**Attendance System Using Face Recognition Using Raspberry Pi**

Submitted By:

**Mr. Pranav Sawant,**

**Mr. Tanish Bhajnik,**

**Mr. Lokesh Vyas**

BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY

2022-2023

UNDER THE GUIDANCE OF

**Ms. Sagarika Prakash**

Submitted in fulfilment of requirement for qualifying B.Sc. (I.T.)

Semester V Examination



**Thakur College of Science and Commerce, Thakur Village, Kandivali (E),**

**Mumbai, Maharashtra-400101**



**Certificate**

This is to certify that the project titled “**Attendance System Using Face Recognition Using Raspberry Pi**” is undertaken at the “Thakur College of commerce and Science” by **Mr. Pranav Sawant (Seat No. \_\_\_\_\_\_\_), Mr. Tanish Bhajnik (Seat No. \_\_\_\_\_\_\_)** and **Mr. Lokesh Vyas (Seat No. \_\_\_\_\_\_\_)** in fulfillment of B.Sc. I.T. degree (Semester V) Examination had not been submitted yet for other examination and does not form part of any other course undergone by the candidates.

It is further certified that the candidates have completed all required phases of the project.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Head of Department

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project Guide Examiner

College seal

**Acknowledgement**

We take much pride in presenting our project. During the development of our project, if we would like to mention the names of certain individuals, without whose assistance, our project would have been difficult undertaking indeed.

We hereby pleased to have this opportunity to express our deep sense of gratitude for our project on “**Attendance System Using Face Recognition Using Raspberry Pi**”, further we are very thankful to our Head of Department “**Dr. Santosh Kumar Singh**”, our internal project guide “**Ms. Sagarika Prakash**” whose valuable guidance and suggestions helped us in accomplishing our project.

Last but not the least, we would like to thank all our friends, family members, non-teaching staff and colleagues for their support and individual help.

**Index**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Topic** | **Pg No** |
| **1**  1.1  1.2  1.3  1.4  1.5  1.6 | **Introduction**  Objective and Scope  Theoretical Background  Problem Definition  Software Requirement Specification (SRS)  Feasibility Study  Details of Hardware and Software | **6 – 11**  6  6  7  7  9  11 |
| **2**  2.1  2.2  2.3  2.4  2.5  2.6  2.7  2.8  2.9  2.10  2.11  2.12  2.13  2.14  2.15  2.16 | **System Analysis and Design**  Prototyping Model  Detailed Life Cycle  Data Flow Diagram  Flow Chart  Entity- Relationship Diagram  Class Diagram  Use-Case Diagram  Activity Diagram  Component Diagram  Table Design  Deployment Diagram  Circuit Diagram  Component Level Description and Specification  Block Diagram  State Transition  System Architecture Diagram | **12 – 43**  12  14  16  21  24  25  26  32  35  36  38  39  40  41  42  43 |
| **3**  3.1  3.2 | **System Planning**  Gantt Chart  Activity Chart Diagram | **44 – 45**  44  45 |
| **4** | **Cost and Benefit Analysis and Software Parameter Estimation** | **46 – 47** |
| **5** | **Future Work** | **48 – 49** |
| **6** | **Reference** | **50** |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Fig. 2.1** | **Prototyping Model** | **12** |
| **Fig. 2.2** | **Detailed Life Cycle** | **15** |
| **Fig. 2.3** | **Context Diagram (Level 0)** | **17** |
| **Fig. 2.4** | **Admin Level 1 DFD** | **18** |
| **Fig. 2.5** | **Faculty Level 1 DFD** | **19** |
| **Fig. 2.6** | **Student Level 1 DFD** | **20** |
| **Fig. 2.7** | **Flow Chart (Raspberry Pi)** | **22** |
| **Fig. 2.8** | **Flow Chart (Website)** | **23** |
| **Fig. 2.9** | **Entity- Relationship Diagram** | **24** |
| **Fig. 2.10** | **Class Diagram** | **25** |
| **Fig. 2.11** | **Admin Login Use-Case** | **26** |
| **Fig. 2.12** | **Student Login Use-Case** | **27** |
| **Fig. 2.13** | **Faculty Login Use-Case** | **28** |
| **Fig. 2.14** | **Admin Function Use-Case** | **29** |
| **Fig. 2.15** | **Faculty Function Use-Case** | **30** |
| **Fig. 2.16** | **Student Function Use-Case** | **31** |
| **Fig. 2.17** | **Admin Activity** | **32** |
| **Fig. 2.18** | **Faculty Activity** | **33** |
| **Fig. 2.19** | **Student Activity** | **34** |
| **Fig. 2.20** | **Component Diagram** | **35** |
| **Fig. 2.21** | **Deployment Diagram** | **38** |
| **Fig. 2.22** | **Circuit Diagram** | **39** |
| **Fig. 2.23** | **Raspberry Pi 3 Board** | **40** |
| **Fig. 2.24** | **USB Camera** | **40** |
| **Fig. 2.25** | **Block Diagram** | **41** |
| **Fig. 2.26** | **Admin and Faculty State Transition** | **42** |
| **Fig. 2.27** | **Student State Transition** | **42** |
| **Fig. 2.28** | **System Architecture Diagram** | **43** |
| **Fig. 3.1** | **Gantt Chart** | **44** |
| **Fig. 3.2** | **Activity Chart Diagram** | **45** |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Table 1** | **Hardware Details** | **11** |
| **Table 2** | **Software Details** | **11** |
| **Table 3** | **Admin Table** | **36** |
| **Table 4** | **Faculty Table** | **36** |
| **Table 5** | **Student Table** | **37** |
| **Table 6** | **Attendance Table** | **37** |
| **Table 7** | **Unadjusted Function Point (UFP)** | **46** |
| **Table 8** | **Complexity Adjustment Factor (CAF)** | **47** |

1. **Introduction**

“*Automation is Solving the Problem once and then putting it on Autopilot.*”

- Michael Hyatt.

Attendance is an important part of daily classroom evaluation. At the beginning and ending of class, it is usually checked by the teacher, but it may appear that a teacher may miss someone or some students answer multiple times. Face recognition-based attendance system is a problem of recognizing face for taking attendance by using face recognition technology based on high-definition monitor video and other information technology.

**1.1 Objective and Scope:**

The objective of this process is to create a software that can autonomously mark the attendance and store them in a database, this technology aims to upgrade the traditional method of marking attendance. The current methods of attendance monitoring are mainly either by speech e.g.: Roll Call or smart cards both these systems are extremely vulnerable to proxies. Our Project has the potential to eliminate this as it collects the attendance data by the means of facial scanning. Unlike the current methods the entries by facial recognition are extremely difficult to fake adding a layer of security to the process. The machine mainly comprises of two main components namely the camera module and the brains in this project we have decided to use raspberry pi as the processor because of its compact size, ready and cheap availability and the pi camera module, the language used is python. The Goal of our project is simple, to create an attendance monitoring system that uses facial recognition to record and mark attendance in the database that should be accessible remotely. As the system uses facial recognition it is much more secure and accurate in comparison to the current methods and virtually does not consume any time as it can work autonomously. As the system is a type of calm technology it can also be used for security and for entering places such as classrooms as it can segregate authorized personnel from the unauthorized ones and send an alert to the admin should an unauthorized person enter the class.

**1.2 Theoretical Background:**

Many organizations, companies and institutions are taking periodic attendance using RFID methods, Biometric Fingerprint method and Registers. These methods generally take more time for calculation. RFID uses electromagnetic fields to automatically identify and track tags attached to persons. Biometric fingerprint identification systems employ fingerprint as a unique identity. It is one of the most accurate systems running effectively today. But recognition of an individual fingerprint from a set of enrolled fingerprints is a difficult process. The fingerprint system does not reveal any information regarding the original fingerprint. This may have been proved to be false as many algorithms reveal that a fingerprint can be reconstructed with minute templates. Iris Recognition is another type of implementation where the iris of people is scanned, stored and then retrieved for the comparison and attendance is managed automatically in the server. But there is difficulty in capturing iris of the students or employees and hence a fast implementation of face recognition with decreased illumination effect can be used.

**1.3 Problem Definition:**

The current systems that are used for updating attendance automatically are usually RFID based, Biometric based. Usually, the manual method of taking attendance is difficult and a time-consuming process. Hence it is important to construct an efficient method for managing attendance automatically. Other than being time consuming these age-old methods can be inaccurate at times and can easily be used to fake attendance.

**1.4 Software Requirements Specifications (SRS):**

There are two types of requirements from a software application: Functional requirements and strategic or Non-Functional requirements.

**1.4.1 Functional Requirements:**

‘Functional Requirements’ of a software application refers to what functionality the software should include. Functional requirement includes but is not limited to calculations, Data processing, data manipulation and storage and such that describes or creates a structure describing what the software must accomplish.

The following are the functional requirements for “*Attendance System Using Face Recognition Using Raspberry Pi”*:

Recognizing faces marking attendance –

The camera acts as a motion detector in passive mode upon sensing motion it enters active mode and starts searching for faces with the help of knowledge from its training data set Upon detecting the faces of the user it clicks multiple pictures and runs it through the data in the database with the help of the algorithms it compares the images pixel by pixel till it finds the picture with the highest match rate, then the system matches the face to its file and marks it the user as present. If the user is moving or for some factor the camera is unable to get a viable picture, it will alert the user regarding it through a signal, similarly the system signals an acknowledgement to the user upon successful detection. The Users whose faces were not detected during the attendance time period are marked as absent.

Data manipulation –

Data manipulation consists of changing data in any way throughout the software, especially in the database. Assuming the owner wishes to add a new employee or remove an employee for whatever reason, they can simply access the database and update it by removing or adding their files. They can also mark or unmark attendances. As this is an extremely powerful feature that can be used in many ways it is reserved only to the admin and faculty level users. An admin user has access to every module and authorization to edit it.

View attendance –

The main feature of *“Attendance System Using Face Recognition Using Raspberry Pi”* is to be able to view attendance. The access to view attendance is given to every user however there are a few restrictions, considering an example of a university, the student user can only view their own attendance, the faculty level users can view their own attendance as well as the attendance for all students and edit student attendance, the admin level user can View attendance of all personnel and edit it.

Account Creation/Registration –

In order to access the system, the system needs to be able to authenticate and authorize the user, by registering themselves they create a profile for themselves that the system uses to authenticate them. A profile must be created in order to access the system and hence is the primary and basic act the user must do. The user can register themselves through the registration page where the system asks for various info about the user such as name, uid, roll no, etc. and asks to create a password and username to be used later for login. The user can use any device for this process, after filling in the details the user will be taken to a facial registration page, over here pictures of the user will be clicked and uploaded to the database for facial recognition.

Account login and maintenance –

In order to use the system, any new user will have to Log themselves in. This module will be available to everyone as it is an essential module for the functioning of the software. During login the user will have to input their key information such as roll number, employment id etc. and password. The ‘Login’ screen is, for obvious reasons, the first screen to be displayed after selecting respective module when the software is invoked. This would accept the User-ID and Password, and if they are confirmed to be correct, opens the landing page for the user depending on his account type. The option for registration is also available on this page. However, the administrators have a bigger role to play here. They can directly create and add accounts as well as access any account.

**1.4.2 Non- Functional Requirements:**

Also known as strategic requirements, the non-functional requirements consist of criteria to judge the system operation and usability rather than the specific functions or features it consists of. It is also known as Qualities of the system and defines system attributes such as security, reliability, performance, maintainability, scalability.

Extensibility –

“*Attendance System Using Face Recognition Using Raspberry Pi”* is a system that has Excellent future scope. It can be integrated into all sorts of work environments for functions such as tracking attendance of employees, students etc. To support it, it should allow integration of new features without affecting the existing system.

Maintainability –

Given the scale of the system which may exponentially increase in the near future, it is extremely important for the *“Attendance System Using Face Recognition Using Raspberry Pi”* to be maintainable.

Security –

As “*Attendance System Using Face Recognition Using Raspberry Pi”* would be storing confidential data, it is necessary for the system data to be secure. Several methods such as authentication and encryption issuing session id’s can be used for this purpose.

Reliability –

Reliability refers to the Mean Time Between Failures (MTBF). This is the predicted elapsed time between inherent failures of a system during operation. This should be an acceptable value. The system must perform without failure in 95 percent of use cases during a month

Usability –

*“Attendance System Using Face Recognition Using Raspberry Pi”* must be user friendly to maximize adaptability and ease the transition from other methods.

Scalability –

The performance of *“Attendance System Using Face Recognition Using Raspberry Pi”* should be consistent irrespective of the load on the system, it should work efficiently and accurately across devices to ensure that it gives equal and independent performance on any device.

**1.5 Feasibility study:**

A feasibility study is a detailed analysis that considers all of the critical aspects of a proposed project in order to determine the likelihood of it succeeding. Success in business may be defined primarily by return on investment, meaning that the project will generate enough profit to justify the investment. However, many other important factors may be identified on the plus or minus side, such as community reaction and environmental impact. A feasibility study assesses the potential for success of the proposed plan or project by defining its expected costs and projected benefits in detail. It's a good idea to have a contingency plan on hand in case the original project is found to be infeasible. A feasibility study is an assessment of the practicality of a proposed plan or project. A feasibility study analyzes the viability of a project to determine whether the project or venture is likely to succeed. The study is also designed to identify potential issues and problems that could arise while pursuing the project. Hence it is necessary to conduct a feasibility study before starting a project.

**1.5.1 Technical:**

A technical feasibility study assesses the details of how a project intends to deliver a product or service to customers. It takes into account the various technical aspects like modules, software, algorithms etc. and the hardware requirements the project utilizes. It's the logistical or tactical plan of how the project will produce, store, deliver, and track its products or services. It is assessed based on the Prepared Outline of the project, the labor and the material requirements. This system is developed in a flexible form, which covers all operation services with the help of optimized code logic to bridge the hardware and software along with Python, HTML, CSS and MySQL, for logic, UI Framework and data management respectively. The Project is deemed Technically Feasible.

**1.5.2 Economical:**

Once the technical feasibility and market studies are complete, it is time to determine Business Feasibility. The first purpose of this effort is to financially model the venture opportunity and achieve a break-even analysis. In other words, based upon the costs of goods sold, capital costs, and management and administration, how much revenue generated from units sold is required to break-even and over what period of time. The simple objective is to determine what level of revenue is required to satisfy the return on investment demanded by the founder and/or the investors. Economic feasibility elements include, but are not limited to: Increased agency revenue, decreased agency revenue, increased agency costs, decreased agency costs, increased revenue to other agencies and/or the general public, decreased revenue to other agencies and/or the general public, Increased costs to other agencies and/or the general public, Decreased costs to other agencies and/or the general public, Other public benefits. In essence economic feasibility study is done to understand if a project will incur a profit or loss to an organization and its extent and thus the execution of the project is dependent on the ROI (Return on Investment) estimated. This project aims at automating the task of attendance marking for organizations such as schools, universities, Companies, etc. The scope and requirement of the project are estimated and aimed to serve a huge market. The Materials required for hardware are readily available and cheap at mass and requires minimal maintenance.

**1.5.3 Resource:**

Resources that are required for the *“Attendance System Using Face Recognition Using Raspberry Pi”* include:

Programming device (Laptop),

Hardware (readily available, Bulk manufactured),

Programming tools (freely available),

Programming individuals.

So, it's clear that the project has the required resource feasibility.

**1.5.4 Operational:**

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. its success is dependent on how well the humans/ users interact with it, If the software for a new system is too difficult to use, employees may make too many errors and avoid using it. Thus, it would fail to show operational feasibility. The proposed system will be deemed operationally feasible if users can use our site with ease. There are many ways of achieving this including but not limited to user training, Designing user friendly/ intuitive UI, etc. This “Attendance System Using Face Recognition Using Raspberry Pi” was designed while keeping it’s operationality and user interaction at heart, making sure that the UI is easy to navigate by users of every level ensuring excellent functionality and use of the system by the client on their end. Thus, in conclusion *“Attendance System Using Face Recognition Using Raspberry Pi”* satisfies all the necessary feasibility requirements and can be deemed feasible.

**1.6 Details of Hardware and Software:**

**1.6.1 Hardware Details** **–**

|  |  |  |
| --- | --- | --- |
| 1 | Hard Disk | 1 TB |
| 2 | RAM | 16 GB |
| 3 | Processor | Intel i5 11th Gen |

**Table: 1**

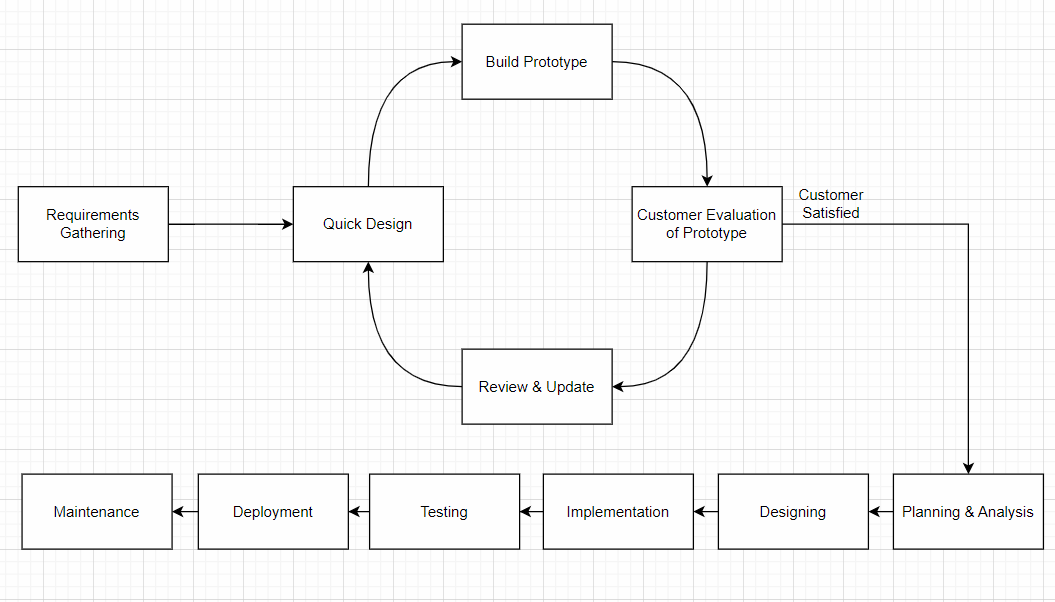
**1.6.2 Software Details** **–**

|  |  |  |
| --- | --- | --- |
| 1 | Operating System | Windows 11 |
| 2 | Database System | Xampp Server |
| 3 | Front End | Visual Studio Code |

**Table: 2**

1. **System Analysis and Design**

**2.1 Prototyping Model:**



**Fig. 2.1**

The Prototyping model functions by creation of a small basic functioning unit of the project with respect to the software requirements and then enhancing it through various iterations until the final deployable product is achieved. In simple words the project is broken down into smaller manageable tasks or modules and these modules are developed over various prototyping iterations.

*“Attendance System Using Face Recognition Using Raspberry Pi”* is a typical project whose specifications and requirements align with the development structure proposed by The Prototyping model. As a result, it was the model selected. As the requirements are well defined and the team is familiar with the development platform however has not had experience with concepts such as machine learning, data cleaning, hardware implementation and deployment building the project using this model satisfied the requirements such as feasible and time constraint. The Prototyping model provides room for error and testing to the team, it allows the team to iterate upon the hardware to test the various permutations and combinations that might arise throughout the development process and allow them to test these to find the one that functions best with the system. At the end of each prototyping cycle the output was a functional prototype of the software. After every iteration the software was tested for bugs and glitches along with functionality. As there is no client, the functional prototypes were discussed among the members and the guide to obtain feedback. Before the final prototype was deployed a complete risk analysis and testing was performed to determine the robustness of the software and determine if the software functions as desired in the intended environment. Keeping in line with The Prototyping model development methods, the project was initialized with the gathering of requirements to check if the project satisfied development criteria. Following the requirement gathering when the project satisfied all necessary criteria a quick sketch was developed. This sketch was the very first prototype of the project and all other prototypes in some way originate from this. A simple working structure was created to give some idea about functioning to the client with the help of a simulator. A basic functioning prototype was created to show the user a partial implementation of the project. This partial implementation was showed to the client to gain input to understand if it satisfied their expectations and requirements. Upon further iterations a satisfactory prototype was reached and the actual development of the project was started. The system underwent various iterations to accommodate various needs and requirements that were discovered throughout the development. For each module various prototypes were created, tested and eliminated until a satisfactory prototype that was ready to be integrated into the system was achieved.

**Requirements Gathering:**

Project Requirements were received and checked if they satisfied development criteria. The criteria were satisfied and the project was accepted for development.

**Quick Design:**

A basic understanding of the functioning of various aspects of the system needs to be created in order to understand client expectations. Keeping the line of communication clear and taking the client input a simple design of the system was created.

**Build Prototype:**

The design was implemented upon and a basic prototype was created with the help of simulator. This prototype was designed to be used as a based concept and be further improved upon. The prototype functioned as a partial implementation to gain approval from the user and verify that it satisfies the expectation of the client. Upon further prototyping a suitable prototype was reached that satisfied the client and was in line with the development team. The current system was built upon iterations of this base prototype.

**Customer Evaluation of Prototype:**

As explained earlier, keeping in line with open communication between the client and the team the partial implementations were shown and discussed with the client and iterated upon until the client demands were met.

**Review & Update:**

Any feedback from the client was reviewed by the team. The reviews were used to add or change functionalities in the prototypes while integrating client ideas and demands in the process. The updated prototype was then shown to the client to obtain feedback.

**2.2 Detailed Life Cycle:**

Our Software Development Life Cycle (SDLC) consists of the following 6 stages -

**Stage 1 – Project Planning & Analysis**

It is the first step of our SDLC. In this stage, we have done the research activity and also, we have gone through the related research papers which have been published in past years. Also, we have referred “Thakur College of Science and Commerce” Website for more related studies.We divided our work into small task. During the planning staged we also decided the costing as well as the requirement part. Also, we have planned how the project will complete in given schedule and budget. In this project we used prototyping model. We decided to keep total 6 phase and for these we use the work break-down structure technique to execute our project. The 6 phases are:

1. Planning and Requirement Analysis
2. Designing
3. Coding
4. Testing
5. Deployment
6. Maintenance

**Stage 2 – Project Designing**

Project Designing represents the features of the software that helps us to develop it effectively, the architecture, the user interface, and the component level detail.

**Stage 3 – Project Execution**

In this phase the plans turned into the action. We allocated our work equally and started our first phase of the project according to the plan.

**Stage 4 – Project Testing**

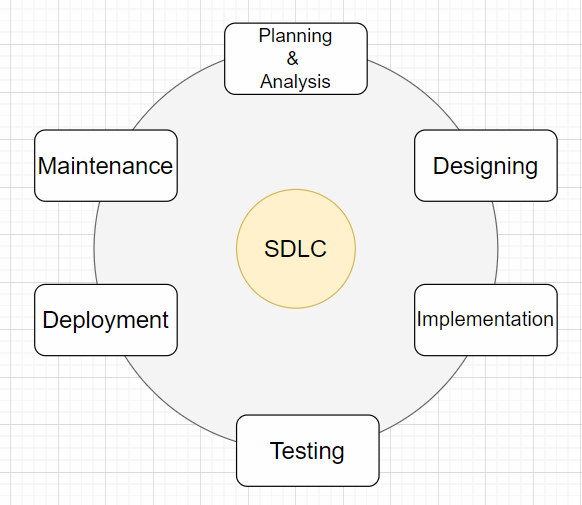
This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

**Stage 5 – Project Deployment**

Once the product is tested and ready to be deployed it is released formally to the Client/User.

**Stage 6 – Project Maintenance**

The purpose of Project Maintenance is to modify and update software applications after delivery to correct faults and to improve performance.



**Fig. 2.2**

**2.3 Data Flow Diagram (DFD):**

A Data Flow Diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled.

A Data Flow Diagram (DFD) represents graphically a flow of data within a system. It illustrates how data is input and output from the system.

It also shows destinations, storage, and sources of the information in the system.

In other words, DFD represents the information flow as well as where data comes from, where data goes and how it is stored.

So, we can say a Data Flow diagram has 4 major elements:

* **Processes** – the main activities that are happening within the system boundary. The process can be as simple as collecting customer data and storing it in the company database. Also, it can be a very complicated process such as creating a report containing bank contracts with customers of all bank clones in a region.
* **External entities** – the sources of information coming to or leaving the system. External entities are outside systems such as people (customers, stakeholders, managers), organizations, computers and other systems that send or receive data from our system.
* **Data stores** – places where data is held such as files or repositories. Data stores show information that is not moving.
* **Data flows** – illustrate the movements that data have between the external entities, data stores, and the processes.

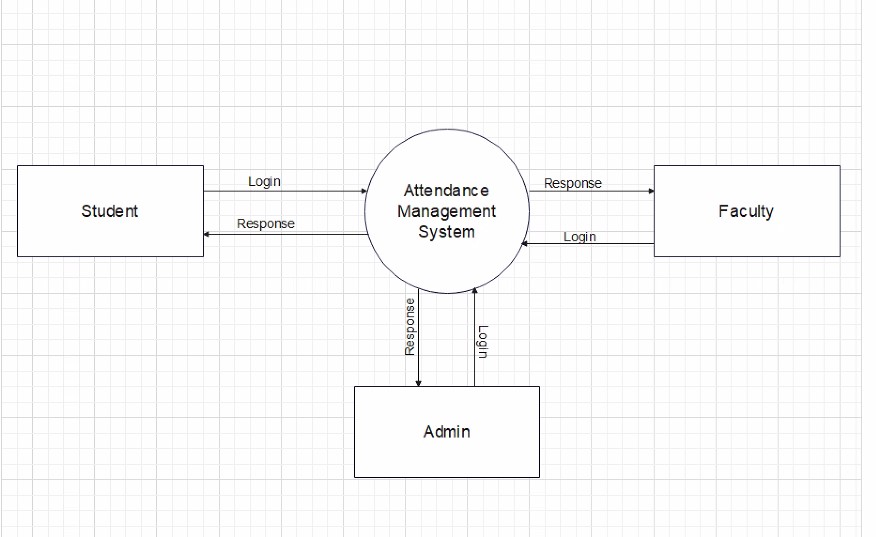
**2.3.1 Context Diagram (Level 0):**

When it comes to simple data flow diagram examples, context one has the top place.

Context data flow diagram (also called Level 0 diagram) uses only one process to represent the functions of the entire system.

It does not go into details as marking all the processes.

The purpose is to express the system scope at a high level as well as to prevent users from deep down into complex details.



**Fig. 2.3**

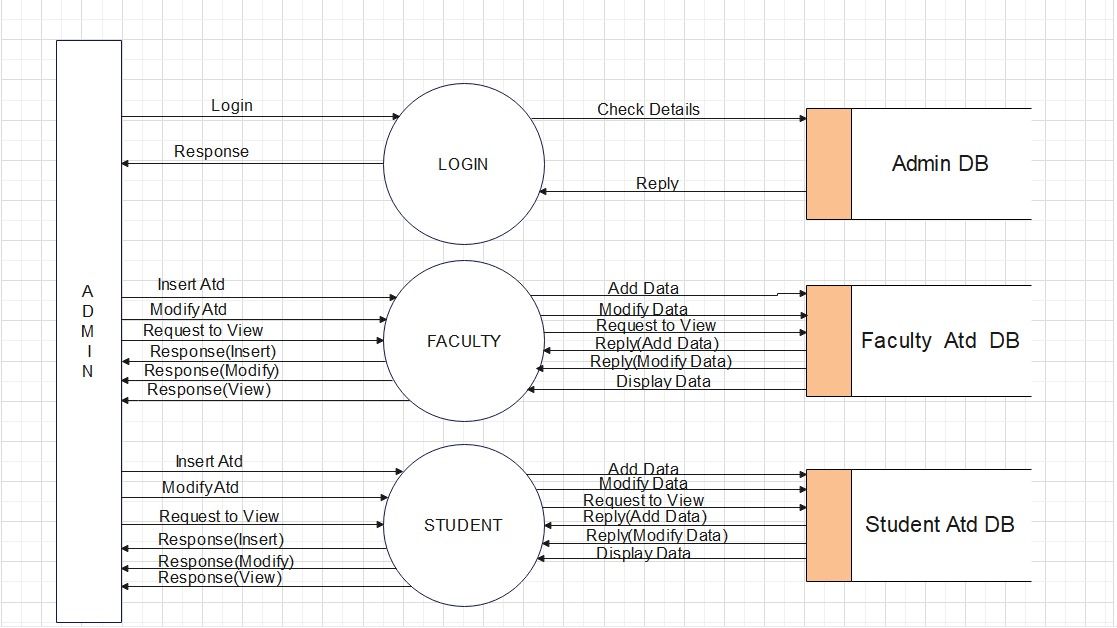
**2.3.2 Level 1 DFD:**

As you saw above context DFD contains only one process and does not illustrate any data store.

This is the main difference with level 1 DFD.

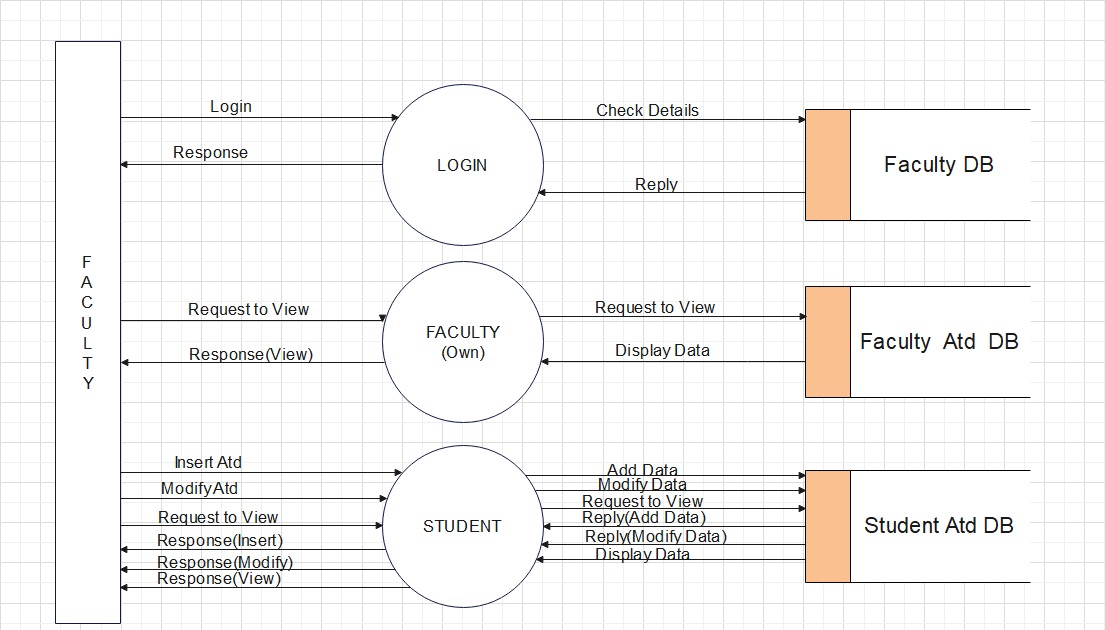
Level 1 DFD breaks down the main process into subprocesses that can then be seen on a deeper level. Also, level 1 DFD contains data stores that are used by the main process.

Admin Level 1 DFD



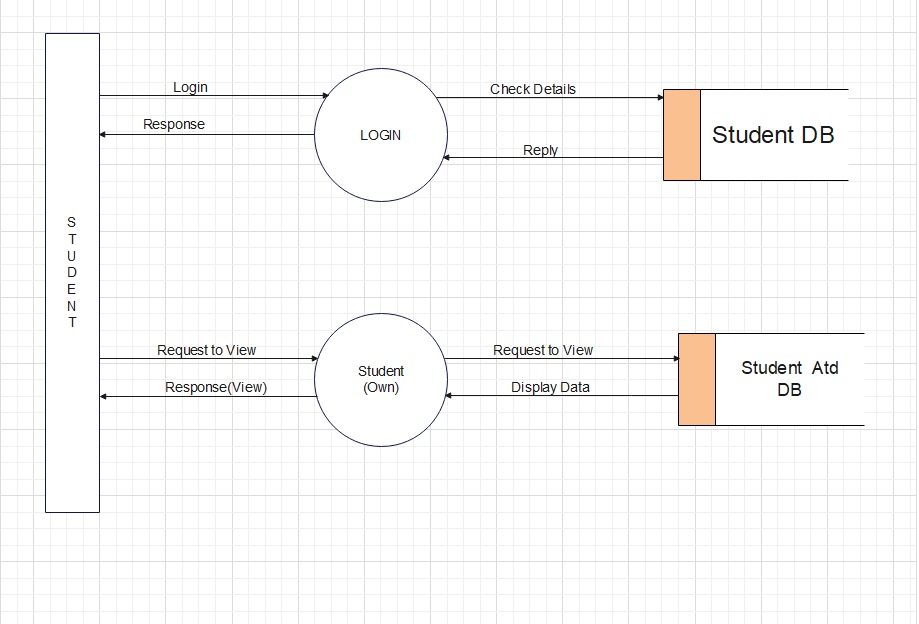
**Fig. 2.4**

Faculty Level 1 DFD



**Fig. 2.5**

Student Level 1 DFD



**Fig. 2.6**

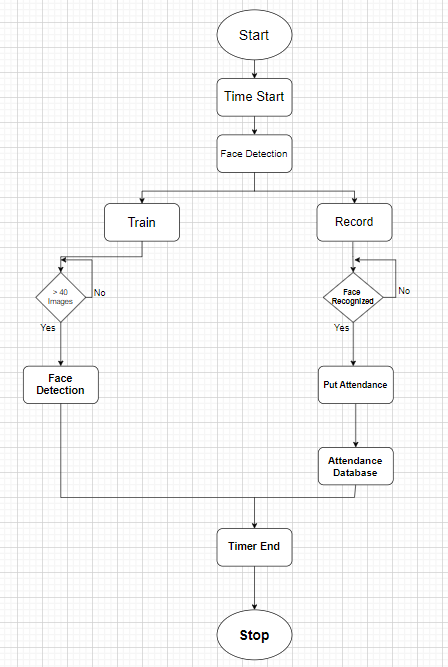
**2.4 Flow Chart:**

The Camera acts as a motion sensor in passive mode, upon detecting motion the camera enters active mode and starts detecting faces with the help of the algorithms. Upon triggering the camera starts a timer to click a burst of images of the detected face. These images, if viable are processed and then sent to the *Face Database* in the form of data. This data is processed in and crossed checked with the existing data in the database to find a match. Assuming the photo taken by the system doesn’t match with any of the existing images in this case the attendance will not be marked autonomously and the user will require to complete the registration process correctly. This database can then be accessed to view or edit the attendance of the individuals.

The user can log into the system with their credentials which will lead them to their respective page with pre allocated clearance or level i.e., *Admin, Faculty, Student.* The ability to edit and update both the databases is reserved to an *Admin Level User*.

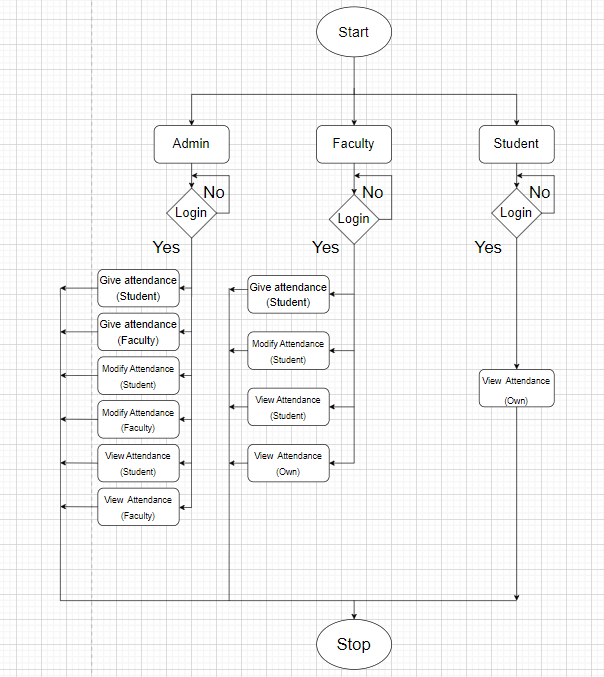
The *Faculty Level* *User* has clearance to mark or remove attendance of an individual from the attendance database and can view any attendance they wish, individual or as a whole. The student level user is limited to viewing their own attendance. Thus, automating the process of attendance marking.

**Flow Chart (Raspberry Pi): -**



**Fig. 2.7**

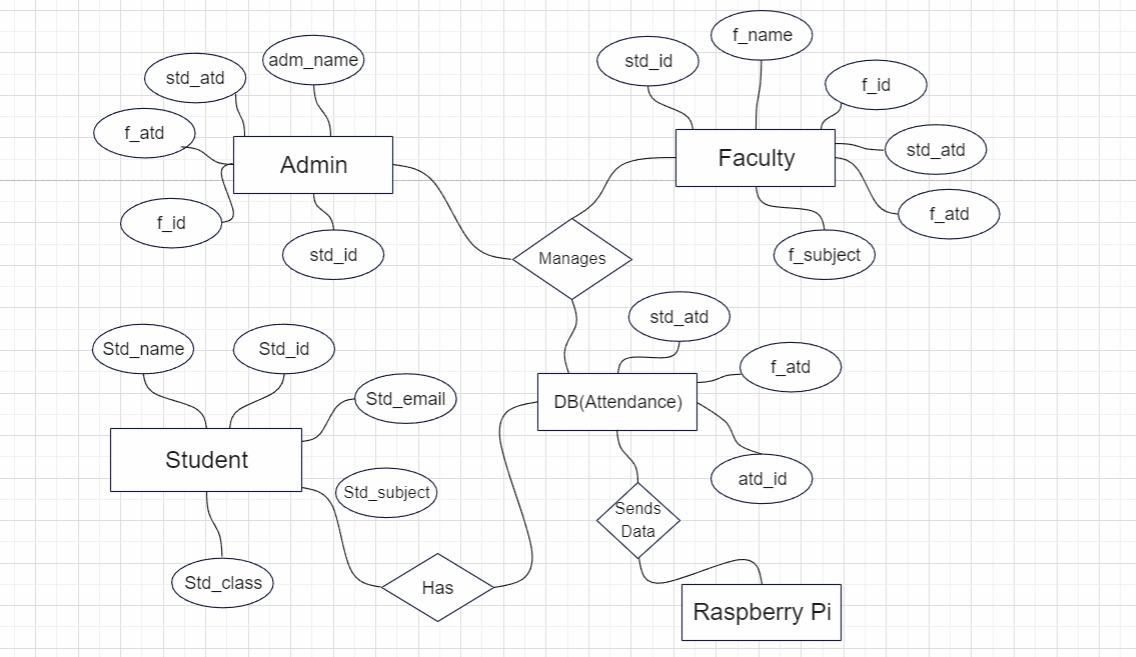
**Flow Chart (Website): -**



**Fig. 2.8**

**2.5 Entity Relationship Diagram:**

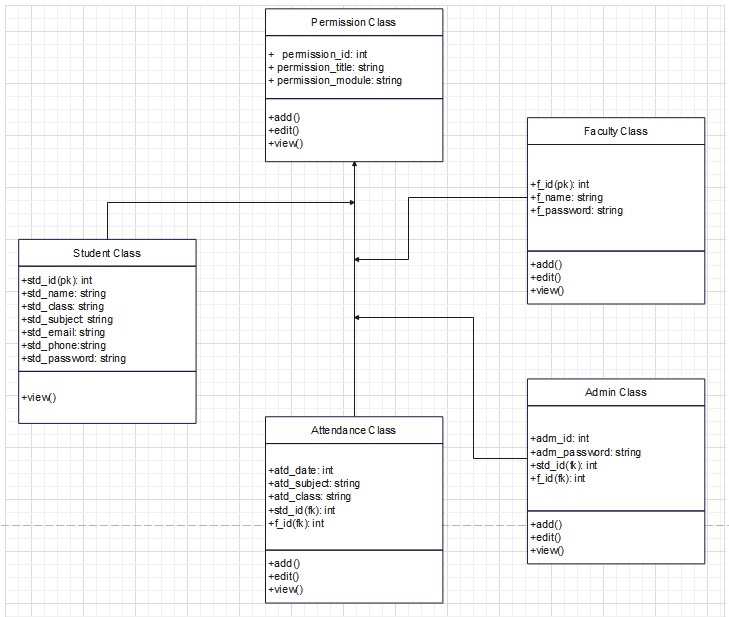
An Entity–Relationship model describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between entities.



**Fig. 2.9**

**2.6 Class Diagram:**

Class diagrams are one of the most useful types of diagrams in UML as they clearly map out the structure of a particular system by modeling its classes, attributes, operations, and relationships between objects. One of the more popular types in UML is the class diagram. Popular among software engineers to document software architecture, class diagrams are a type of structure diagram because they describe what must be present in the system being modeled. No matter your level of familiarity with UML or class diagrams, our UML software is designed to be simple and easy to use.

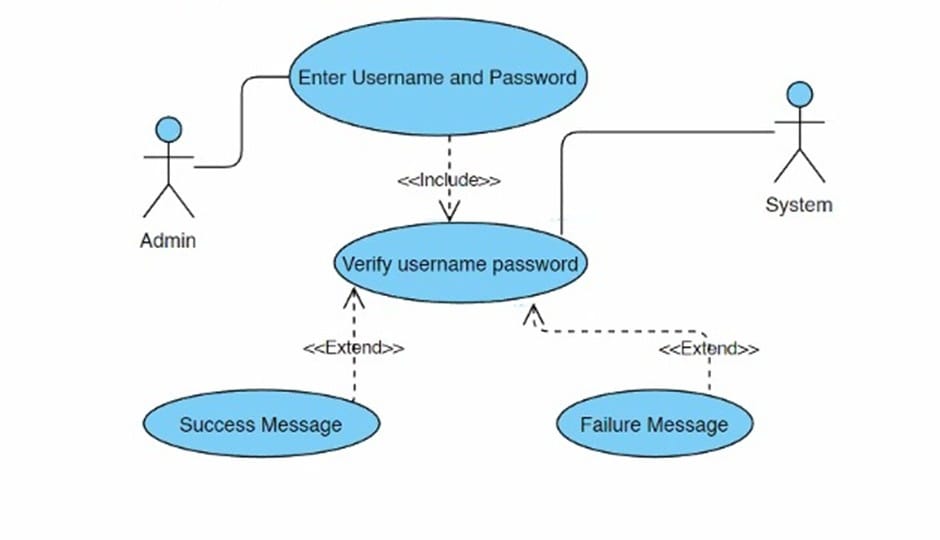


**Fig. 2.10**

**2.7 Use-Case Diagrams**:

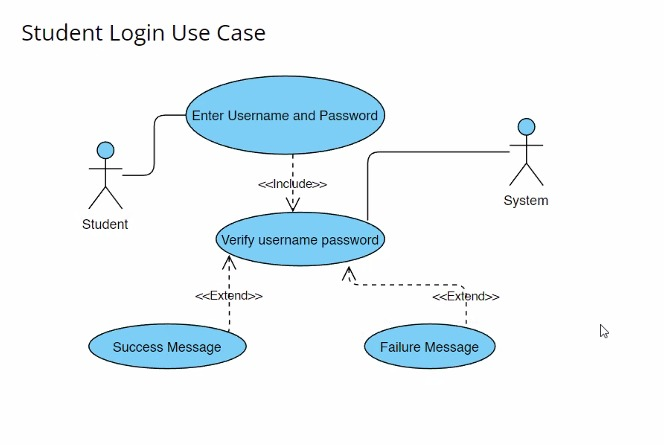
A use case diagram shows various use cases and different types of users the system has and diagram depictions of how the user interacts with the system. It takes into account every possible interaction of developer intended features of the system and the user. Uses cases are used as a software modeling technique that helps developers determine which features to implement, and determine how to gracefully resolve errors.

**2.7.1 Admin Login Use Case:**



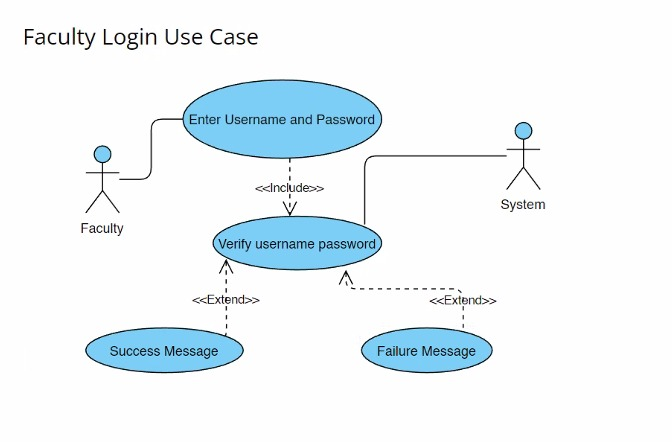
**Fig. 2.11**

**2.7.2 Student Login Use Case:**

****

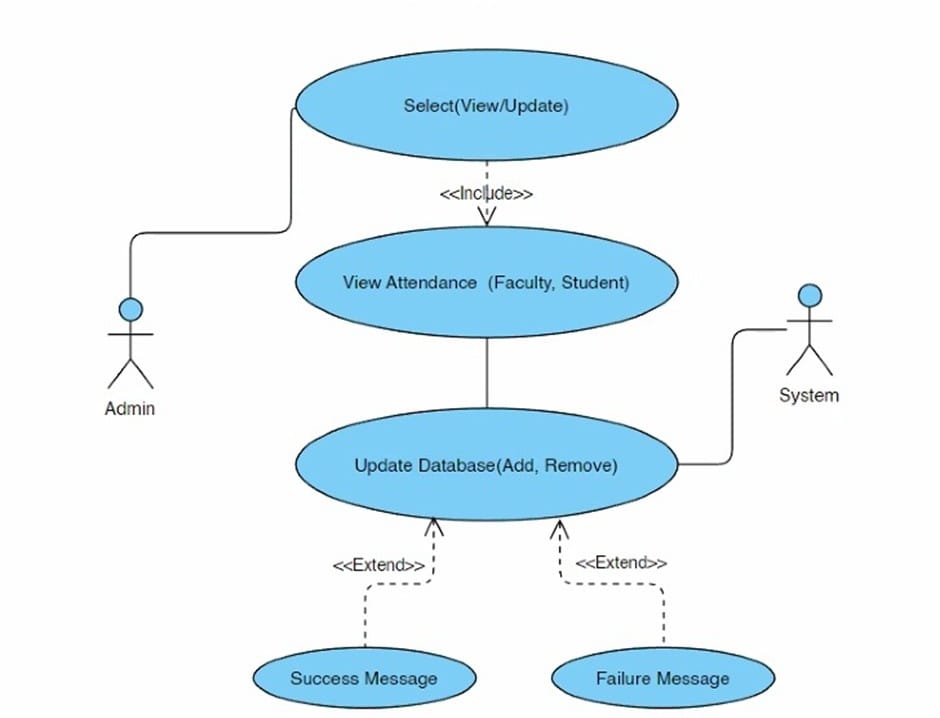
**Fig. 2.12**

**2.7.3 Faculty Login Use Case:**

****

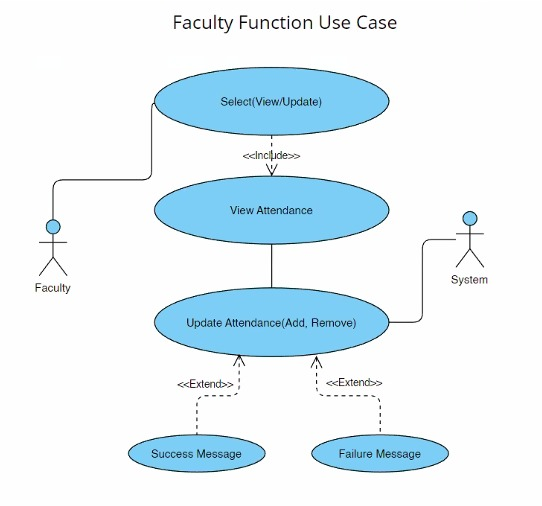
**Fig. 2.13**

**2.7.4 Admin Function Use Case**:



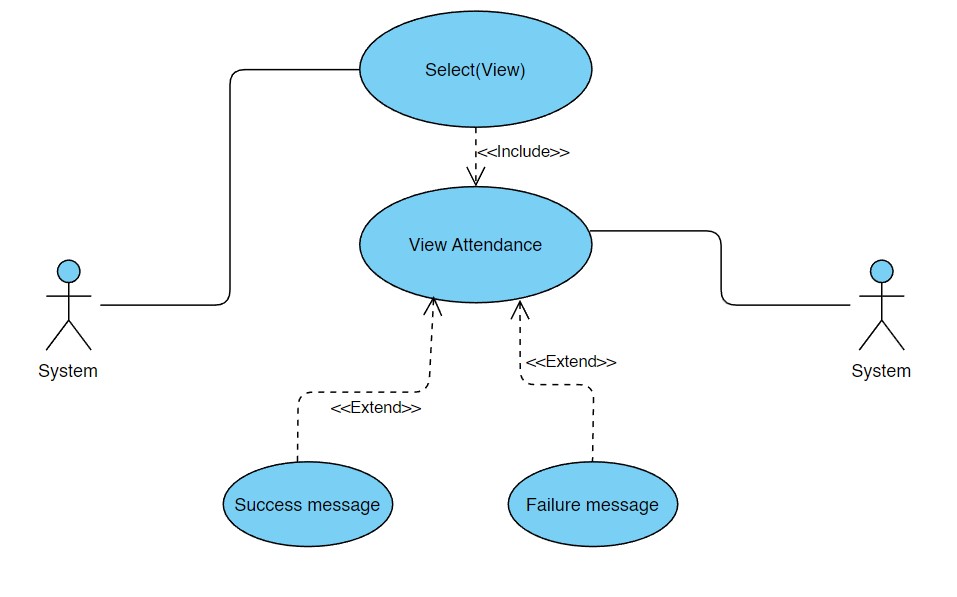
**Fig. 2.14**

**2.7.5 Faculty Function Use Case:**

****

**Fig. 2.15**

**2.7.6 Student Function Use Case:**

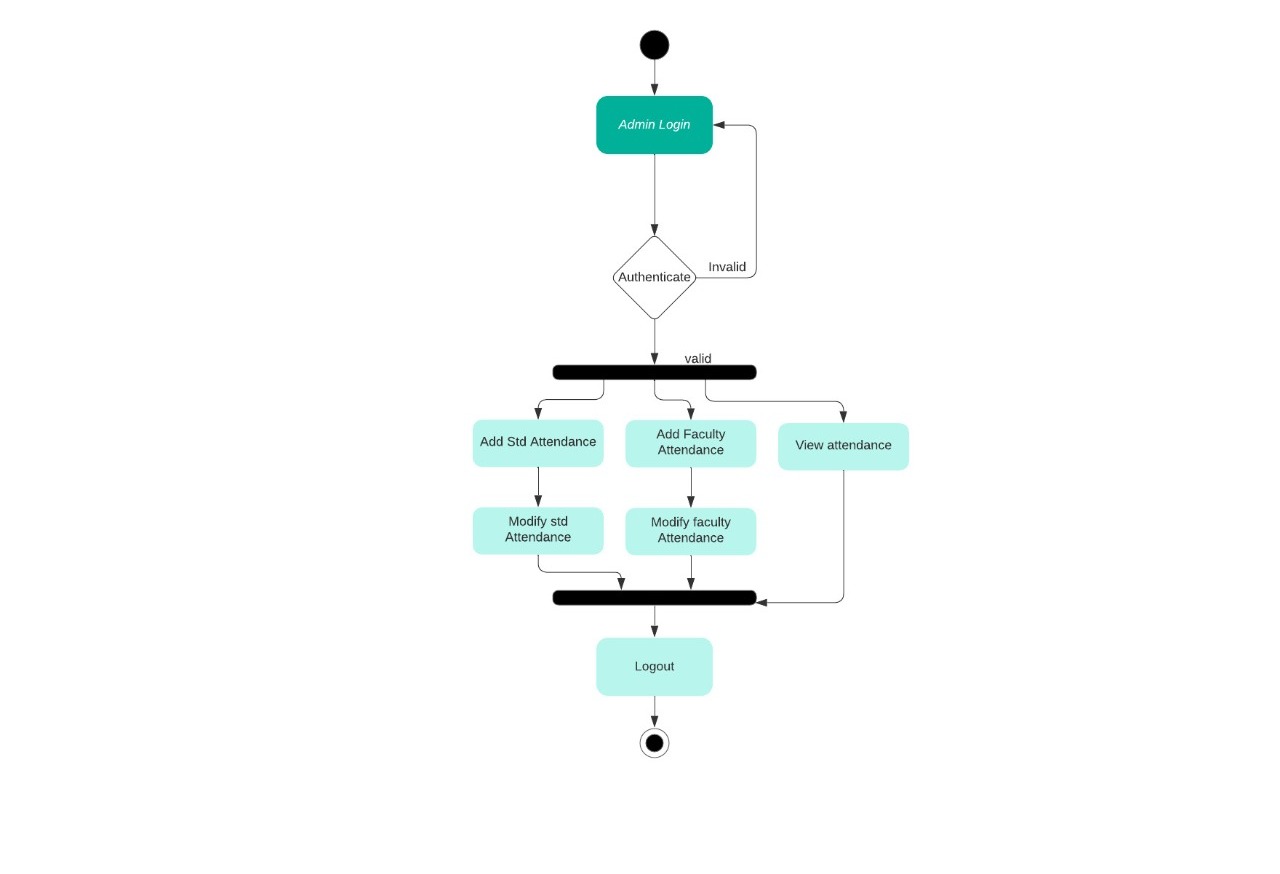


**Fig. 2.16**

**2.8 Activity Diagram:**

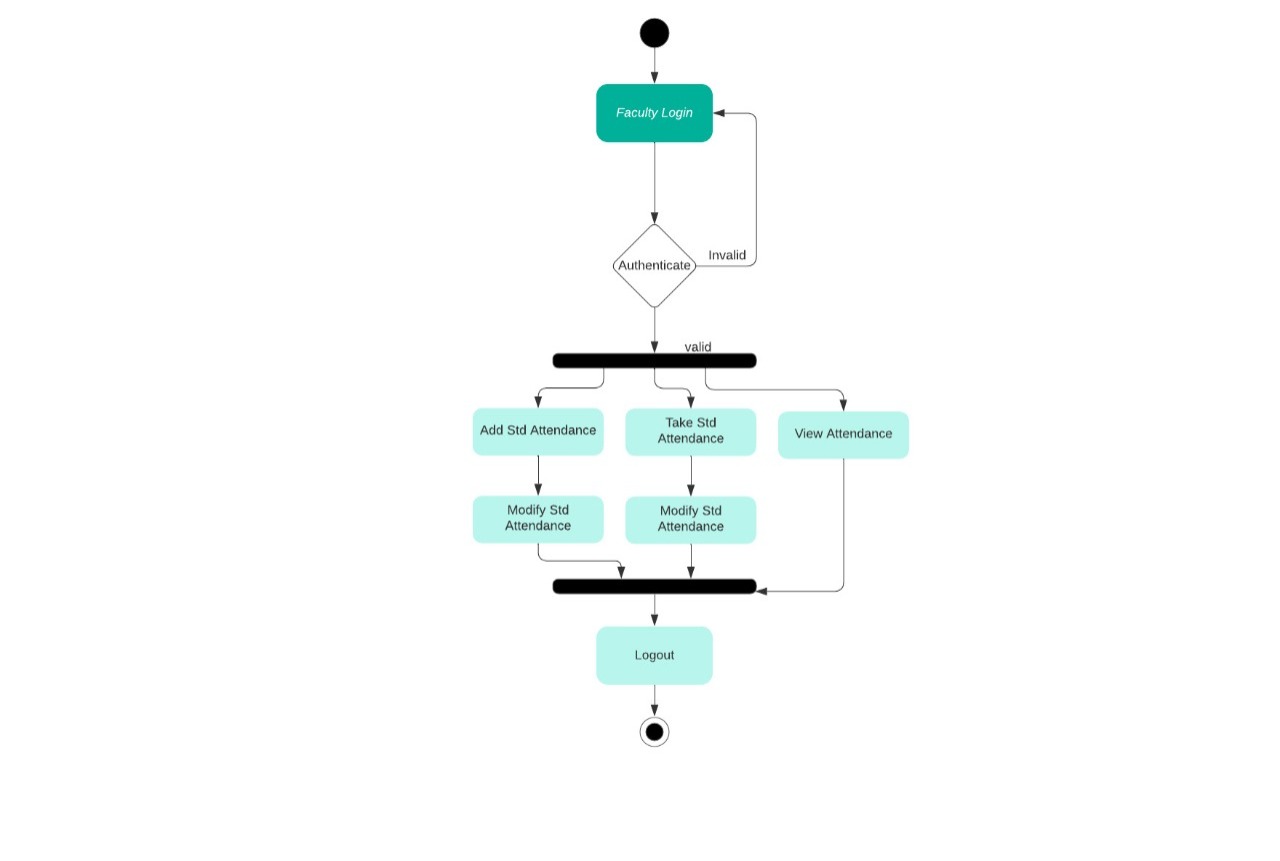
Activity diagram is another important behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity. Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for modeling how a collection of use cases coordinates to represent business workflows.

**2.8.1 Admin Activity:**



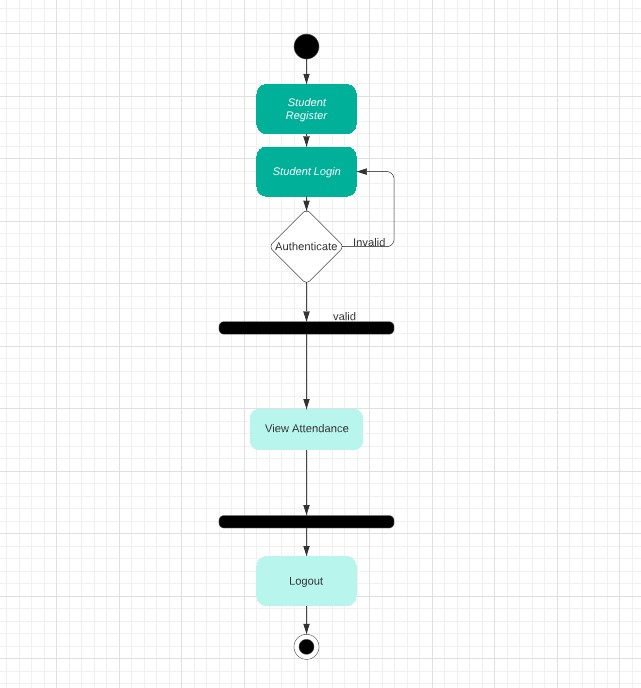
**Fig. 2.17**

**2.8.2 Faculty Activity:**



**Fig. 2.18**

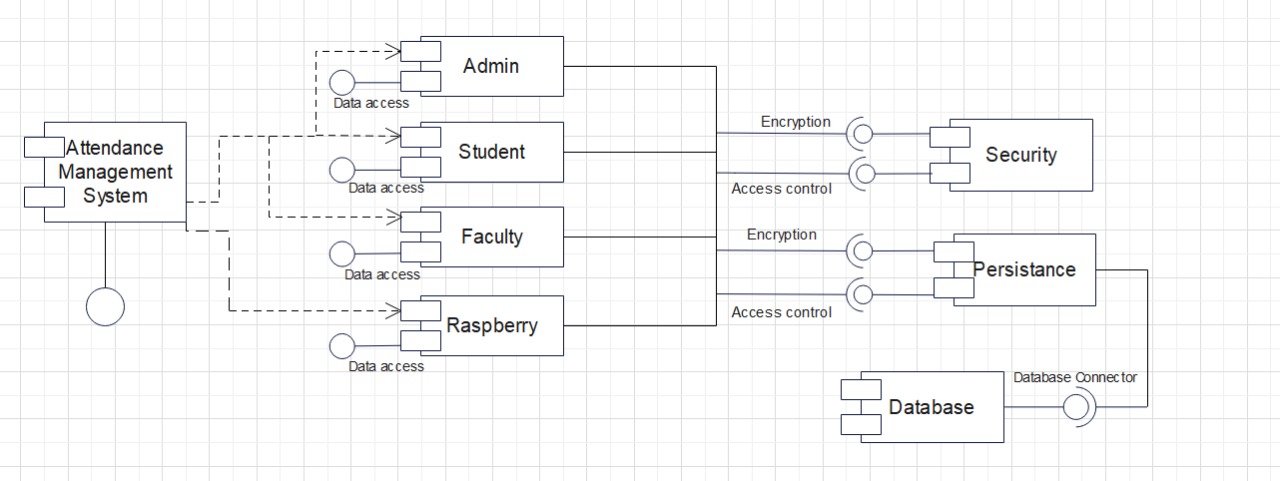
**2.8.3 Student Activity:**



**Fig. 2.19**

**2.9 Component Diagram:**

Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.



**Fig. 2.20**

**2.10 Table Design:**

**2.10.1 Admin Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **attributes** | **Description** | **Datatypes** | **Size** | **Required** |
| Adm\_id | The unique system generated ID given to admin | Integer | 10 | Yes |
| Adm\_Password | Unique Password for admin | Varchar | 50 | Yes |
| F\_id | ID of Faculty | Integer | 10 | Yes |
| Std\_id | ID of Student | Integer | 10 | Yes |

**Table: 3**

**2.10.2 Faculty Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **attributes** | **Description** | **Datatypes** | **Size** | **Required** |
| F\_id | The unique system generated ID given to admin | Integer | 10 | Yes |
| F\_name | To define the Faculty Name | Varchar | 50 | Yes |
| F\_class | Details of different class | Varchar | 10 | Yes |
| F\_subject | Details of different subjects | Varchar | 20 | Yes |
| F\_password | Unique Password for faculty | Varchar | 50 | Yes |

**Table: 4**

**2.10.3 Student Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **attributes** | **Description** | **Datatypes** | **Size** | **Required** |
| Std\_id | The unique system generated ID given to admin | Integer | 10 | Yes |
| Std\_name | To define the Student Name | Varchar | 50 | Yes |
| Std\_class | Details of different class | Varchar | 10 | Yes |
| Std\_subject | Details of different subjects | Varchar | 20 | Yes |
| Std\_email | Email of student | Varchar | 100 | Yes |
| Std\_phone | Contact Number of Student | Varchar | 12 | Yes |
| Std\_password | Unique Password for student | Varchar | 50 | Yes |

**Table: 5**

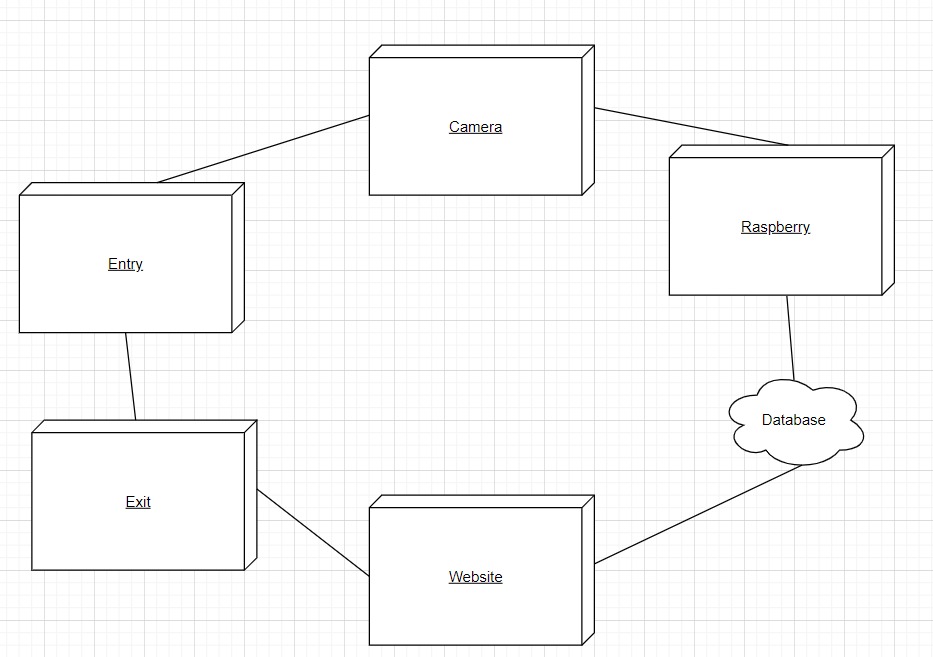
**2.10.4 Attendance Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **attributes** | **Description** | **Datatypes** | **Size** | **Required** |
| Atd\_date | Date of Attendance | Date | 10 | Yes |
| Atd\_subject | Subject of Attendance | Varchar | 20 | Yes |
| Std\_id | ID of Student | Integer | 10 | Yes |
| F\_id | ID of Faculty | Integer | 10 | Yes |
| Atd\_class | Attendance for class | Varchar | 10 | Yes |

**Table: 6**

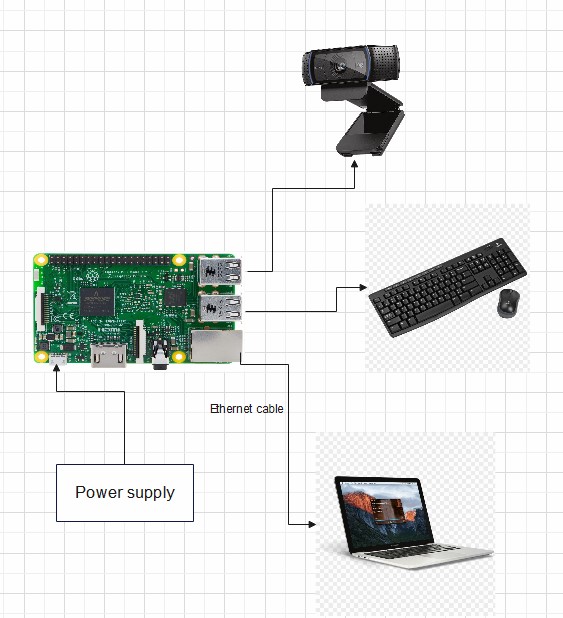
**2.11 Deployment Diagram:**

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

****

**Fig. 2.21**

**2.12 Circuit Diagram:**

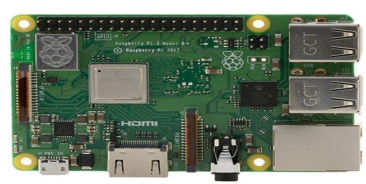


**Fig. 2.22**

**2.13 Component Level Description & Specification:**

1. Raspberry Pi 3 Board:

Raspberry Pi can be called a mini-computer which is of the same size as that of a credit card. The Raspberry Pi is a small and affordable single-board computer that we will use to design and develop practical IoT devices while learning about programming language and computer hardware systems. In addition, you will learn how to set up the Raspberry Pi, get a Linux operating system running, and write and execute some basic Python code on the Raspberry Pi. You will also learn how to use Python based IDE (integrated development environments) for the Raspberry Pi and how to trace and debug Python code on the device.



**Fig. 2.23**

1. USB Camera:

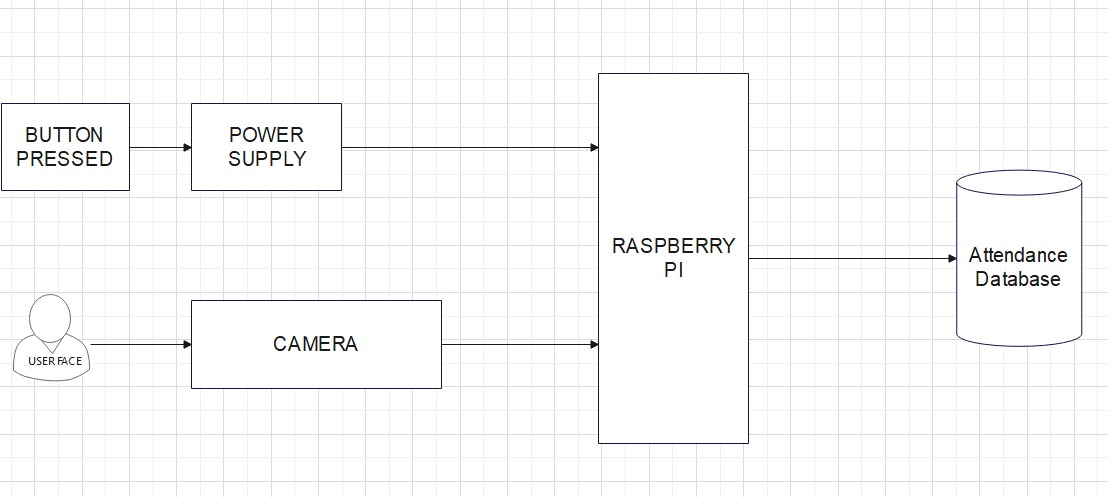
The camera plays a vital role in mechanization purposes. The camera is used in our work to monitor the room from a remote place and to capture video to detect faces. For interfacing the USB camera with Raspberry Pi use the terminal and run "sudo raspi-config". If the "camera" option is not scheduled, you will need to dash a few instructions to update your Raspberry Pi. For this Run "sudo apt-get update" and "sudo apt-get upgrade" commands. Again run "sudo raspi-config" again - you must now see the "camera “preference and enable it.



**Fig. 2.24**

**2.14 Block Diagram:**

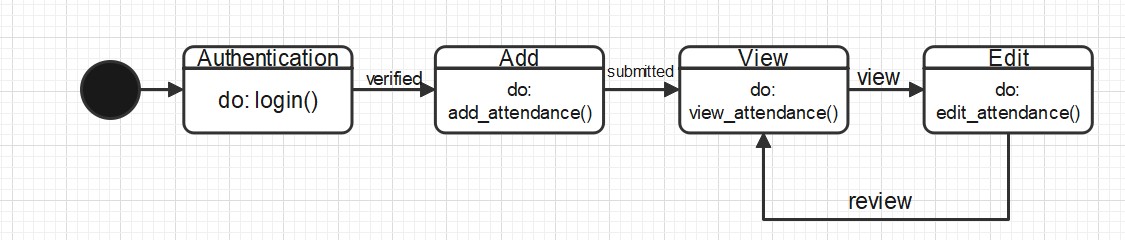
A block diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. They are heavily used in engineering in hardware design, electronic design, software design, and process flow diagrams.

****

**Fig. 2.25**

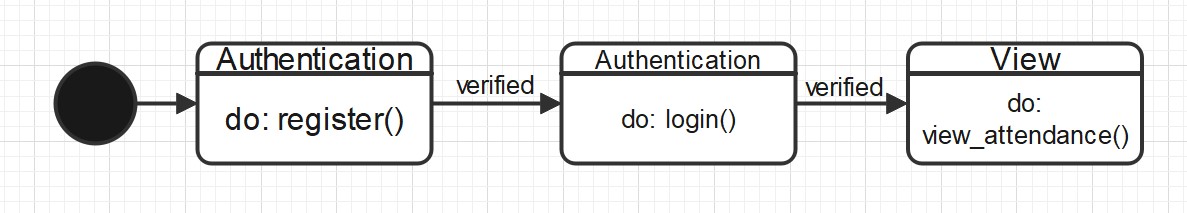
**2.15 State Transition:**

**2.15.1 Admin and Faculty State Transition**



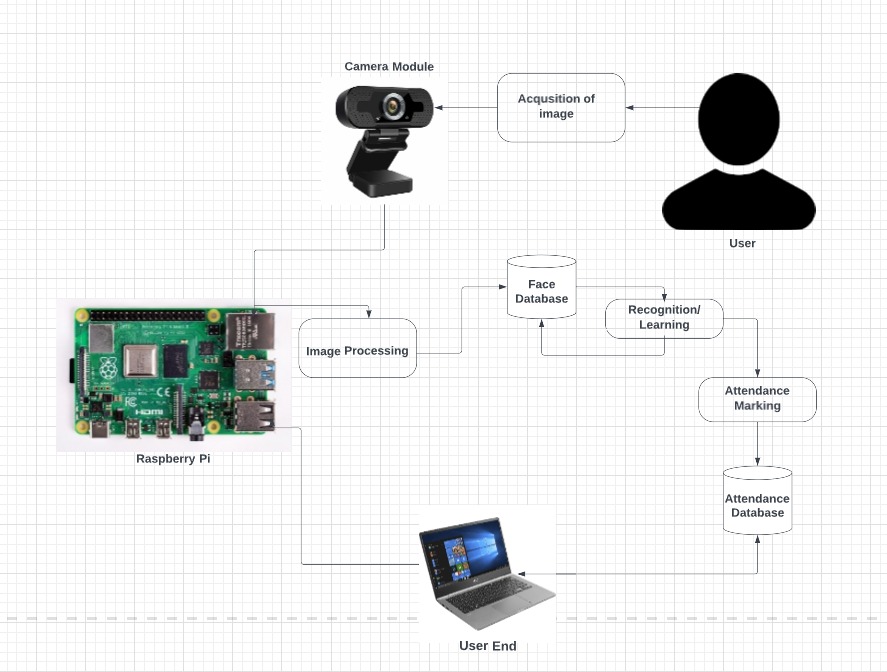
**Fig. 2.26**

**2.15.2 Student State Transition**



**Fig. 2.27**

**2.16 System Architecture Diagram:**



**Fig. 2.28**

**2.17 Machine Learning**

Machine learning (ML) is a field of inquiry devoted to understanding and building methods that ‘learn’, that is methods that leverage data to improve performance on some set of tasks. It is a branch of artificial intelligence (AI) and computer science, which focuses on the use of the data and algorithms to imitate the way that human learn, gradually improving its accuracy. ML algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in speech recognition, email filtering, face recognition and much more.

**2.18 Face Recognition**

Face recognition is a method of identifying or verifying the identity of an individual using their face. Face recognition system is used to identify people in photos, videos, or in real-time. Here we give the system a set of images of a person, and try to recognize the person by comparing the image from the database.

**2.19 Face Recognition Based Attendance System**

A face recognition attendance system incorporates face recognition technology to recognize and verify an student’s or person’s facial features and to record attendance automatically. A face recognition attendance system is an non-contact approach to managing students attendance.

**2.20 Working of Face Recognition Module**

1. First, the image is captured from a webcam or camera.
2. Eye locations are determined.
3. Then the image is converted to grayscale and cropped and converted to a template used by the search engine for facial comparison results.
4. Image is searched and matched using an algorithm to compare the template to other images in the database.

**2.21 Modules We Used**

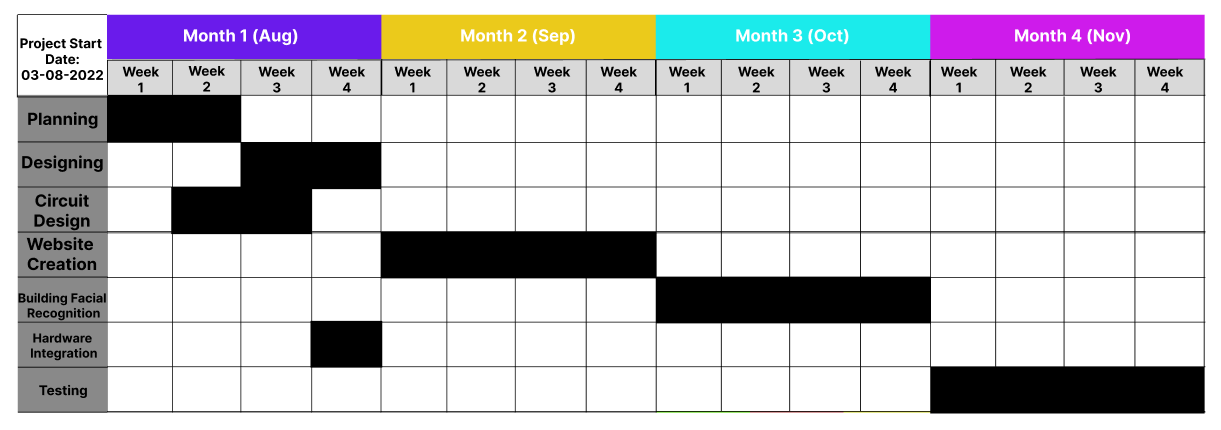
**2.22 Test Cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.No** | **Test case** | **Expected result** | **Steps** | **Outcome** |
| **1** | **Registration(User Data)** | **Details successfully stored** | **Fill out the details in the registration page.** | **Success** |
| **2** | **Login(User data)** | **Retrieving existing account** | **Fill out the details in the login page.** | **Success** |
| **3** | **Give attendance**  **(Manual)** | **Attendance should be marked** | **Enter the students credentials** | **Success** |
| **4** | **View attendance** | **Attendance should be displayed** | **Login and view the attendance page.** | **Success** |
| **5** | **Facial recognition** | **Faces should be correctly identified.** | **Program should run automatically** | **Success** |
| **6** | **Recognition capacity** | **Should be able to recognize 10 faces.** | **Program should run automatically** | **Failed, recognizing only 5 faces.** |
| **7** | **Attendance marking (Auto)** | **Attendance should be marked automatically** | **Program should run automatically** | **Success** |
| **8** | **Module Access testing** | **Various modules such as Admin, Faculty should work as intended** | **Testing Each module with respect to it’s functions** | **Success** |
| **9** | **Security** | **The security features such as encryption should work as planned** | **Testing image encryption and database security** | **Success** |
| **10** | **Usability** | **All links and buttons should work as expected** | **Conducting a level 1 testing on the system wrt buttons and links** | **Success** |
|  |  |  |  |  |

1. **System Planning**

**3.1 Gantt Chart:**

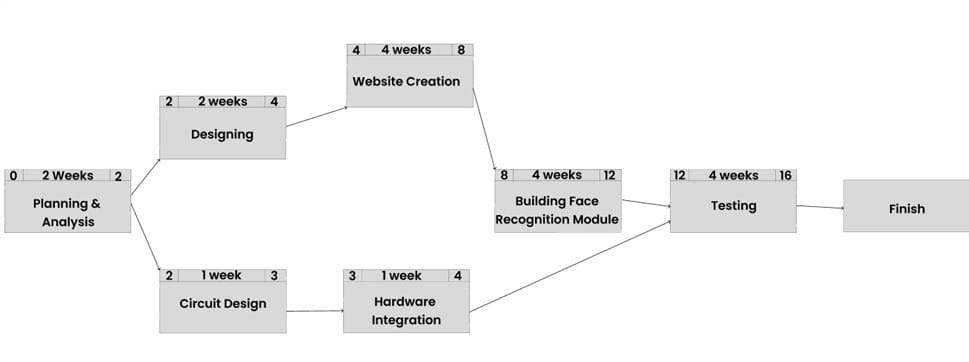
4 Months Activity

****

**Fig. 3.1**

**3.2 Activity Chart Diagram:**

Project Completion Estimate Time: 16 Weeks



**Fig. 3.2**

1. **Cost and Benefit Analysis and Software Parameter Estimation**

As defined, “Cost Estimation is a statement that gives the value of the cost incurred in the manufacturing of finished goods. Cost estimation helps in fixing the selling price of the final product after charging appropriate overheads and allowing a certain margin for profits.”

There are 5 Functional units used to calculate Function Point (FP):

1. Internal Logic Files (**ILF**): To control the information or logically related data that is present within the system.

2. External Interface Files (**EIF**): The control data referenced by the system but present in another system.

3. External Inputs (**EI**): Data/ control info that comes from outside our system.

4. External Outputs (**EO**): Data that goes out of the system after generation.

5. External Inquires (**EQ**): Combination of input-output resulting data retrieval.

To Compute FP:

We’ll use, **FP** = **UFP** \* **CAF**

Where, **UFP** = Unadjusted Function Point &

**CAF** = Complexity Adjustment Factor

Step 1: Calculate **UFP**

To find **UFP** we need to sum all the Complexities of all the **EI**, **EO**, **EQ**, **ILF** and **EIF**.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function Unit | Weighting Factors | | | | | |
| Count |  | Low | Average | High |  |
| External Inputs (EI) | 12 | \* | 3 | 4 | 6 | 48 |
| External Outputs (EO) | 10 | \* | 4 | 5 | 7 | 50 |
| External Inquires (EQ) | 14 | \* | 3 | 4 | 6 | 56 |
| Internal Logic Files (ILF) | 4 | \* | 7 | 10 | 15 | 40 |
| External Interface Files (EIF) | 14 | \* | 5 | 7 | 10 | 98 |
| Total Count | 292 | | | | | |

**Table: 7**

Step 2: Calculating **CAF**:

Formula: **CAF** = 0.65 + (0.01 \* **DI**)

Where, **DI** = Value adjustment factors based on responses to the following 14 questions

|  |  |  |
| --- | --- | --- |
| 1 | Data Communication | 5 |
| 2 | Distributed Data Processing | 4 |
| 3 | Performance Criteria | 4 |
| 4 | Heavily Utilized Hardware | 5 |
| 5 | Online Data Entry | 5 |
| 6 | High Transaction Rate | 0 |
| 7 | Online Updating | 2 |
| 8 | End-user Efficiency | 5 |
| 9 | Complex Computations | 4 |
| 10 | Reusability | 2 |
| 11 | Ease of Installation | 2 |
| 12 | Ease of Operation | 3 |
| 13 | Portability | 2 |
| 14 | Maintainability | 4 |
| Degree of Influence (DI) | | 47 |

**Table: 8**

Step 3: Calculating **Function Point**

Function Point (FP) = UFP\*(0.65+0.01\*DI)

Function Point (FP) = 292\*(0.65+0.01\*47)

Function Point (FP) = 292\*1.12

Function Point (FP) = 327.04

That Means, Function Point is **327.04**

1. **Future Work**

The same system can be utilized for several security applications where authentication is needed to access the privileges of the respective system. It can be used in recognizing guilty parties involving in unauthorized business. Face recognition algorithm can be improved with respect to the utilization of resources so that the project can recognize a greater number of faces at a time which can make the system far better. Many variants of the project can be developed and utilized for home security and personal or organizational benefits. We can also trace a particular student in an organization quickly with the help of this system. Identification using Facial recognition is a powerful tool. It can be used as means of authentication for not only attendance but a vast number of applications in almost every field. The system can be utilized for several security applications where authentication is needed to access the privileges of the respective system. Assume a system or a research facility containing confidential data that is only meant to be seen by certain people. Traditionally an id/password is issued for security and authentication however if someone gains access to these, the system is compromised and it is relatively easy to gain access to. Now, replacing these same id/passwords with facial recognition makes the authentication much more secure. In order to gain access to the system the hacker would have to steal a person’s face, which as of yet is a rather difficult task. Hence this system can be used in authentication processes to identify personnel that can and can’t access a certain resource or system. This however requires a more dynamic facial recognition algorithm one that can differentiate between a picture and person. Moreover, the system would need to adapt to changes in a person’s faces such as aging or facial hair etc. It would require a more sophisticated system with multiple cameras and different sensors such as infrared sensor. Every minute a child gets abducted, a car gets stolen, someone breaks the law, some of these are minor crimes but some of these are major crimes such as abduction of a child and is particularly time sensitive. In a similar security environment, the system can be deployed in public areas to help identify a person that has gone missing. The system would simply keep scanning every face that passes in front of it. When it finds a suspect, it can send an alert with the location and time to the authorities. This makes it much more likely to catch the suspect quickly with very low chances of mistaking the person for someone else. Similarly, it can be used to find a missing person. Even if say a person particularly a kid gets lost, the kid can simply stand in front of a system camera and the camera will scan the person send an alert and the person can be easily found and quickly. If the person that’s lost hasn’t been registered as lost yet, the system would just capture their face anyways and store it with location and time. When their face is searched in the database. The most recent image with time and location pops up, this would help the authorities to narrow down their search area and the person could be found quicker as the authorities would have some idea where the person could be. Another upcoming area are authenticating or registering emotions using facial recognition also known as Facial Emotion Recognition (FER) Facial Emotion Recognition (from real-time or static images) is the process of mapping facial expressions to identify emotions such as disgust, joy, anger, surprise, fear, or sadness - or compound emotion such as sadly angry - on a human face with image processing software. Facial emotion detection's popularity comes from the vast areas of potential applications. It’s different from facial recognition, whose goal is to identify a person, not an emotion. Face expression may be represented by geometric or appearance features, parameters extracted from transformed images such as eigenfaces, dynamic models, and 3D and models. Integrating the system with home security systems can be an excellent addition to automation and security. Imagine the owner forgets the key or pin, they can simply open the door with their face. If the owner is expecting a delivery say from amazon, a system could be put in place that sends a picture of the delivery person to the home security system. The system then creates a onetime access pass for the person to enter the house keep the item and exit. This could be timed for further security. The person has to exit in 2mins or the system sounds alarms. This would also come into effect if the owner is expecting company. The owner can register them into the system and the door will open for them upon recognition. This can be configured as one time, multi time, multi time during a specific time period, etc.

1. **Reference**
2. Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, pp. I-511). IEEE.
3. NirmalyaKar, MrinalKantiDebbarma, AshimSaha, and DwijenRudra Pal, “Implementation of Automated Attendance System using Face Recognition”, International Journal of Computer and Communication Engineering, Vol. 1, No. 2, July 2012.
4. Soundrapandiyan Rajkumar, J. Prakash, “Automated attendance using Raspberry pi”, International Journal of Pharmacy and Technology, Sep 2016.
5. *An Integrated Approach to Software Engineering* by Pankaj Jalote, 3rd Ed.
6. <https://www.academia.edu/52121795/Face_Recognition_Attendance_System_using_Raspberry_Pi> - (PDF) Face Recognition Attendance System using Raspberry Pi | IJRASET Publication.
7. <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3851342> - Face Recognition System Attendance System using Raspberry Pi by Evanjalin A.B., Christy D., Karthika N., Reshma R.S.: SSRN.