER 3

## System Requirements

* 1. **Software Requirements:**

|  |  |
| --- | --- |
| NAME OF THE COMPONENT | SPECIFICATION |
| operating system | windows 7,Linux |
| Libraries | Java AWT, SWING |
| Browser | Chrome, Microsoft edge |
| Programming language | Java Programming |

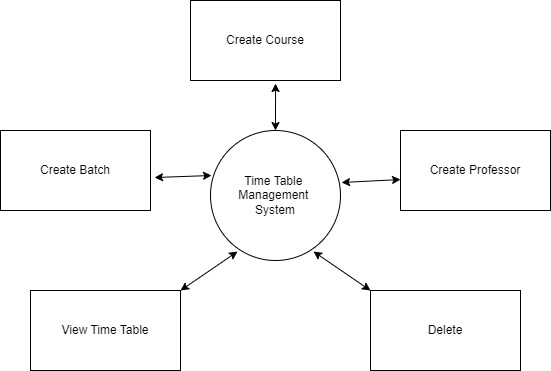
* 1. **Hardware Requirements:**

|  |  |
| --- | --- |
| NAME OF THE COMPONENT | SPECIFICATION |
| Processor | Standard processor with 1.6GHZ |
| RAM | Minimum 4GB RAM |
| Hard Disk | Minimum 10 GB |

# CHAPTER-4

## System Design

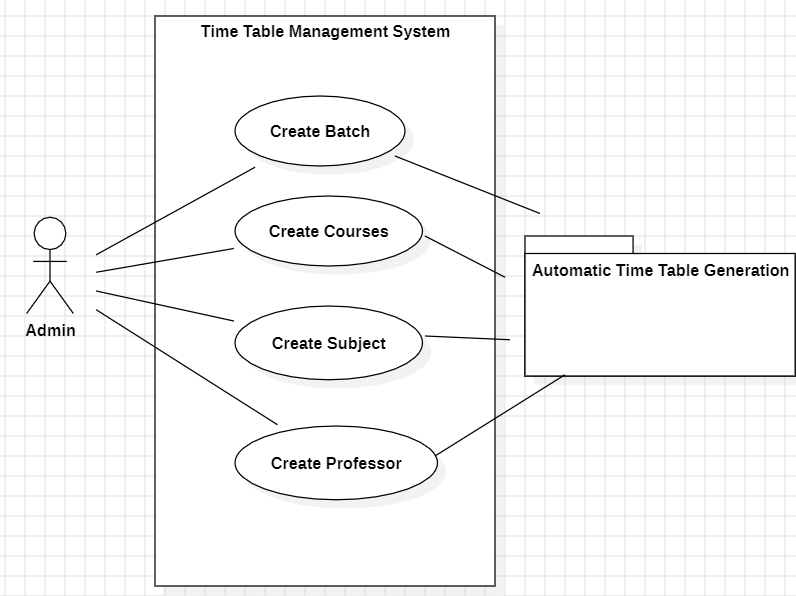
* 1. **Dataflow Diagrams / UML Diagrams**



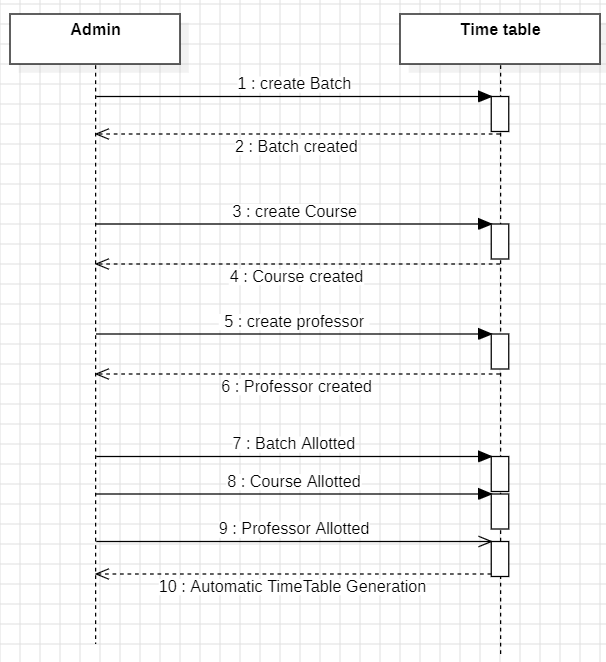
## ER Diagram

## 

**USER USE CASE DIAGRAM:**

****

**RELATIONSHIP DIAGRAM**

****

# CHAPTER 5

## IMPLEMENTATION

This Java application seems to be a simple Swing GUI toolkit implementation of a timetable generator. Several components, including JFrame, JPanel, JLabel, JComboBox, JButton, JTable, and JScrollPane, are used by the programme to build an interactive interface for batch, professor, and course management.

To store instances of Batch, Course, and Professor, the programme first creates three HashMaps and gives each one a distinct name. These entities stand in for the essential components that go into creating a timeline. The graphical user interface is divided into multiple panels, each with a distinct function.

1. **Home Panel**: - Shows an image and a welcome message.

- Makes use of a backdrop colour and null layout to accommodate the positioning of bespoke components.

2**. View Panel**: – This lets customers see the schedule by choosing a professor or batch from JComboBoxes.

- Adjusts the timetable display dynamically according to the batch or professor that is selected.

- Creates a tabular representation of the timetable using JScrollPane and JTable.

3. **Delete Panel**: - Offers the ability to remove a batch, professor, or course.

- Selects the item to be deleted using JComboBoxes and provides buttons for deletion in associated rows.

- A unique component placement scheme with a background colour.

4. **Create Batch Panel**: facilitates the process of naming and assigning courses to a new batch.

- Chooses professors and courses for batch creation using JComboBoxes.

- Has buttons to complete the batch creation and add more courses to the batch.

5. **Establish a Professor Panel**:

- Assists in giving a new professor a name and courses to teach.

- Chooses items using JComboBox Options for adding courses to the professor and completing professor creation are provided in the course for the professor assignment.

6. **Create Panel for Course**:

- Permits the establishment of a brand-new course with designated credits and a name.

- Makes use of JComboBox to choose the course credit total.

- Has a button to complete the construction of the course.

Event listeners are used by the programme to react to user input, such as clicking buttons or selecting things from JComboBoxes. Operations involving file input and output have exception handling implemented. JOptionPanel is also used to show the user notifications and informative messages.

All things considered, the application provides a graphical interface for handling timetable-related objects, with features for viewing, adding, and removing batches, professors, and courses. The arrangement of various panels in a tabbed interface improves the user experience, resulting in a more participatory and user-friendly schedule creation process.

## INTRODUCTION TO JAVA

Java is a popular object-oriented programming language that was created by Sun Microsystems and published in 1995. It is well-known for being platform independent, easy to use, and adaptable. This paper offers a succinct introduction to Java, emphasizing its main characteristics and importance in the field of software development.

**Important characteristics**:

1. **Object-Oriented Paradigm**: highlights the use of classes and objects to create modular, well-organized code. Promotes the ideas of polymorphism, inheritance, and encapsulation.

2. **Platform Independence**: - Follows the principle of "write once, run anywhere".

- When code is compiled, it becomes bytecode, which may run on any device that has the Java Virtual Machine (JVM) installed.

3. **Syntax and Structure**: - Readable and uncomplicated syntax, akin to languages based on C. - Made to be easily understood and learned.

4. **Memory Management**: - Memory management is made easier with automatic garbage collection. - Reduces the danger of memory leaks.

5. **Rich Standard Library**: It is an extensive compilation of pre-made classes and packages.

- Encourages code reuse and simplifies routine programming tasks.

6. **Multithreading**: - Multithreading is supported natively.

- Promotes simultaneous execution, improving the responsiveness of the application.

7. **Security**: - The first emphasis was on using applets to enable secure execution in web browsers.

- Constant dedication to strong security procedures.

8. **Community and Ecosystem**: - Broad and vibrant developer community; - Vast ecosystem of tools, frameworks, and libraries for various applications.

9. **Enterprise-Level Applications**: - Often used in the development of enterprise-level applications.

## WHY TO USE JAVA?

## Java is a popular and flexible programming language that is important to the creation of a timetable management system for a number of reasons. Java is a great choice for developing reliable and effective applications, such as a timetable management system, because of its ease of use, independence from platforms, and abundance of libraries.

## Platform independence is a major factor in the use of Java in time table management systems. Any hardware or operating system with a Java Virtual Machine (JVM) can run Java programmes. This guarantees that the timetable management system is accessible and functional across various platforms, which is advantageous for users who may possess a variety of devices, including Macs, Linux computers, and Windows PCs.

## Java's object-oriented design, which encourages modular and well-organized code, is another important feature. Different entities, including batches, courses, and professors, must be represented and managed in a timetable management system. Because of Java's object-oriented features, developers can represent these entities as objects with attributes and behaviours, which helps to organise and maintain the code.

## Creating graphical user interfaces (GUIs) with Java's extensive library, especially the Swing library, is crucial for creating interfaces that are easy to use in time management systems. Java's GUI features can be effectively utilised to create graphical representations of schedules, batch details, and course information, offering users a user-friendly and aesthetically pleasing interface.

## Java's robust networking support is also helpful for timetable management systems that might need to communicate with other users or components. A central database may need to have data fetched or updated, and Java's networking features make it easier to implement such functionalities.

## Any application handling sensitive data must prioritise security, and Java comes with strong security features. Java's security mechanisms help protect against threats and unauthorised access in a timetable management system, which stores sensitive information about professors, classes, and schedules.

## Additionally, the exception handling feature of Java improves the dependability of the system. In a complicated programme such as a timetable management system, mistakes and outliers are unavoidable. Because of Java's exception handling, developers can detect, manage, and recover from errors in a graceful manner, keeping the system stable and user-friendly.

## In summary, Java's object-oriented design, platform independence, rich library, networking capabilities, security features, and strong exception handling make it indispensable for a time table management system. Together, these characteristics aid in the creation of a dependable, scalable, and user-friendly timetable management system that can meet the various demands of organisations or educational institutions.

## STRUCTURE OF JAVA:

## The Java code provided implements a Timetable Management System using the Swing framework for graphical user interfaces. The structure of the code can be summarized as follows:

## 1. Import Statements:

## The code begins with importing necessary Java packages, including AWT, Swing, file I/O, and image processing.

## 2. Main Class and Entry Point:

## The main class, named `Main`, contains the `main` method as the program's entry point. It initializes maps to store information about batches, courses, and professors.

## 3. Graphical User Interface (GUI) Setup:

## - Multiple frames (`JFrame`) and panels (`JPanel`) are used for organizing different sections of the application.

## - Various Swing components such as labels, buttons, combo boxes, and tabbed panes are utilized for building the GUI.

## 4. Data Structures:

## - Java's `Map` interface (specifically `HashMap`) is used to efficiently store and manage information about batches, courses, and professors.

## 5. Event Handling:

## - Action listeners (`ActionListener`) are implemented to capture user interactions, such as selecting batches or professors, triggering dynamic updates in the displayed timetable.

## 6. File Handling:

## - The application reads an image file (`mrcet.jpeg`) using the `ImageIO` class to display an image on the home panel.

## 7. Timetable Representation:

## - A 2D array (`timet`) represents the timetable structure, and it is dynamically updated based on user selections.

## 8. Dynamic UI Updates:

## - The GUI is designed to dynamically update in response to user actions, such as creating new batches or professors.

## 9. Error Handling:

## - Basic error handling is implemented using conditional statements and dialog boxes to inform the user about invalid inputs or resource limitations.

## In summary, the Java code structures a graphical application for managing timetables, utilizing Swing components, data structures for efficient data management, and event handling for user interaction. It also includes features for dynamic updates and basic error handling to enhance the user experience.

## LAYOUT:

1. **Package Declaration**

The package declaration is used to organize classes into namespaces. It is optional but commonly employed to maintain a structured codebase.

2**. Import Statements**

Import statements allow the usage of classes from other packages. They simplify code by providing access to predefined functionalities.

3. **Class Declaration**

The class declaration is the main structure of a Java program. It includes the class name and may extend a superclass or implement interfaces.

4. **Main Method**

The `main` method serves as the entry point for the program. It has a specific signature (`public static void main(String[] args)`) and is where program execution starts.

5. **Variables and Fields**

Variables are declared with a type, a name, and an optional initial value. They can be defined at the class level or within methods.

6. **Methods**

Methods define behavior and are invoked by other methods or objects. They have a return type, a name, parameters (if any), and a body.

7. **Control Flow Statements**

Control flow statements like `if`, `else`, `for`, `while`, and `switch` determine the execution flow of the program.

8. **Exception Handling**

Java supports exception handling using `try`, `catch`, `finally`, and `throw` keywords. It helps manage errors and unexpected situations.

9. **Comments**

Comments provide explanatory notes within the code. They are ignored by the compiler and are essential for code documentation.

10. **Object-Oriented Concepts**

Java is an object-oriented language, and code is often organized into classes and objects. Concepts like inheritance, encapsulation, and polymorphism are commonly used.

11. **Whitespace and Formatting**

Proper indentation and formatting improve code readability. Consistent use of tabs or spaces is a common practice.

**USE OF JAVA AWT AND SWING:**

Java AWT (Abstract Window Toolkit):

Java AWT is the original Java GUI (Graphical User Interface) library. It provides a set of classes for creating graphical user interfaces and performing various GUI-related operations. Some key features and components of AWT include

1. **Components**:

- AWT includes basic GUI components like buttons, text fields, labels, checkboxes, and more.

- These components are heavyweight, meaning they are dependent on the underlying operating system's windowing toolkit.

2. **Layout Managers**:

- AWT uses layout managers to define the arrangement of components within a container.

- Common layout managers include `FlowLayout`, `BorderLayout`, and `GridLayout`

3. **Event Handling**:

- AWT supports event handling through listeners.

- Event listeners are interfaces that respond to user actions such as button clicks or mouse movements.

4. **Graphics and Drawing**:

- AWT provides the `Graphics` class for drawing shapes, text, and images on components.

- Custom painting is often done by overriding the `paint()` method.

5. **Platform Dependence**:

- AWT components are platform-dependent, and their appearance varies across different operating systems.

- This can lead to inconsistencies in the look and feel of applications.

**Java Swing:**

Swing is a more advanced and lightweight GUI toolkit built on top of AWT. It addresses some limitations of AWT and provides a rich set of components and features. Key aspects of Swing include:

1. **Lightweight Components**:

- Unlike AWT, Swing components are lightweight, which means they are implemented entirely in Java and are not dependent on the underlying operating system.

- This results in a consistent look and feel across different platforms.

2. **Pluggable Look and Feel** (PLAF):

- Swing allows the use of different look and feel themes.

- Developers can choose from a variety of look and feel options, such as the default Metal look, Windows look, or the Nimbus look.

3. **Rich Set of Components:**

- Swing provides an extensive set of components, including advanced components like tables, trees, and tabbed panes.

- These components are highly customizable and offer a more modern appearance.

4. **Layout Managers**:

- Swing uses layout managers similar to AWT, but with additional managers like `BoxLayout` and `GridBagLayout`.

- Layout management is more flexible and powerful in Swing.

5. **Double Buffering**:

- Swing components use double buffering, reducing flickering and improving performance in graphics rendering.

## Testing

Testing is a process of executing a program with the indent of finding an error. Testing is a crucial element of software quality assurance and presents ultimate review of specification, design and coding.System Testing is an important phase. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing. A good test case is one that has a high probability of finding an as undiscovered error. A successful test is one that uncovers an as undiscovered error.

## Testing Objectives:

1. Testing is a process of executing a program with the intent of finding an error
2. A good test case is one that has a probability of finding an as yet undiscovered error
3. A successful test is one that uncovers an undiscovered error Testing Principles:

* All tests should be traceable to end user requirements
* Tests should be planned long before testing begins
* Testing should begin on a small scale and progress towards testing in large
* Exhaustive testing is not possible
* To be most effective testing should be conducted by a independent third party.

The primary objective for test case design is to derive a set of tests that has the highest livelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used.

They are

* + White box testing.
  + Black box testing.

## White-box testing:

White box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

## Block-box testing:

Black box testing is designed to validate functional requirements without regard to the internal workings of a program. Black box testing mainly focuses on the information domain of the software, deriving test cases by partitioning input and output in a manner that provides through test coverage. Incorrect and missing functions, interface errors, errors in data structures, error in functional logic are the errors falling in this category.

## Testing strategies:

A strategy for software testing must accommodate low-level tests that are necessary to verify that all small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements.

## Testing fundamentals:

Testing is a process of executing program with the intent of finding error. A good test case is one that has high probability of finding an undiscovered error. If testing is conducted successfully, it uncovers the errors in the software. Testing cannot show the absence of defects, it can only show that software defects present.

## Testing Information flow:

Information flow for testing flows the pattern. Two class of input provided to test the process. The software configuration includes a software requirements specification, a design specification and source code.

Test configuration includes test plan and test cases and test tools. Tests are conducted and all the results are evaluated. That is test results are compared with expected results. When erroneous data are uncovered, an error is implied and debugging commences.

## Unit Testing:

Unit testing is essential for the verification of the code produced during the coding phase and hence the goal is to test the internal logic of the modules. Using the detailed design description as a guide, important paths are tested to uncover errors with in the boundary of the modules. These tests were carried out during the programming stage itself. All units of Vienna SQL were successfully tested.

## Integration testing :

Integration testing focuses on unit tested modules and build the program structure that is dictated by the design phase.

## System testing:

System testing tests the integration of each module in the system. It also tests to find discrepancies between the system and its original objective, current specification and system documentation. The primary concern is the compatibility of individual modules. Entire system is working properly or not will be tested here, and specified path ODBC connection will correct or not, and giving output or not are tested here these verifications and validations are done by

giving input values to the system and by comparing with expected output. Top-down testing implementing here.

## Acceptance Testing:

This testing is done to verify the readiness of the system for the implementation. Acceptance testing begins when the system is complete. Its purpose is to provide the end user with the confidence that the system is ready for use. It involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements.

## Tools to special importance during acceptance testing include:

Test coverage Analyzer – records the control paths followed for each test case. Timing Analyzer – also called a profiler, reports the time spent in various regions of the code are areas to concentrate on to improve system performance. Coding standards – static analysers and standard checkers are used to inspect code for deviations from standards and guidelines.

## Test Cases:

Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

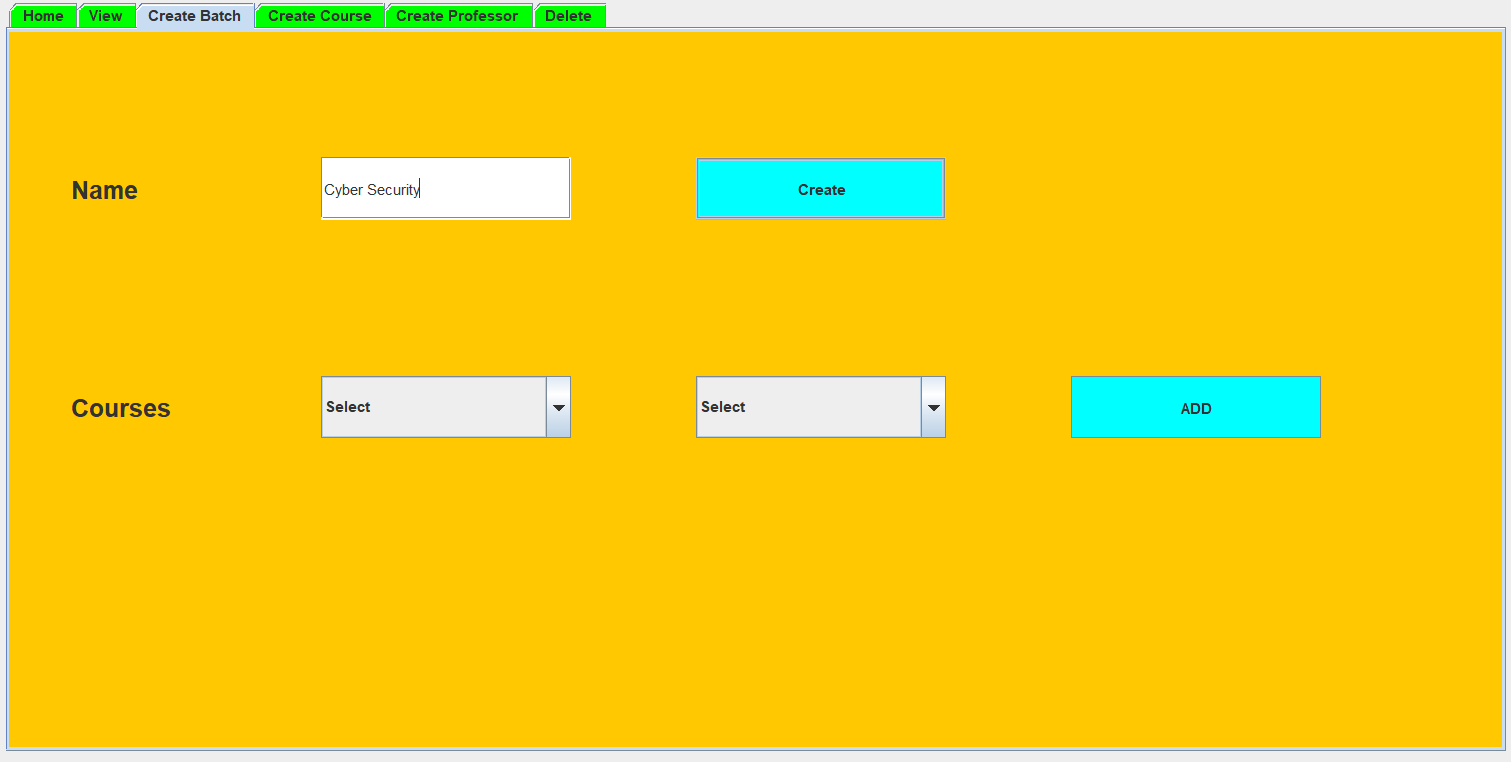
Using White-Box testing methods, the software engineer can drive test cases that

* + Guarantee that logical decisions on their true and false sides
  + Exercise all logical decisions on their true and false sides.
  + Execute all loops at their boundaries and within their operational bounds.
  + Exercise internal data structure to assure their validity. The test case specification for system testing has to be submitted for review before system testing commences.

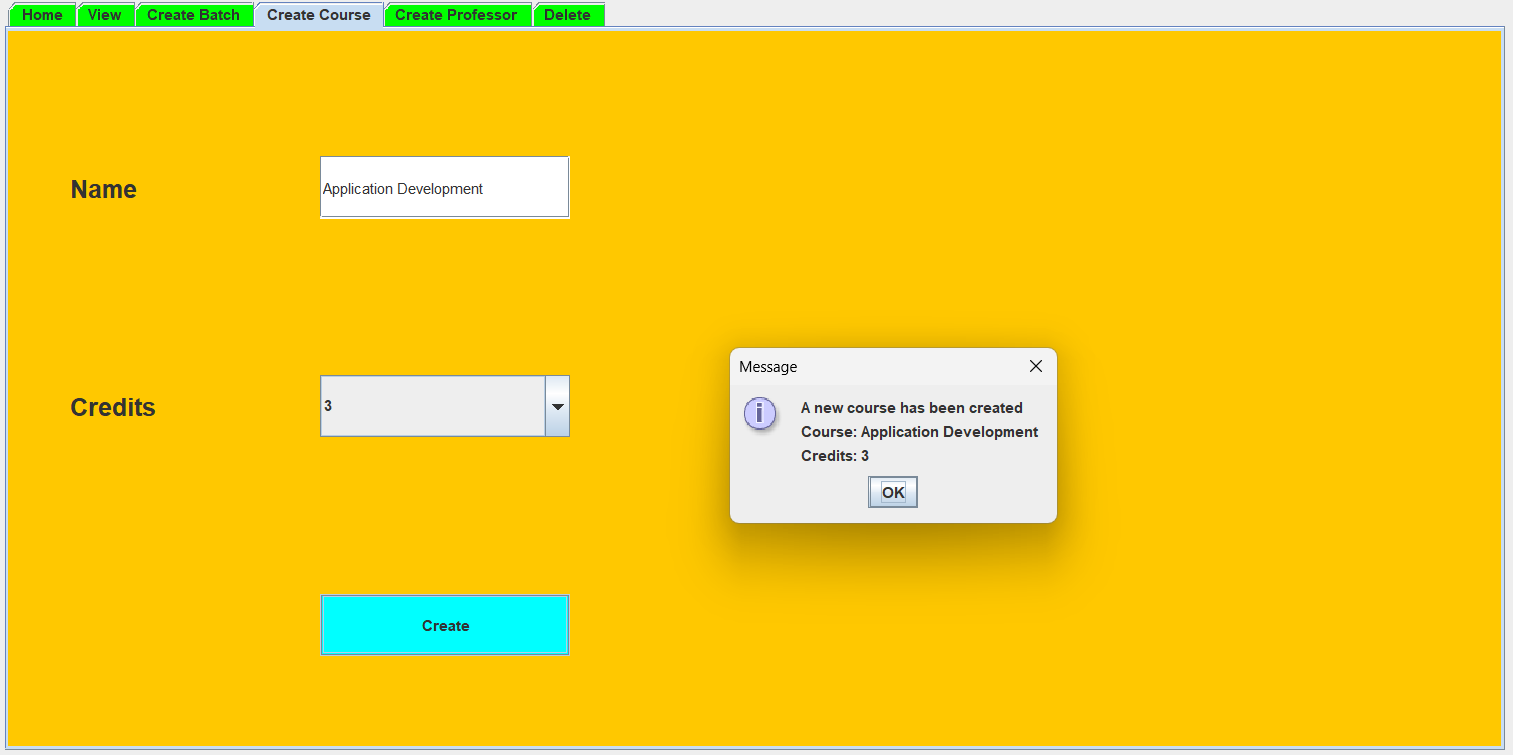
## RESULTS

****

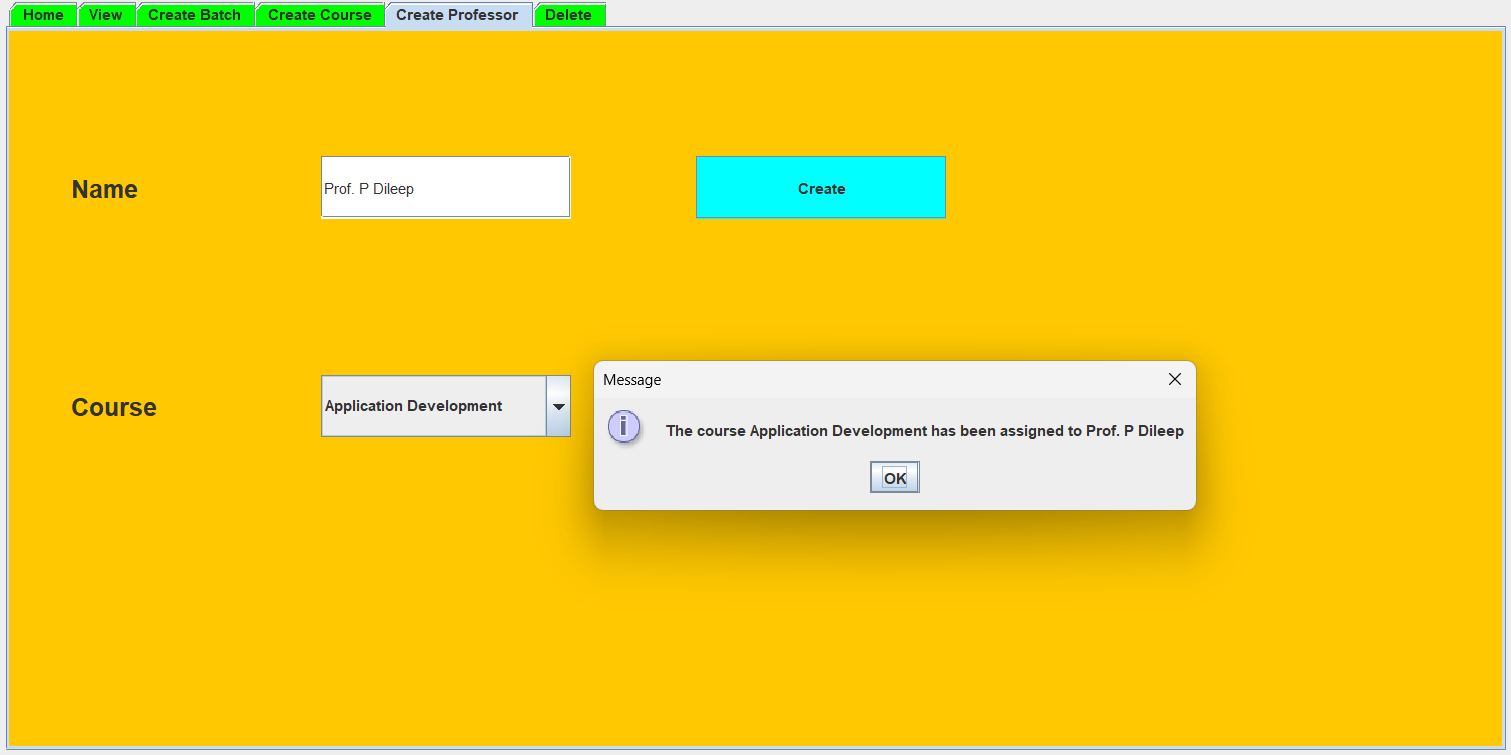
**Figure 1- Dashboard**



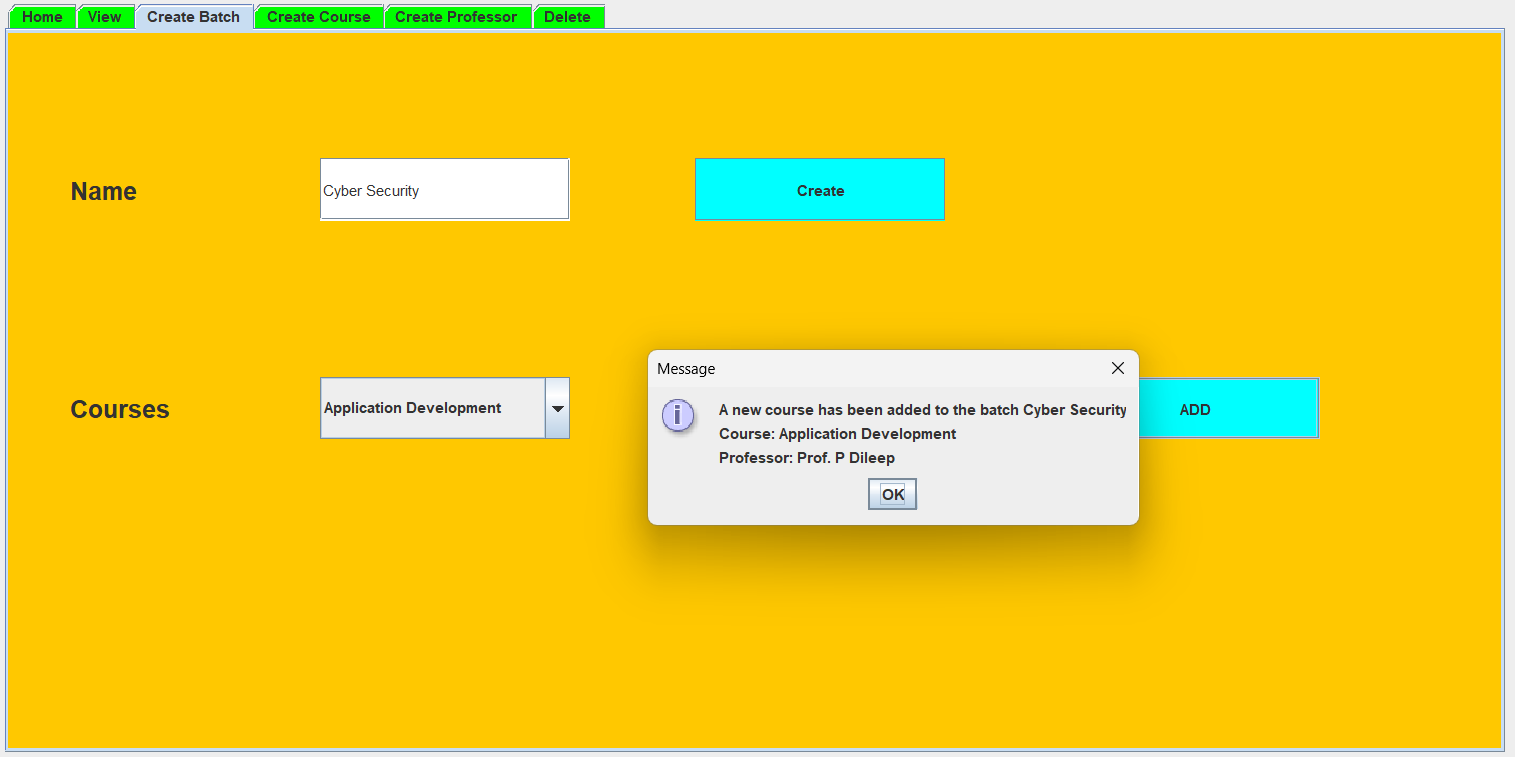
## Figure 2 - Create Batch

****

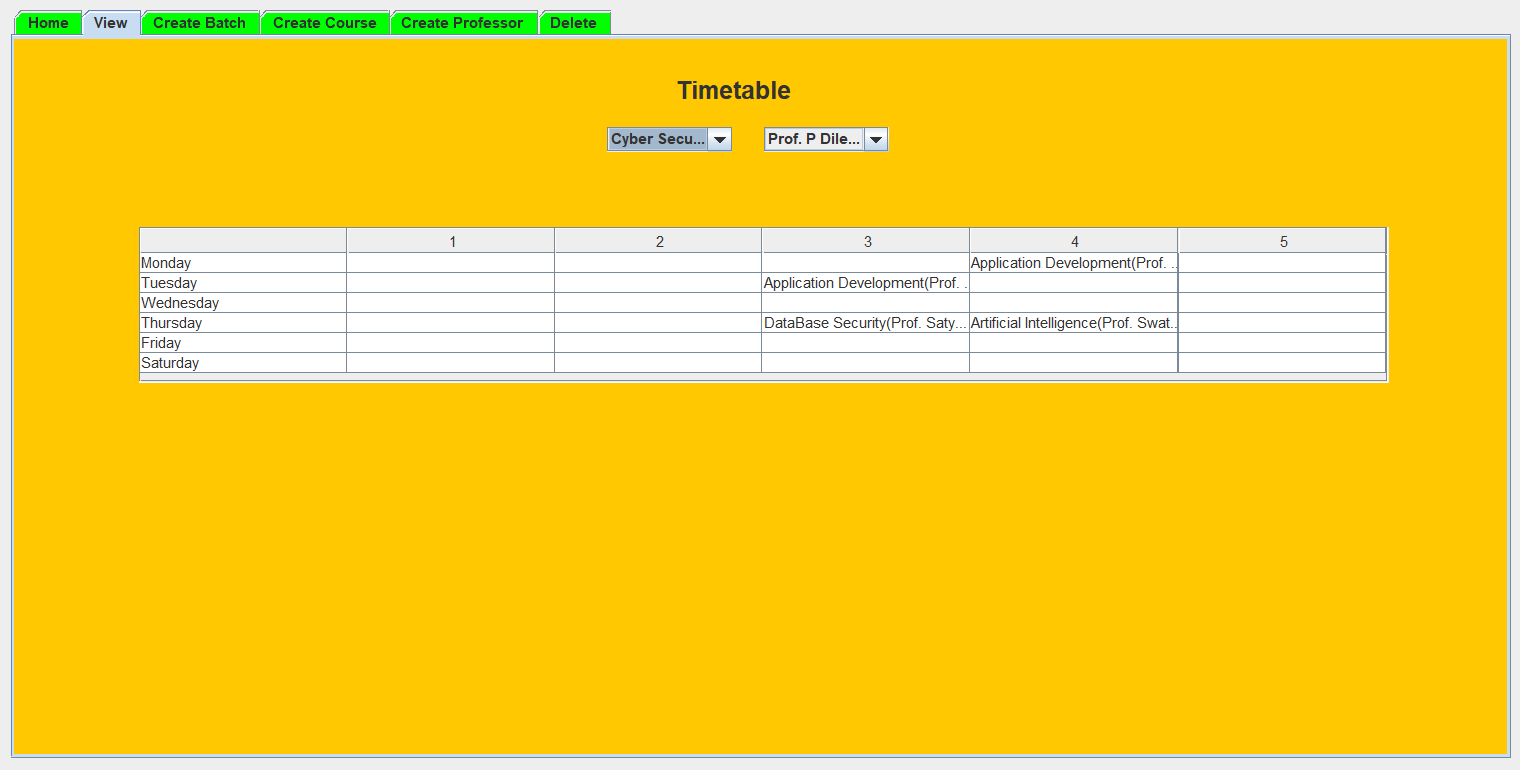
**Figure 3 -Create Course**

****

## Figure 4 -Create Professor



**Figure 5 -Allotment**

****

## Figure 6 -Time Table Generation



**Figure 7 -Delete Batch, Course, Professor**

## CONCLUSION

The documentation for the Automatic Timetable Management System (ATMS) provides an in-depth examination of a system intended to automate and streamline the difficult process of creating timetables in educational establishments.

It discusses the architecture of the system, defines user roles and permissions, and provides specifics on functions like course management, faculty assignments, and schedule creation. The documentation offers information about the database structure, security protocols, scheduling algorithms, user interface design, and system integration with external systems.

In order to ensure the system's dependability, it also describes testing procedures. Finally, it provides details on deployment, upkeep, and the ATMS's overall effect on the effectiveness of school scheduling. This documentation provides a comprehensive overview of the system's design, functionality, and implementation, acting as a guide for all parties involved.

It's complicated task to handle numerous classes and allocating subjects along with the faculty members at a time manually. So our system will help to overcome this disadvantage. Hence, we can produce timetable for any number of courses and multiple academic years. This system will help to produce dynamic pages so that for administering such a system we can make use of the different tools which are broadly applicable and free to use.

**FUTURE SCOPE:**

**The following improvements could be done for the future development of the timetables generating system:**

1. In future we can add lunch breaks and lab sessions for each subject
2. We can make a classroom wise timetable to manage classrooms according to timetable.
3. Also, timetables for each faculty can be made.
4. Making web application.
5. Semantic Understanding.
6. User-Friendly Interfaces.
7. Mobile Accessibility.
8. Collaboration Features.
9. Continuous Improvement

## REFERENCES:

1. A. Abhinaya1, K. Sahithi2, K. Akaanksha3

Online Application of Automatic Time-Table Generator

1. Jaydev Bhatti, Chitkara University ,International Journal of Scientific & Technology Research 8(11):631-635,”Timetable Handling Mechanism Using Python", May 2020
2. Sheng Xiang Yang, a member of IEEE and Sadaf Naseem Jat Mimetic algorithm for university course timetabling problem,

March 2018

1. Shabina Sayed, Ansari Ahmed, Ansari Aamir, and Ansari Zaeem, “Automated Timetable Generator” (2015)

International Journal for Innovative Research in Science & Technology

1. Vittoria Maniezzo, Alberto colorni, Marco Dorigo, A Genetic Algorithm To Solve The Timetable Problem, Centre for Emergent Computing.