# ABSTRACT

## - PROBLEM DEFINITION

We have been using a lot of Online Platforms to watch movies and TV Shows. Out of them, prediction of movies and tv shows that a user wants to watch is one of the prominent features which help users to watch movies of their personal preference.

Movie Predictor is an application designed specifically for the purpose of predicting movies to users according to their tastes, previously watched movies, and their previous ratings given to movies. To achieve this, we analyse the available datasets and perform a set of AI filtering techniques to get the most precise prediction. Their will be lot of users who share similar tastes in watching movies. We use those users’ data to characterise other users as they use the application. This helps users to find the most interesting movies in their genres, which they would like to watch. We have tested this over 100k records of multiple users which was very precise in predicting the movies which a user wants to watch in comparison with users with similar watch pattern.

## - PROJECT AIM

The project aims to provide precise list of movies and TV Shows a user wants to watch with the highest accuracy possible. Also, the AI system will learn according to the users’ current watch patterns and start suggesting movies according to the changes in their patterns such as change of genres, ratings and watch lists. Movie-Predictor will let users to sign-in with their Google Accounts providing easy logging in. Later, the app will ask users to select their preferences on genres, actors, directors, etc which they would like to watch. With the current available data and as a new user, the app will provide a list of movies to the user and from there, user can rate the movies, add movies to their watch bucket. All these actions will be used by our machine learning algorithm to predict the movies and shows for the user next time the machine learning algorithm trains the model. All of the user actions on the movies and shows list will be recorded and send to the backend to store and process later whenever prediction algorithm trains the model again.

Users will be provided with a movie and TV shows forum section where they can discuss about the corresponding movies. Such a way, the behavioural pattern of users will be analysed in relation with how they respond in the forums. Also, they can find more movies to their tastes from the forum discussions. The users usage pattern on the app will be recorded in the backend to give more accuracy to the machine learning algorithm. Such that, if a user rates a movie far away from their usual watch pattern, next time when the model trains, the algorithm will predict movies for that user biased to that newly rated movie as well.

In a lot of ways, the Movie-Predictor app will help users to watch and track their favourite movies and TV shows.

# INTRODUCTION

## - PROJECT INTRODUCTION

Movie-Predictor is a Movie Recommendation System with a Machine Learning Filtration Algorithm to predict the movies and TV Shows according to the rating and preference of a user towards a domain specific movie or genre. So, when the user provides some data about self, the data is processed against a wide variety of Movie Sets where the ratings, genre details are already pre-recorded. For which, we use the largest Movie Dataset from MovieLens to do the prediction.

Almost everyone today uses technology to stream movies and television shows. While figuring out what to stream next can be daunting, recommendations are often made based on a viewer’s history and preferences. This is done through machine learning and AI predictions. This is implemented in Python with data from the MovieLens Dataset. Generated by more than 6,000 users, MovieLens currently includes more than 1 million movie ratings of 3,900 films.

**Advantages**

Mostly the Recommendation system will be predicting movies for the user as per the past preferences of the user. User will be able to select the movies of their preferences, choices and according to that, the app dashboard will prefer more such kind of movies for the user every time the machine learning model is trained again and again according to their behavioural and usage patterns.

But here, we will use a combined filtering algorithm with user-based collaborative filtering and content-based genre filtering where we will compare the user preferences with other users’ behaviour which are already recorded in the data set and is evolving as per their suggestions and usage of the app and the feature and characteristics of the movie and its genre.

# SYSTEM ANALYSIS

## – ANALYSIS OF PROJECT

### – Existing System

Students in higher educational institutions interact with many systems, such as registration and learning management. Mining student data collected from these systems helps discover useful insights with respect to student behavior. Researchers in the field of educational data mining and learning analytics have already demonstrated that such data can be used to predict student academic performance, preferences and engagement .

A survey on the application of data mining methods to achieve different educational purposes, such as the way the input and output of the educational process affect each other was presented. The survey describes the way different research work on determining student failure/success rate in order to help students before they reached risk of failure, and effective resource utilization and cost minimization were also studied. The authors clustered the students based on their class performance and overall attendance (low, medium and high). This helped the authors in predicting the students’ graduation performance in final year at university using only pre- university marks and examination marks of early years at university. The authors of Studied the effect of student class attendance on their academic scores and registered a significant relation between attendance and the academic score. The study used the T-test to measure the relation between the percentage of attendance and the percentage score of the students. A survey on the application of data mining in learning management systems is presented. The survey describes the way each step of the data mining process is applied to the field of elearning, from preprocessing to interpreting results. The survey focused on the open-source Modular Object-Oriented Development Learning Environment (MOODLE) as the source of the educational data.

### – Proposed System

We propose a method that utilizes the attendance records for all students. The collected data is then passed to a data mining technique for analysis. Specifically, the method mines the records for association rules, where each rule links the absences of one student to the absences of one or more students. The method makes use of the market basket analysis [15], which is a data mining technique that retailers, such as Amazon and eBay use to find associations between their products. Retailers generate association rules by inspecting their transactions and finding items that frequently appear together. For example, the following rule may be used by a retailer recommender system.

{*Gaming\_Console*, *Motion\_Detector*} Þ {*Motion\_Game*}

The rule states that if a customer buys a certain gaming console, as well as that console’s motion detector, then the customer will likely to buy a motion-based game, such as a dancing game. The set of the two items, *Gaming\_Console* and *Motion\_Detector*, represents the left side of the association rule, whereas the set of one item, *Motion\_Game*, represents the right side of the rule. Such rules help retailers recommend products to customers. A popular algorithm to generate association rules is the apriori algorithm [16].

In our work, we use the apriori algorithm to find rules associating the absences of one student to the absences of one or more students. The generated rules help instructors and advisors identify students who are frequently absent together. Instructors and advisors may meet with such students to investigate the reason behind this behavior. Such intervention supports student academic success. which illustrates the proposed method. The next subsection defines the necessary terminology for the apriori algorithm.

* 1. **- METHODOLOGY OF THE STUDY**

#### Feasibility study

A feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that is spend on it. Feasibility study lets the developer foresee the future of the project and its usefulness. Feasibility study is attest of system proposed regarding its workability, impact on the organization, ability to meet the needs and effective use of resources

There are four aspects in the feasibility study portion of the preliminary investigation.

* Technical Feasibility
* Economical Feasibility
* Behavioural Feasibility
* Operational Feasibility

The proposed system must be evaluated from a technical point of view first, and if technically feasible their impact on the organization must be assessed. If compatible, the operational system can be devised.

#### Technical feasibility

The system must be evaluated from the technical point of view first. In this we first check the availability of technical resourses. Project uses technologies Python, Android, Mysql, Xamp are free to use. As its open sourse, anyone can simply download these technologies.so the project is technically feasible.

#### Economic feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return the cost it would require. In the development of this project there is no extra cost or efforts are required.in addition there are lots of benefits from this project. So the project is economically feasible.

#### Behavioural feasibility

The present system is easily understandable. The maintenance and working of the new system needs less human effort. All the behavioural aspects are considered carefully and have found the project is behaviourally feasible.

##### • Legal Feasibility study

In this privacy concern issues are considered. However in this project we do not face any legal issues, because personal data like personal id proof is not required or used in this system.

##### • Scheduled Feasibility study

It is defined as the probability of a project to be completed within its scheduled time limits, by a planned due date. In this case the project is completed before the scheduled time.

#### Operational feasibility

This analysis involves how it will work when it is installed and the assessment of political and managerial environment in which it is implemented. In this project if a student is absent we can predict the correctness of reason and check the chances to absent more students. Reduce absent percentage is the aim of the project, we can do it by this project so this is operationally feasible.

## ATTENDANCE PREDICTION

Students in higher educational institutions interact with many systems, such as registration and learning management. Mining student data collected from these systems helps discover useful insights with respect to student behavior. Researchers in the field of educational data mining and learning analytics have already demonstrated that such data can be used to predict student academic performance, preferences and engagement.

A survey on the application of data mining methods to achieve different educational purposes, such as the way the input and output of the educational process affect each other was presented. The survey describes the way different research work on determining student failure/success rate inorder to help students before they reached risk of failure, and effective resource utilization and cost minimization were also studied. The authors clustered the students based on their class performance and overall attendance (low, medium and high). This helped the authors in predicting the students’ graduation performance in final year at university using only pre- university marks and examination marks of early years at university. The authors of [5] Studied the effect of student class attendance on their academic scores and registered a significant relation between attendance and the academic score. The study used the T-test to measure the relation between the percentage of attendance and the percentage score of the students. A survey on the application of data mining in learning management systems is presented in [6]. The survey describes the way each step of the data mining process is applied to the field of elearning, from preprocessing to interpreting results. The survey focused on the opensource Modular Object-Oriented Development Learning Environment (MOODLE) as the source of the educational data.

### 3.5 - MODULES

**TECHINCAL**

* Dataset preparation
* Read Data
* Apply ML
* Match pattern
* Output generation
* Generate output

**HOD**

* Register
* Login
* Register and manage teachers
* Register class
* Assign class
* View attendance
* Post department info
* Check absent reason
* Predict reason

**TEACHER**

* Login
* Register and manage students
* Mark attendance
* Upload study materials

**STUDENT**

* Login
* View attendance status
* Submit absent reason
* View department info
* Download study materials

### Developing Solution Strategies

* The first step is to collect the absence records for each student in each class. A list of student ids needs to be extracted, *Stu\_IDs*. Similarly, a list of classes/sections needs to be extracted, *Sec\_IDs*, such as CS101\_3.
* For each section, extract the dates where that section had a session and the instructor of that section recorded the attendance. Append the *Sec\_ID* to each date, creating a list of dates for each section, *Sec\_ID\_Date\_IDs*. For example, if section *CS101\_3* had a session on Apr 12th, then *CS101\_Apr12* is created. This is repeated for each section to create a list of lists, *Master\_Sec\_Date\_List*. Example element of *Master\_Sec\_Date\_List* is *CS101\_3\_Apr12*, where *CS101\_3* is the section id and *Apr12* is the date of one of the sessions. Each element of *Master\_Sec\_Date\_List* is therefore a transaction label.
* Create a matrix where the rows represent the elements of *Master\_Sec\_Date\_List* ids and the columns represent the student ids. The celli,j in created matrix represents whether student, *j*, was present (1) or absent (0) during session *i* (note that *i* is composed of a section id and a date id). Table II is an illustration of the transaction matrix that we supply to the apriori algorithm. The second row shows that student1 was absent along with student4 on Jan 7th in section CS101\_1.
* Fill the matrix created in Step 3 with student absence values based on the records pulled from the attendance system.
* Run the apriori algorithm on the computed matrix from Step 4. The support and confidence thresholds need to be specified. The support threshold is based on the number of sessions. If a student takes 5 courses in a semester, where each section holds 30 sessions throughout the semester, then the maximum number of sessions is 150. Consider two students having an exact class schedule, the maximum number of times both are absent is 150. However, it is highly unlikely that a student misses all of sessions, in all of the classes. Therefore, one may set the support threshold a reasonable value. For example, a threshold of 10 help generate association rules, where each rule is supported by 10 incidents were the two students were absent together. The confidence threshold is based on the required rule strength. Setting the threshold to 0.7 help generate rules, where each rule associates the absence of one student to another with 70% accuracy. Setting the support and confidence thresholds also depends on the number of rules the user is willing to go through. The lower the thresholds, the more rules the apriori generates.
* The output of the apriori algorithm is a set of rules that meets the thresholds set in Step 5. If a student appears in a rule, then the advisor of that student may receive a notification with regards to that rule. An example output is displayed in Table III. Table III lists a set of the actual anonymized rules generated based on the attendance records collected from the author institution. The experiment detail is explained the next section.
* The advisor needs to investigate the rules. Note that some rules may be the result of students coincidentally missing classes, without peer pressure. The advisor may check on the student overall performance to set the priority for meeting students. Students who miss classes and are under probation needs urgent attention, compared to students with good standing status.
* Finally, advisors provide feedback for the person responsible for generating the rules with regards to the usefulness of the generated rules. The feedback helps tweak the support and confidence thresholds for the next rounds.

### 3.6 - TIMELINE CHART

|  |  |  |
| --- | --- | --- |
| **System Analysis** | **Starting Date** | **Ending Date** |
| System Study | 20-04-2021 | 30-04-2021 |
| Requirement Analysis | 02-05-2021 | 09-05-2021 |

|  |  |  |
| --- | --- | --- |
| **Database Design** | **Starting Date** | **Ending Date** |
| Physical database Design | 10-05-2021 | 05-05-2021 |
| DFD | 18-05-2021 | 20-05-2021 |
| Table Design | 30-05-2021 | 13-06-2021 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Design Phase** | **Starting Date** | | **Ending Date** | |
| User interface Design | 16-06-2021 | | 30-06-2021 | |
| Coding | 02-07-2021 | | 22-07-2021 | |
| **Testing and**  **Implementation** | | **Starting Date** | | **Ending Date** |
| Unit Testing | | 25-07-2021 | | 30-07-2021 |
| Module Testing | | 01-08-2021 | | 08-08-2021 |
| Integration Testing | | 09-08-2021 | | 14-08-2021 |
| Implementation | | 16-08-2021 | | 24-08-2021 |

### 3.7 - COST ESTIMATION

|  |  |
| --- | --- |
| **PURPOSE** | **COST** |
| Cost of the home page with basic design | **3000** |
| Cost for other pages | **1000** |
| Total Pages | **22** |
| Total Cost | **25000/-** |

# 4.SYSTEM

**REQUIREMENTS**

## 4.1 – SOFTWARE AND HARDWARE REQUIREMENTS

### HARDWARE REQUIREMENTS

The selection of hardware is very important in the existence and proper working of any of the software. When selecting hardware, the size and capacity requirements are also important. The hardware must suit all application developments**.**

* Processor : i3 or above.
* System Bus : 32Bit or 64Bit
* RAM : 4 GB or Above
* HDD : 500 GB or Above
* Monitor : 14” LCD or Above
* Key Board : 108 Keys
* Mouse : Any Type of mouse
* Mobile : Android supported mobile phone

### SOFTWARE SPECIFICATION

One of the most difficult tasks is selecting software, once the system requirement is

find out then we have to determine whether a particular software package fits for those system requirements. This section summarizes the application requirement.

* Operating System : Windows 10 Any 32 bit or 64 bit platform
* Front End : Android, .Python
* Back End : MySQL Sever
* IDE : Eclipse or Android studio

: Python 3.6 or above

: PyCharm

* Browser : Microsoft Edge, Google Chrome

**4.2 - OPERATING SYSTEM**:

### WINDOWS 10

Windows 10 is a series of personal computer operating systems produced by Microsoft as part of its Windows NT family of operating systems. It is the successor to Windows 8.1, and was released to manufacturing on July 15, 2015, and broadly released for retail sale on July 29, 2015. Windows 10 receives new builds on an ongoing basis, which are available at no additional cost to users, in addition to additional test builds of Windows 10 which are available to Windows Insiders. The latest stable build of Windows 10 is Version 1903 (May 2019 Update). Devices in enterprise environments can receive these updates at a slower pace, or use long-term support milestones that only receive critical updates, such as security patches, over their ten-year lifespan of extended support.

## 4.3 TECHNOLOGY

### PYTHON

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

### ANDROID

Android is a [mobile operating system b](https://en.wikipedia.org/wiki/Mobile_operating_system)ased on a modified version of the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel) and other [open source s](https://en.wikipedia.org/wiki/Open-source_software)oftware, designed primarily for [touchscreen m](https://en.wikipedia.org/wiki/Touchscreen)obile devices such as [smartphones a](https://en.wikipedia.org/wiki/Smartphone)nd [tablets.](https://en.wikipedia.org/wiki/Tablet_computer) Android is developed by a consortium of developers known as the [Open Handset Alliance a](https://en.wikipedia.org/wiki/Open_Handset_Alliance)nd commercially sponsored by [Google.](https://en.wikipedia.org/wiki/Google) It was unveiled in November 2007, with the [first commercial Android device l](https://en.wikipedia.org/wiki/HTC_Dream)aunched in September 2008.

It is [free and open source s](https://en.wikipedia.org/wiki/Free_and_open_source)oftware; its source code is known as Android Open Source

Project (AOSP), which is primarily licensed under the [Apache License.](https://en.wikipedia.org/wiki/Apache_License) However most Android devices ship with additional [proprietary software p](https://en.wikipedia.org/wiki/Proprietary_software)re-installed, most notably [Google Mobile Services (](https://en.wikipedia.org/wiki/Google_Mobile_Services)GMS) which includes core apps such as [Google Chrome,](https://en.wikipedia.org/wiki/Google_Chrome) the [digital distribution p](https://en.wikipedia.org/wiki/Digital_distribution)latform [Google Play a](https://en.wikipedia.org/wiki/Google_Play)nd associated [Google Play Services](https://en.wikipedia.org/wiki/Google_Play_Services) development platform.

### BACK END

#### MySQL Server

It is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups. MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses.

### PYCHARM IDE

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains (formerly known as IntelliJ). It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda. PyCharm is crossplatform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License, and there is also Professional Edition with extra features – released under a proprietary license.

### ECLIPSE IDE

Eclipse is an [integrated development environment (](https://en.wikipedia.org/wiki/Integrated_development_environment)IDE) used in [computer programming.](https://en.wikipedia.org/wiki/Computer_programming) It contains a base [workspace a](https://en.wikipedia.org/wiki/Workspace)nd an extensible [plug-in s](https://en.wikipedia.org/wiki/Plug-in_(computing))ystem for customizing the environment. Eclipse is written mostly in [Java a](https://en.wikipedia.org/wiki/Java_(programming_language))nd its primary use is for developing Java applications, but it may also be used to develop applications in other [programming languages v](https://en.wikipedia.org/wiki/Programming_language)ia plug-ins, including [Ada](https://en.wikipedia.org/wiki/Ada_(programming_language))[, ABAP](https://en.wikipedia.org/wiki/ABAP)[, C](https://en.wikipedia.org/wiki/C_(programming_language))[, C++](https://en.wikipedia.org/wiki/C%2B%2B)[, C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language))[, Clojure](https://en.wikipedia.org/wiki/Clojure)[, COBOL](https://en.wikipedia.org/wiki/COBOL)[, D](https://en.wikipedia.org/wiki/D_(programming_language))[, Erlang](https://en.wikipedia.org/wiki/Erlang_(programming_language))[, Fortran](https://en.wikipedia.org/wiki/Fortran)[, Groovy,](https://en.wikipedia.org/wiki/Groovy_(programming_language)) [Haskell](https://en.wikipedia.org/wiki/Haskell_(programming_language))[, JavaScript](https://en.wikipedia.org/wiki/JavaScript)[, Juli](https://en.wikipedia.org/wiki/Julia_(programming_language))[a,[7](https://en.wikipedia.org/wiki/Eclipse_(software)#cite_note-7)[] Lasso](https://en.wikipedia.org/wiki/Lasso_(programming_language))[, Lua](https://en.wikipedia.org/wiki/Lua_(programming_language))[, NATURAL](https://en.wikipedia.org/wiki/Software_AG)[, Perl](https://en.wikipedia.org/wiki/Perl)[, PHP](https://en.wikipedia.org/wiki/PHP)[, Prolog](https://en.wikipedia.org/wiki/Prolog)[, Python,](https://en.wikipedia.org/wiki/Python_(programming_language)) [R](https://en.wikipedia.org/wiki/R_(programming_language))[, Ruby (](https://en.wikipedia.org/wiki/Ruby_(programming_language))including [Ruby on Rails f](https://en.wikipedia.org/wiki/Ruby_on_Rails)ramework), [Rust](https://en.wikipedia.org/wiki/Rust_(programming_language))[, Scala,](https://en.wikipedia.org/wiki/Scala_(programming_language)) and [Scheme.](https://en.wikipedia.org/wiki/Scheme_(programming_language)) It can also be used to develop documents with [LaTeX (](https://en.wikipedia.org/wiki/LaTeX)via a TeXlipse plug-in) and packages for the software [Mathematica.](https://en.wikipedia.org/wiki/Mathematica) Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

### SUBLIME TEXT

Sublime Text Editor is a full featured Text editor for editing local files or a code base. It includes various features for editing code base which helps developers to keep track of changes. Various features that are supported by Sublime are Syntax Highlight, Auto Indentation, File Type Recognition, Sidebar with files of mentioned directory, Macros, Plug-in and Packages

Sublime Text editor is used as an Integrated Development Editor (IDE) like Visual Studio code and NetBeans. The current version of Sublime Text editor is 3.0 and is compatible with various operating systems like Windows, Linux and MacOS.

# 5. DESIGN AND DEVELOPMENT

## 5.1 - SYSTEM DEFINITION

## 5.2 - Structural Chart

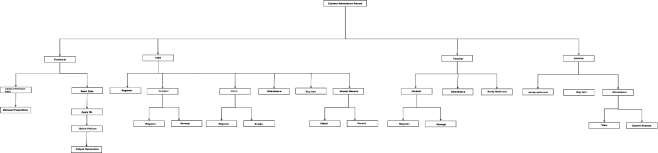
As well as a DFD, it is also useful to develop a structural system mode. This structural model shows how a function is realized by a number of other functions, which it calls. Structure charts are a graphical way to represent this decomposition hierarchy. Like DFD, they are dynamic rather than static system models. They show how one function calls others. They do not show a static bock structure of a function or procedure.

A function is represented on a structure chart as a rectangle. The hierarchy is displayed by linking rectangles with lines. Input and outputs are indicated with annotated arrows. An arrow entering a box implies input, a box implies output. Data stores are shown as rounded rectangles and user inputs as circles.

### Rules to be applied

Many systems can be considered as three stages, input validation and output. If data validation is require, function to implement these should be subordinate to an input function. The role of function near the top of the structural hierarchy may be to control and coordinate a set of lower level hierarchy. The objective of design process is to have loosely coupled highly cohesive components. Each node in the structure chart should have between two and seven subordinates.

**STRUCTURAL CHART**



## 5.3 - ENTITY RELATIONSHIP DIAGRAMS (ERDs)

The E-R model was introduced by P.P Chen. Entity –Relationship modelling is a detailed, logical representation of the entities, associations and data elements for an organization or business area. This technique is used in database design, that helps to describe how entities in an enterprise are related to one another. The entity relationship model for data uses three features to describe data. These are the following

### Entities

An entity is a person, place, thing or event of interest to the organization and about which data are captured, stored or processed .For example, an Employee is an entity.

### Attributes

Various type of data items that describe an entity are known as attributes. For example Name, address, DOB (Date Of Birth) etc are attributes of the entity Employee.

### Relationship

An association of several entities in an Entity-Relation model is called relationship.

#### Name Symbol Meaning

Rectangle  Represents entity set

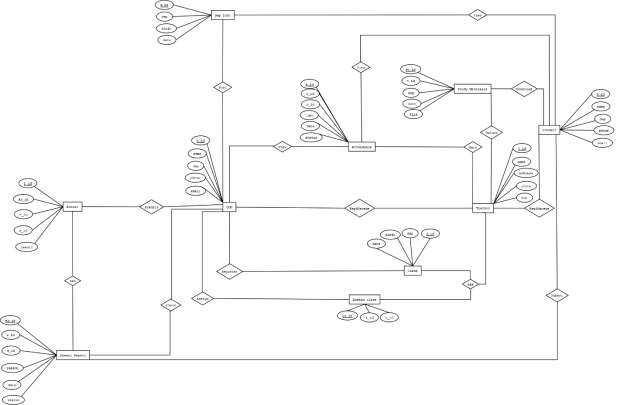
Oval Represents attributes

Diamond Represents relationships among entity set.

Line Links attributes to entity set and entity set to Relationships.

While drawing the entity relationship (E-R) diagram, entity names are represented by a rectangle, relationships are represented by a diamond and oval shapes are used for representing attributes. Three types of relationships exist among entities. These are One to one (1:1) one to many (1: M) many to many (M: M) . One to one relationship is an association only between two entities. One to many relationship exist when one entity is related to more than one entity.

## ER-DIAGRAM



### 5.4 - DATAFLOW DIAGRAMS

Data Flow Diagram is a way of expressing system requirements in a graphical form. It has the purpose of identifying major transformation that will become programs in system design.

A Data Flow Diagram (DFD) or a Bubble chart is a graphical tool for structured analysis. DFD models a system by using external entities from which data flow to process, which transforms the data and creates, output-data-flows which go to other processes or external entities or files. There are various symbol used in a DFD. Bubbles represent the processes; Named arrows indicate the data flow. External entities are representing by rectangles and are outside the system such as vendors or customers with whom the system interacts. That either supply or consume data. Entities supplying data are known as source and those that consume data are called sinks. Generally, DFD‟s are used as a design notation to represent architectural design (External design) and top level design (Internal design) specification. DFD‟s represent the system in hierarchical manner with one top level and many lower level diagrams with each representing separate parts of the system. A DFD shows what kind of information about will be input to and output from the system, where the data will come from and go to, and where the data will be stored.

**To construct a data flow diagram the following symbols are used:**

**Arrow**

**Circle**

**Open End Box**

**Squares**

Five rules for constructing a data flow diagram

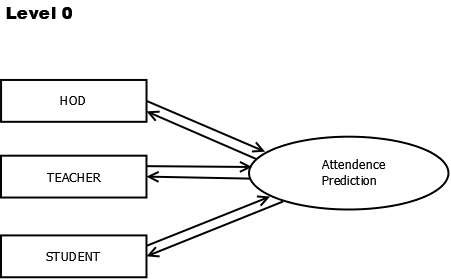
1.Arrows should not cross each other.

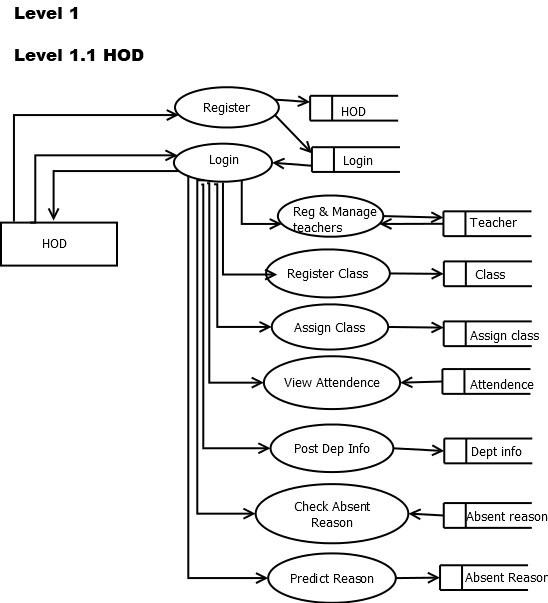
2.Squares, circles and files must bear names.

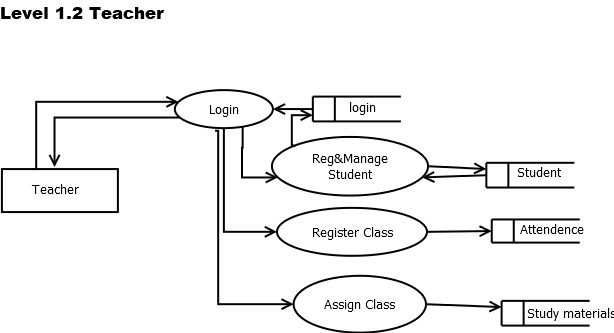
3.Decomposed data flow squares and circles can have same names.

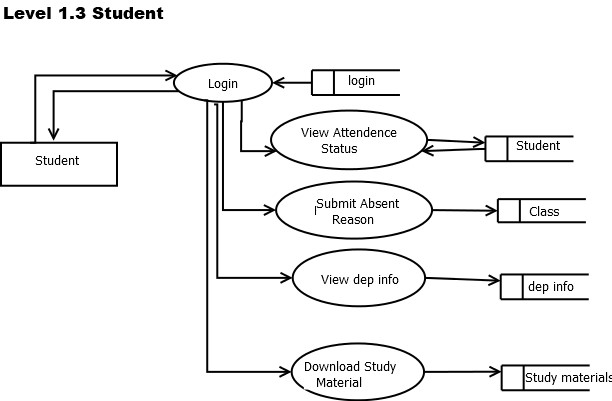
4.Choose meaningful names for data flow.

5.Draw all data flows around the outside of the diagram









### 5.5 - SYSTEM DESIGN AND DEVELOPMENT

Design is the step in the development phase for every engineered product or system. Computer Software designing techniques like engineering design approach in the other, changes continuously as new method, better analysis and broader understanding evolve.

System design involves translating information requirement and conceptual design into technical specification and general flow of processing. After the user requirements are identified, related information is gathered to verify the problem and after evaluating the existing system, a new system is proposed. The proposed system consists of various table, their maintenance and report generation.

#### SYSTEM DESIGN

##### Architectural Design

Architecture is an overall design of the system. Architecture takes into consideration the overall working of the system. Large system can be decomposed into sub-systems that provide some related set of services. The initial design process of identifying this sub-system and establishing a framework for sub-system control and communication is called architecture design. Architecture design usually comes before detailed system specification. Architecture decomposition is necessary to structure and organize the specification. There is no generally accepted process depends on application knowledge and on the skill and intuition of the system architect.

### 5.6 - INPUT AND OUTPUT DESIGN

**HOD:**

The primary role of HOD is register and manage all the teachers and assigning class for them. HOD also register class and view attendance. In addition HOD post department info and check absent reason

**Teacher:**

Teacher can register and manage students . Teachers mark the attendance of student in each hour. Teachers can upload study materials of her subjects.

**Student:**

Student can view the attendance status. If they are absent, the reason can be submitted in the application. Students can view the department info posted by HOD.Students can download the study materials which is uploaded by the teacher for each subjects.

## Input Design

Input design indicates the conversation of the user-originated inputs into the computer represent able form. The first step in the system design is to design input within predefine guidelines. In input design, User oriented data are converted to a computer based format. Input design is the link that ties the information system into the worlds of its users.

The goal of input design is to make data entry as easy, logical and free from errors as possible. Input data are collected and organized into groups of similar data, appropriate input media we selected for processing. It consists of developing specifications and procedures for entering data into a system and must be in a simple format. A form can be used to enter these details using „VB‟ tools such as command boxes, text boxes etc.

## Output Design

Once the output requirements are determined, the system designer can decide what to include in the system and how to structure it so that the required output can be produced designing computer output should proceed in an organized, well throughout manner; the right output element is designed so that the people will find the system executed. The usefulness of the system is evaluated on the basis of their output.

### 5.7 - Logical Design

In the logical design, represents the dataflow diagram of the proposed system. A data flow diagram is a graphical representation that depicts information flow and transforms that are applied as data to move from input to output.

A dataflow diagram may be used to represent a system or software at any level of abstraction. DFD‟s can be partitioned into levels that represent increasing information flow and functional details.

A level 0 DFD, also called fundamental system model or a context model, represents the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows, respectively. Each of the process represented at level 1 is a sub function of the overall system depicted in the context model.

### 5.8 - Database Design

The objective in the development of the database technology has been to treat data as an organizational resource and make information access easy, inexpensive and flexible for the user in the database design. A databases an integrated collection of data and provides centralized access to data. The organization of data in the database aims to achieve two major activities. They are data integrity and data independence. The organization of data in the database aims to achieve the following objectives

* Controlled redundancy
* Ease of learning in use
* Data independence
* More information in low cost
* Accuracy and integrity
* Recovery from failures
* Privacy and security
* Performance

#### 5.9 - TABLES

1. HOD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | datatype | Size | Constraints | Description |
| h\_id | int | 11 | Primary key | HOD id |
| Name | Varchar | 30 | Not null | HOD Name |
| Address | Varchar | 30 | Not null | HOD address |
| Dept | Varchar | 30 | Not null | HOD department |
| Phone | Varchar | 30 | Not null | HOD phone number |
| Email | Varchar | 30 | Not null | HOD email |
| Password | Varchar | 30 | Not null | HOD password |

1. LOGIN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | datatype | Size | constraints | Description |
| L\_id | int | 11 | Primary key | Login id |
| Type | Varchar | 30 | Not null | Type User |
| u\_id | Varchar | 30 | Foreign key | User id |
| username | Varchar | 30 | Not null | Username |
| password | Varchar | 30 | Not null | password |

1. Teacher

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | datatype | Size | constraints | Description |
| t\_id | int | 11 | Primary key | Teacher id |
| h\_id | int | 11 | Foreign key | HOD id |
| Name | Varchar | 30 | Not null | Teacher Name |
| Address | Varchar | 30 | Not null | Teacher address |
| Dept | Varchar | 30 | Not null | Teacher department |
| Phone | Varchar | 30 | Not null | Teacher phone number |
| Email | Varchar | 30 | Not null | Teacher email |
| password | Varchar | 30 | Not null | Teacher password |

1. Class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | datatype | Size | constraints | Description |
| C\_id | int | 11 | Primary key | Class id |
| t\_id | int | 11 | foriegn key | Teacher id |
| Dept | Varchar | 30 | Not null | Department |
| Type | Varchar | 30 | Not null | Type |
| Course | Varchar | 30 | Not null | Course |
| Sem | Varchar | 30 | Not null | Semester |

1. Assign class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | datatype | Size | constraints | Description |
| As\_id | int | 11 | Primary key | Assigning class id |
| C\_id | int | 11 | foriegn key | Class id |
| t\_id | int | 11 | forirgn key | Teacher id |
| Date | date | 10 | Not null | Date |

1. Attendance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | datatype | Size | constraints | Description |
| A\_id | int | 11 | Primary key | Attendence id |
| S\_id | int | 11 | foriegn key | Student id |
| T\_id | int | 11 | Foreign key | Teacher id |
| Date | date | 11 | Not null | Date |
| attendence | Varchar | 30 | Not null | Attendance |
| H1 | Varchar | 30 | Not null | 1st Hour |
| H2 | Varchar | 30 | Not null | 2nd Hour |
| H3 | Varchar | 30 | Not null | 3rd Hour |
| H4 | Varchar | 30 | Not null | 4th Hour |
| H5 | Varchar | 30 | Not null | 5th Hour |
| Status | Varchar | 30 | Not null | Status |

1. Dept info

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | datatype | Size | constraints | Description |
| D\_id | int | 11 | Primary key | Department id |
| H\_id | int | 11 | Foreign key | HOD id |
| Name | Varchar | 30 | Not null | Dept Name |
| Info | Varchar | 30 | Not null | Department information |
| Date | date | 11 | Not null | Date |

1. Absent Reason

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | datatype | Size | constraints | Description |
| Ab\_id | int | 11 | Primary key | Absent id |
| S\_id | int | 11 | foriegn key | Student id |
| A\_id | int | 11 | Not null | Attendence id |
| Reason | Varchar | 30 | Not null | Absent reason |
| date | date | 11 | Not null | Date |

1. Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | datatype | Size | constraints | Description |
| s\_id | int | 11 | Primary key | Student id |
| Name | Varchar | 30 | Not null | Student name |
| c-id | int | 11 | foriegn key | Class id |
| dept | Varchar | 30 | Not null | Department |
| phone | Varchar | 30 | Not null | Phone number |

1. Study Materials

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | datatype | Size | constraints | Description |
| Sm\_id | int | 11 | Primary key | Study material id |
| T\_id | int | 11 | Foreign key | Teacher id |
| sem | Varchar | 30 | Not null | Semester |
| file | Varchar | 30 | Not null | File |
| date | date | 11 | Not null | Date |

**6. TESTING**

# 6. TESTING

## 6.1 - SYSTEM TESTING

### System Testing

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer based system .Although each test has a different purpose, all work to verify that all system elements have been properly integrated and perform allocated functions.

#### Testing Methodologie

The following are testing methodologies used here:

* Unit testing
* Module testing
* Integration testing

#### 6.2 - UNIT TESTING

Unit testing enables a programmer to detect error in coding. A unit test focuses verification of the smallest unit of software design. In this project each form’s code is checked uniquely, to check the functioning of both input and output forms.in each form a input is inserted then check if its stored in correct location and original output is produced.

#### 6.3 - MODULE TESTING

Instead of testing whole software program at once, module testing recommends testing the smaller building blocks of the program. In this project we can perform module testing as, when HOD posts an information check whether the student can view the information or not. Multiple process are connected here, so each modules are checked to ensure the other modules are correct.

#### 6.4 - INTEGRATION TESTING

In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units. All modules are combined and run as whole project. It checks the project works properly.

# 7. IMPLEMENTATION

Implementation is the stage of the project where the theoretical design is turned into a working system. At this Stage the main work load, the greatest upheaval and the major impaction the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can cause chaos and confusion.

Implementation includes all those activities that take place to convert from the old system to new one. The new system may be totally new, replacing an existing manual or automated system or it may be a major modification to an existing system. Proper implementation is essential to provide a reliable system o meet the organization requirements. Successful implementation may not guarantee improvement in the organization using the new system, but improper installation will prevent it.

The process of putting the developed system in actual system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specification. The system personnel check the feasibility of the system.

The most crucial stage is achieving a new successful system and giving confidence on the new system for the user that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover.

# 8. SYSTEM MAINTENANCE

The maintenance phase of the software cycle is the time in which a software product perform useful work.

After a system successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for the system maintenance is for it to make adaptable to the changes in the system environment. There may be social, technical and other environmental changes, which affect a system, which is being implemented software product enhancement may involve providing new functional capabilities, improving user displays and mode of iteration, upgrading the performance characteristics of the system. So only through proper system maintenance procedures, the system may adapt to cope up with these changes.

Software maintenance is of course, far more than “finding mistakes “we may define maintenance by describing four activities that are undertaken after a program is released for use. The first maintenance activity occurs because it is unreasonable to assume that software testing will uncover all latent errors in a large software system. During the use of any large program, errors will occur and be reported to the developer

The second activity that contributes to a definition of maintenance occurs because of the rapid change that is encountered in every aspect of computing. Therefore, adaptive maintenance an activity that modifies software to properly interface with changing environment is both necessary and commonplace.

The third activity that may be applied to a definition of maintenance occurs when a software package is successful. As the software is used, recommendation for new capabilities, modification to existing function, and general enhancement are received from users. To satisfy request in this category, perfective maintenance is performed. This activity for the majority of all effort expanded on software maintenance.

The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability or to provide a better basis for future enhacement. Often called preventive maintenance, this activity is characterized by reverse engineering and reengineering technique.

# 9. FUTURE ENHANCEMENT

The application developed is designed in such a way that any further enhancement can be done with ease. The system has the capability for easy integration with other system. New modules can be added to the existing systsm with less effort

* Facility for modifying user detail.
* More interactive user interface.
* Facilities for Backup creation.
* Can be done as Web page.
* Can be done as Mobile Application.

# 10. CONCLUSION

Student attendance systems, such as the one at the authors’ institution, keep absence records of each student for each of his or her classes in each semester. The records are used for enforcing attendance policies only to ensure students absences do not exceed a certain threshold. Mining student attendance records can be useful for identifying student who miss classes due to peer pressure. We investigated the value of analyzing the records pulled from the student attendance system. We presented a method that applies a market basket analysis, using the apriori algorithm, on student attendance data. The method was used to mine over 50,000 attendance records that belong to over 2000 students at the author institution. The method was successful in generating more than a 100 association rules. Each rule links the absences of one student to the absences of one or more students.

The main limitation of this work is the fact that the method needs a significant number of records to be able to generate the rules. At least, half of the semester may pass before enough absences can be recorded by class instructors. Nevertheless, advisors may benefit from this approach to better advise students for the next semester.

# 11. BIBLIOGRAPHY

Yuichiro Mori, Hideharu Kojima, Eitaro Kohno, Shinji INOUE, Tomoyuki OHTA, and Yoshiaki KAKUDA- A self- configurable new generation children tracking system based on mobile Ad Hoc networks consisting of Android mobile terminals. Reshma.M & Amruta K.M.- Survey on different technologies of child tracking system, Saranya, J.Selvakumar- Implementation of children tracking system on Android Mobile terminal.

ONLINE REFERENCES

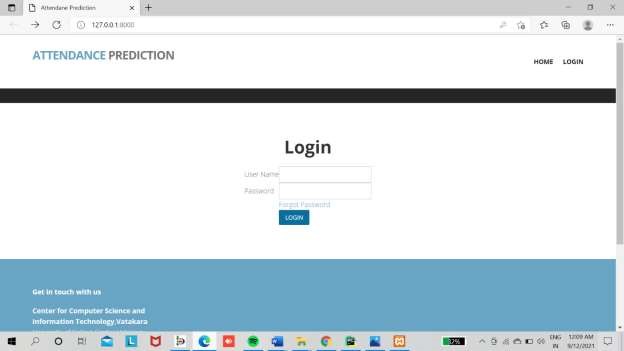
1. [**http://www.w3schools.com/sql/default.asp**](http://www.w3schools.com/sql/default.asp)
2. [**http://www.dotnetspider.com/tutorials/visualstudiotutorials.asp**](http://www.dotnetspider.com/tutorials/visualstudiotutorials.asp)
3. **http://www.wikipedia.com**
4. [**https://www.javatpoint.com/html.tutorial**](https://www.javatpoint.com/html.tutorial)
5. **https://www.w3schools.com/js/**

# 12.APPENDIX

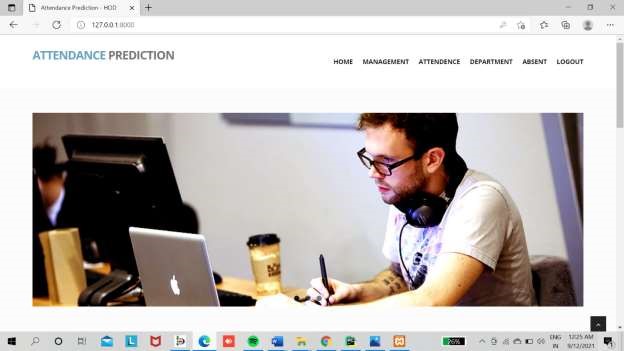
## 12.1 - FINAL RESULT

**Screenshot**

### Login Page



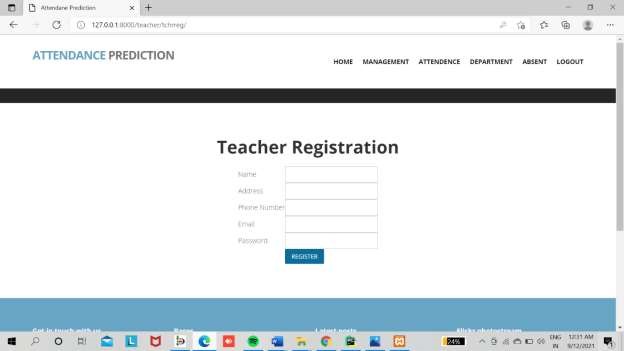
### HOD Home page



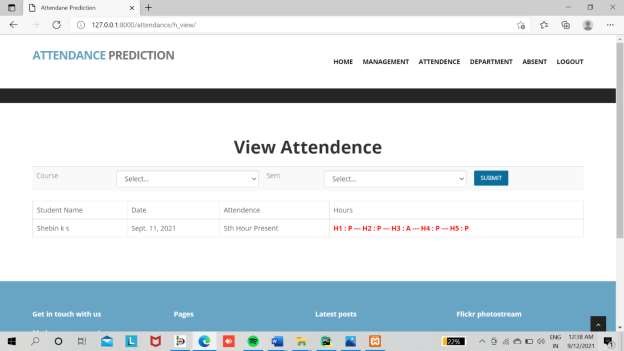
### Teacher Home Page



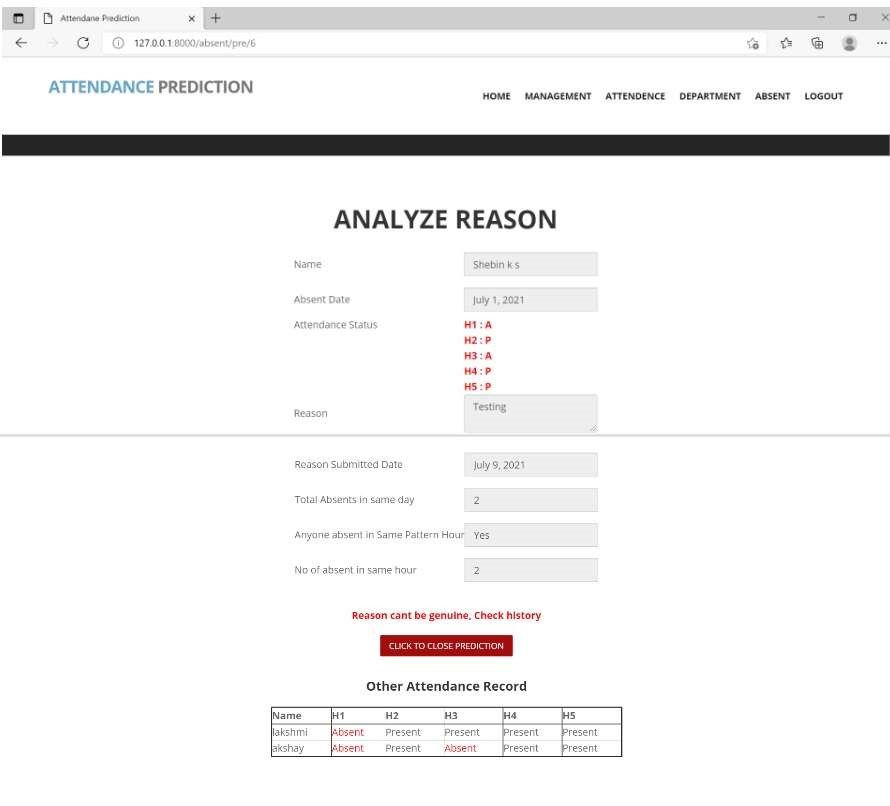
### Teacher Registration page



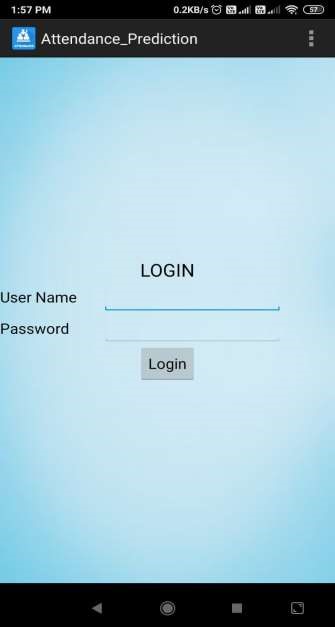
### View Attendance page



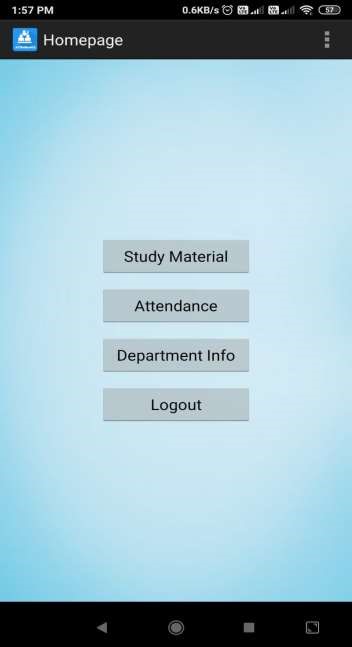
### Reason analysis page



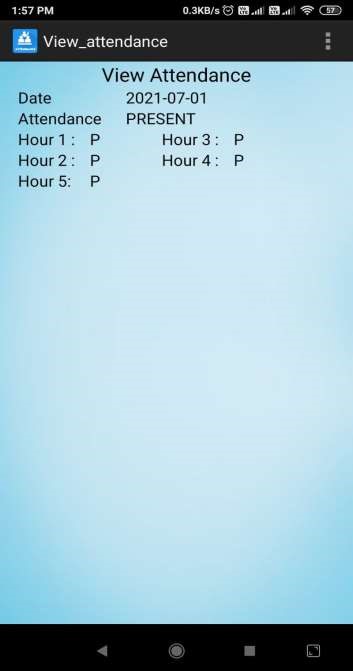
### Student login



### Student home



### Student attendance



#### Student Submit Absent Reason

