Introduction of Project

Today one cannot afford to rely on the fallible human beings of be really wants to stand against today's merciless competition where not to wise saying "to err is human" no longer valid, it's outdated to rationalize your mistake. So, to keep pace with time, to bring about the best result without malfunctioning and greater efficiency so to replace the unending heaps of flies with a much-sophisticated hard disk of the computer.

One has to use the data management software. Software has been an ascent in atomization various organizations. Many software products working are now in markets, which have helped in making the organizations work easier and efficiently. Data management initially had to maintain a lot of ledgers and a lot of paperwork has to be done but now software product on this organization has made their work faster and easier. Now only this software has to be loaded on the computer and work can be done.

This prevents a lot of time and money. The work becomes fully automated and any information regarding the organization can be obtained by clicking the button. Moreover, now it's an age of computers of and automating such an organization gives the better look.

Objective of the project

Data is gold in today's world. In any form it is very useful and can have many benefits. But if data is not organized what is the use of it. It would become like old days where registers were kept in which data was stored and to search for a particular record would take days or even months. This project is a small step towards stopping that old practice and start using new technologies to make work easier.

Designed for schools in mind, this is a Hostel-Management System. A system to remove all the other registers or many excel sheets and give one file. This system is fully digitalized and can search for records, add new one, delete records, change records in seconds and without any duplication. All the user will have to do is click some buttons and type some numbers. Many projects like this one exists but this is specially made to cater the needs of schools.

This also allowed us students to use our practical skills and knowledge in computer science to do the following:

1. Write programs utilizing modern software tools.
2. Apply object-oriented programming principles effectively when developing small to medium sized projects.
3. Write effective procedural code to solve small to medium sized problems.
4. Students will demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory and software development.
5. Students will demonstrate ability to conduct a research or applied Computer Science project, requiring writing and presentation skills which exemplify scholarly style in computer science.

Introduction of Python

Python is a high-level general purpose programming language that is used in a wide variety of application domains. Python has the right combination of performance and features that demystify program writing.

Python is dynamically –typed and garbage – collected. It supports Multiple programming including structured (Particularly, Procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Python is an easy to learn language with support from all over the world. It is also an open sourced free to use language even for commercial purposes. It is a cross-platform language therefore no particular OS required. It is an interpreted object orients language which can also be embedded in applications as scripting interface.

The advantages of using python are that it has a smooth learning curve, is versatile with an extensive toolset and can be easily teamed up with other languages. The disadvantages are that it is not fast, has no multithreading, has high memory consumption and hassles with mobile and front–end development.

Introduction to MySQL

- MySQL is a database system used for developing web-based software applications.
- MySQL used for both small and large applications.
- MySQL is a relational database management system (RDBMS).
- MySQL is fast, reliable, and flexible and easy to use.
- MySQL supports standard SQL (Structured Query Language).
- MySQL is free to download and use.
- MySQL was developed by Michael Widenius and David Axmark in 1994.
- MySQL is presently developed, distributed, and supported by Oracle Corporation.
- MySQL Written in C, C++

Features:

- MySQL server design is multi-layered with independent modules.
- MySQL is fully multithreaded by using kernel threads. It can handle multiple CPUs if they are available.
- MySQL provides transactional and non-transactional storage engines.
- MySQL has a high-speed thread-based memory allocation system.
- MySQL supports in-memory heap table.
- MySQL Handles large databases.
- MySQL Server works in client/server or embedded systems.
- MySQL Works on many different platforms

Famous users:

- Some of the most famous websites like Facebook, Wikipedia, Google (not for search), YouTube, Flickr.
- Content Management Systems (CMS) like WordPress, Drupal, Joomla, phpBB etc.
- A large number of web developers worldwide are using MySQL to develop web applications

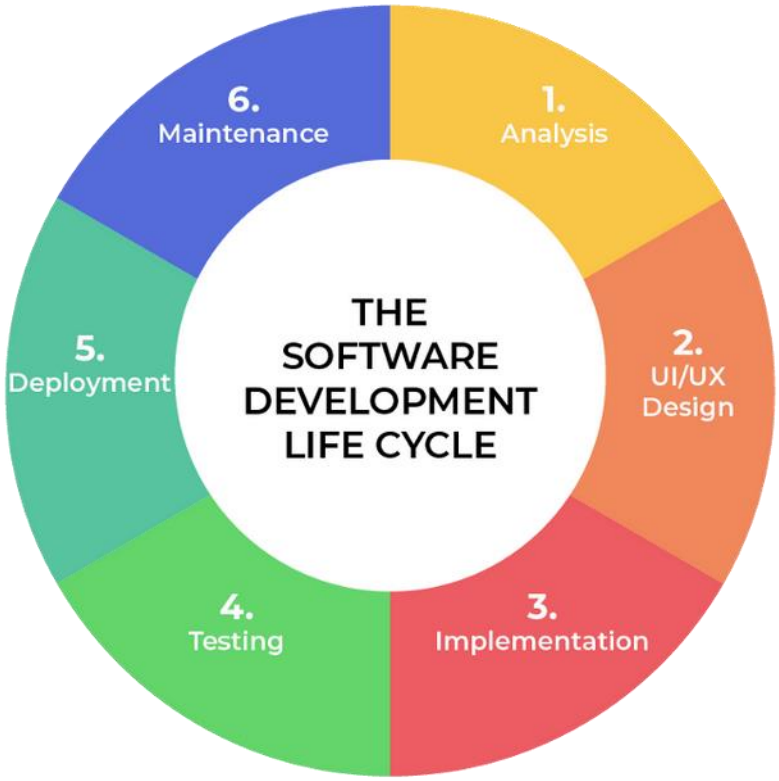
PROPOSED SYSTEM

Today one cannot afford to rely on the fallible human beings of be really wants to stand against today's merciless competition where not to wise saying "to err is human" no longer valid, it's out-dated to rationalize your mistake. So, to keep pace with time, to bring about the best result without malfunctioning and greater efficiency so to replace the unending heaps of flies with a much sophisticated hard disk of the computer.

One has to use the data management software. Software has been an ascent in atomization various organisations. Many software products working are now in markets, which have helped in making the organizations work easier and efficiently. Data management initially had to maintain a lot of ledgers and a lot of paperwork has to be done but now software product on this organization has made their work faster and easier. Now only this software has to be loaded on the computer and work can be done.

This prevents a lot of time and money. The work becomes fully automated and any information regarding the organization can be obtained by clicking the button. Moreover, now it's an age of computers of and automating such an organization gives the better look.

SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)



The systems development life cycle is a project management technique that divides complex projects into smaller, more easily managed segments or phases. Segmenting project sallow managers to verify the successful completion of project phases before allocating resources to subsequent phases.

Software development projects typically include initiation, planning design development, testing, implementation and maintenance phases. However, the phases may be divided differently depending on the organization involved.

For example, initial project activities might be designated as request, requirements-definition, and planning phases, or initiation, concept-development, and planning phases. End users of the system under development should be involved the output of each phase to ensure the system is being built to deliver the needed functionality

PHASES OF SYSTEM DEVELOPMENT LIFE CYCLE

INITIATION PHASE

The Initiation Phase begins when a business sponsor identifies a need or an opportunity.

The purpose of the Initiation Phase is to:

- Identify and validate an opportunity to improve business accomplishments of the organization or a deficiency related to a business need.
- Identify significant assumptions and constraints on solutions to that need.
- Recommend the exploration of alternative concepts and methods to satisfy the need including questioning the need for technology, i.e., will a change in the business process offer a solution?
- Assure executive business and executive technical sponsorship. The Sponsor designates a Project Manager and the business need is documented in a Concept Proposal. The Concept Proposal includes information about the business process and the relationship to the Agency/Organization.
- Infrastructure and the Strategic Plan. A successful Concept Proposal results in a Project Management Charter which outlines the authority of the project manager to begin
  the project

Careful oversight is required to ensure projects support strategic business objectives and resources are effectively implemented into an organization's enterprise architecture. The initiation phase begins when an opportunity to add, improve, or correct a system is identified and formally requested through the presentation of a business case. The business case should, at a minimum, describe a proposal's purpose, identify expected benefits, and explain how the proposed system supports one of the organization's business strategies. The business case should also identify alternative solutions and detail as many informational, functional, and network requirements as possible.

SYSTEM CONCEPT DEVELOPMENT PHASE

The System Concept Development Phase begins after a business need or opportunity is validated by the Agency/Organization Program Leadership and the Agency/Organization CIO.

The purpose of the System Concept Development Phase is to:

- Determine the feasibility and appropriateness of the alternatives.
- Identify system interfaces.
- Identify basic functional and data requirements to satisfy the business need.
- Establish system boundaries; identify goals, objectives, critical success factors, and performance measures.
- Evaluate costs and benefits of alternative approaches to satisfy the basic functional requirements
- Assess project risks
- Identify and initiate risk mitigation actions, and Develop high-level technical architecture, process models, data models, and a concept of operations. This phase explores potential technical solutions within the context of the business need.
- It may include several trade-off decisions such as the decision to use COTS software products as opposed to developing custom software or reusing software components, or the decision to use an incremental delivery versus a complete, onetime deployment.
- Construction of executable prototypes is encouraged to evaluate technology to support the business process. The System Boundary Document serves as an important reference document to support the Information Technology Project Request (ITPR) process.
- The ITPR must be approved by the State CIO before the project can move forward.

PLANNING PHASE

The planning phase is the most critical step in completing development, acquisition, and maintenance projects. Careful planning, particularly in the early stages of a project, isnecessary to coordinate activities

and manage project risks effectively. The depth and formality of project plans should be commensurate with the characteristics and risks of a given project. Project plans refine the information gathered during the initiation phase by further identifying the specificactivities and resources required to complete a project.

A critical part of a project manager'sjob is to coordinate discussions between user, audit, security, design, development, and network personnel to identify and document as many functional, security, and networkrequirements as possible. During this phase, a plan is developed that documents the approach to be used and includes a discussion of methods, tools, tasks, resources, project schedules, and user input. Personnel assignments, costs, project schedule, and target dates are established.

A Project Management Plan is created with components related to acquisition planning, configuration management planning, quality assurance planning, concept of operations, system security, verification and validation, and systems engineering management planning.

REQUIREMENTS ANALYSIS PHASE

This phase formally defines the detailed functional user requirements using high-level requirements identified in the Initiation, System Concept, and Planning phases. It also delineates the requirements in terms of data, system performance, security, and maintainability requirements for the system. The requirements are defined in this phase to a level of detail sufficient for systems design to proceed. They need to be measurable, testable, and relate to the business need or opportunity identified in the Initiation Phase. The requirements that will be used to determine acceptance of the system are captured in the Test and Evaluation Master Plan.

The purposes of this phase are to:

- Further define and refine the functional and data requirements and document them in the Requirements Document,

- Complete business process reengineering of the functions to be supported (i.e., verify what information drives the business process, what information is generated, who generates it, where does the information go, and who processes it),
- Develop detailed data and process models (system inputs, outputs, and the process.
- Develop the test and evaluation requirements that will be used to determine acceptable system performance**.**

DESIGN PHASE

The design phase involves converting the informational, functional, and network requirements identified during the initiation and planning phases into unified design specifications that developers use to script programs during the development phase. Program designs are constructed in various ways. Using a top-down approach, designers first identify and link major program components and interfaces, then expand design layouts as they identify and link smaller subsystems and connections. Using a bottom-up approach, designers first identify and link minor program components and interfaces, then expand design layouts as they identify and link larger systems and connections. Contemporary design techniques often use prototyping tools that build mock-up designs of items such as application screens, database layouts, and system architectures. End users, designers, developers, database managers, and network administrators should review and refine the prototyped designs in an iterative process until they agree on an acceptable design. Audit, security, and quality assurance personnel should be involved in the review and approval process. During this phase, the system is designed to satisfy the functional requirements identified in the previous phase. Since problems in the design phase could be very expensive to solve in the later stage of the software development, a variety of elements are considered in the design to mitigate risk. These include:

- Identifying potential risks and defining mitigating design features.
- Performing a security risk assessment.
- Developing a conversion plan to migrate current data to the new system.
- Determining the operating environment.
- Defining major subsystems and their inputs and outputs.
- Allocating processes to resources.

- Preparing detailed logic specifications for each software module. The result is a draft System Design Document which captures the preliminary design for the system.

- Everything requiring user input or approval is documented and reviewed by the user. Once these documents have been approved by the Agency CIO and Business Sponsor, the final System Design Document is created to serve as the Critical/Detailed Design for the system.

- This document receives a rigorous review by Agency technical and functional representatives to ensure that it satisfies the business requirements. Concurrent with the development of the system design, the Agency Project Manager begins development of the Implementation Plan, Operations and Maintenance Manual, and the Training Plan.

DEVELOPMENT PHASE

The development phase involves converting design specifications into executable programs. Effective development standards include requirements that programmers and other project participants discuss design specifications before programming begins. The procedures help ensure programmers clearly understand program designs and functional requirements. Programmers use various techniques to develop computer programs. The large transaction oriented programs associated with financial institutions have traditionally been developed using procedural programming techniques. Procedural programming involves the line-by-line scripting of logical instructions that are combined to form a program. Effective completion of the previous stages is a key factor in the success of the Development phase. The Development phase consists of:

- Translating the detailed requirements and design into system components.
- Testing individual elements (units) for usability.
- Preparing for integration and testing of the IT system.

INTEGRATION AND TEST PHASE

- Subsystem integration, system, security, and user acceptance testing is conducted during the integration and test phase. The user, with those responsible for quality assurance, validates that the functional requirements, as defined in the functional requirements document, are satisfied by the developed or modified system. OIT Security staff assesses the system security and issue a security certification and accreditation prior to installation/implementation. –

Multiple levels of testing are performed, including:

- Testing at the development facility by the contractor and possibly supported by end users
- Testing as a deployed system with end users working together with contract personnel
- Operational testing by the end user alone performing all functions. Requirements are traced throughout testing, a final Independent Verification & Validation evaluation is performed and all documentation is reviewed and accepted prior to acceptance of the system.

IMPLEMENTATION PHASE

This phase is initiated after the system has been tested and accepted by the user. In this phase, the system is installed to support the intended business functions. System performance is compared to performance objectives established during the planning phase. Implementation includes user notification, user training, installation of hardware, installation of software onto production computers, and integration of the system into daily work processes. This phase continues until the system is operating in production in accordance with the defined user requirements.

OPERATIONS AND MAINTENANCE PHASE

The system operation is on-going. The system is monitored for continued performance in accordance with user requirements and needed system modifications are incorporated. Operations continue as long as the system can be effectively adapted to respond to the organization's needs. When modifications or changes are identified, the system may re-enter the planning phase.

The purpose of this phase is to:

- Operate, maintain, and enhance the system.
- Certify that the system can process sensitive information.
- Conduct periodic assessments of the system to ensure the functional requirements continue to be satisfied.
- Determine when the system needs to be modernized, replaced, or retired.

Algorithm

Step- 1: Run the Hostel management software

Step- 2: Login as Administrator

Step- 3: If Login:

Choose:

1. Add student data:
2. Modify student data:
3. Search for student data:
4. See students of a hostel:
5. Delete student data:

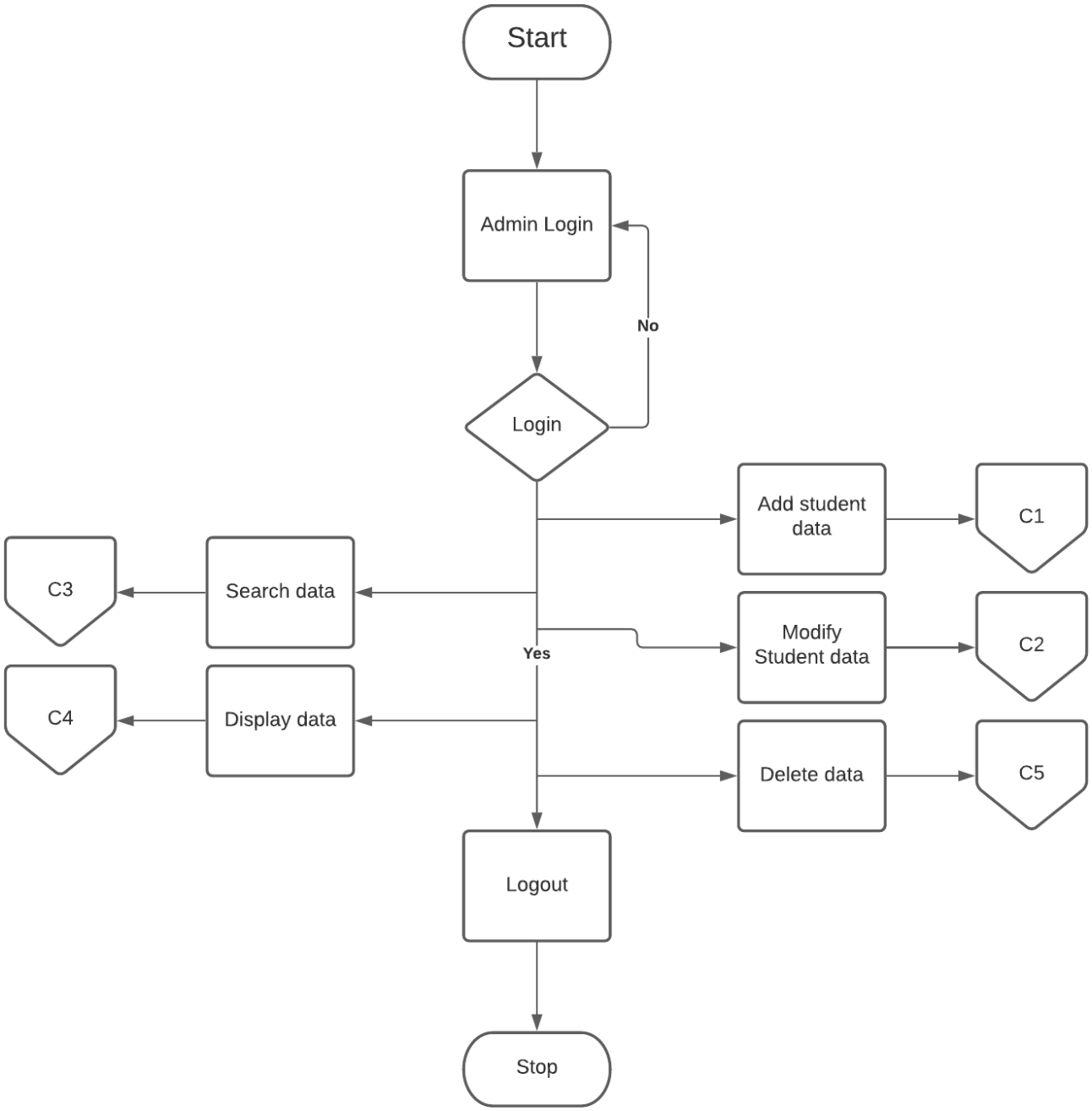
If not login:

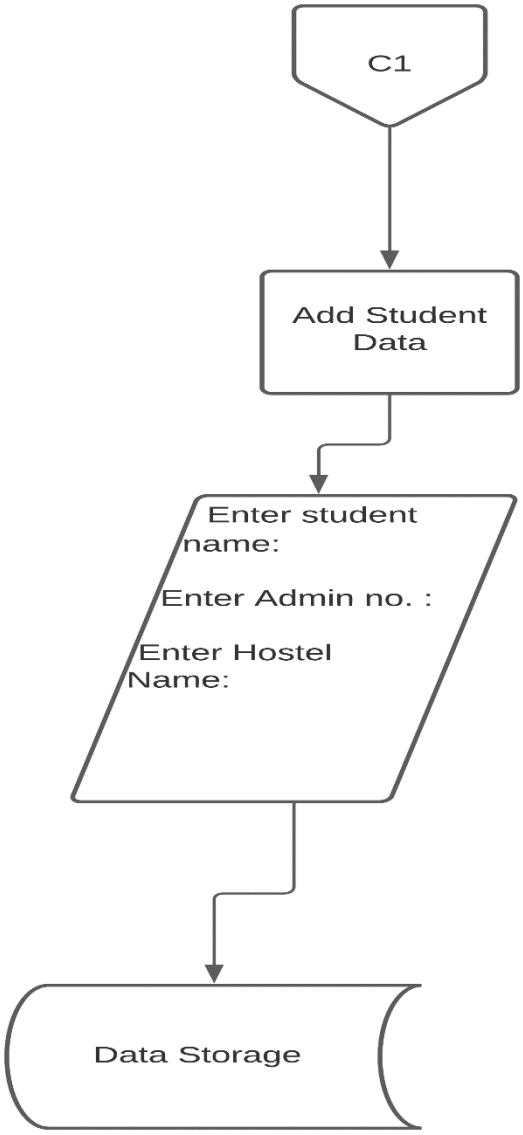
Try again

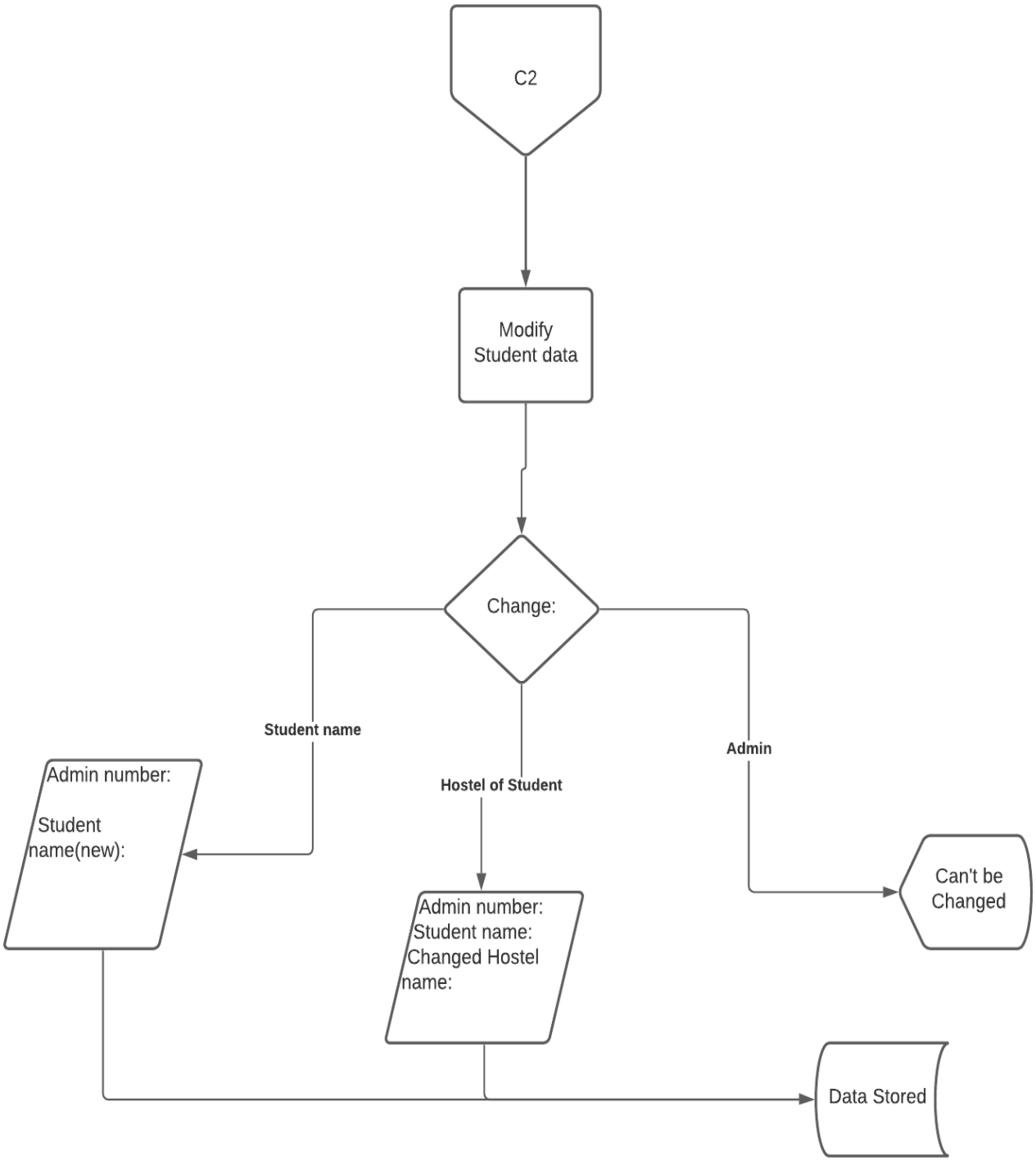
Step- 4: Admin Logout

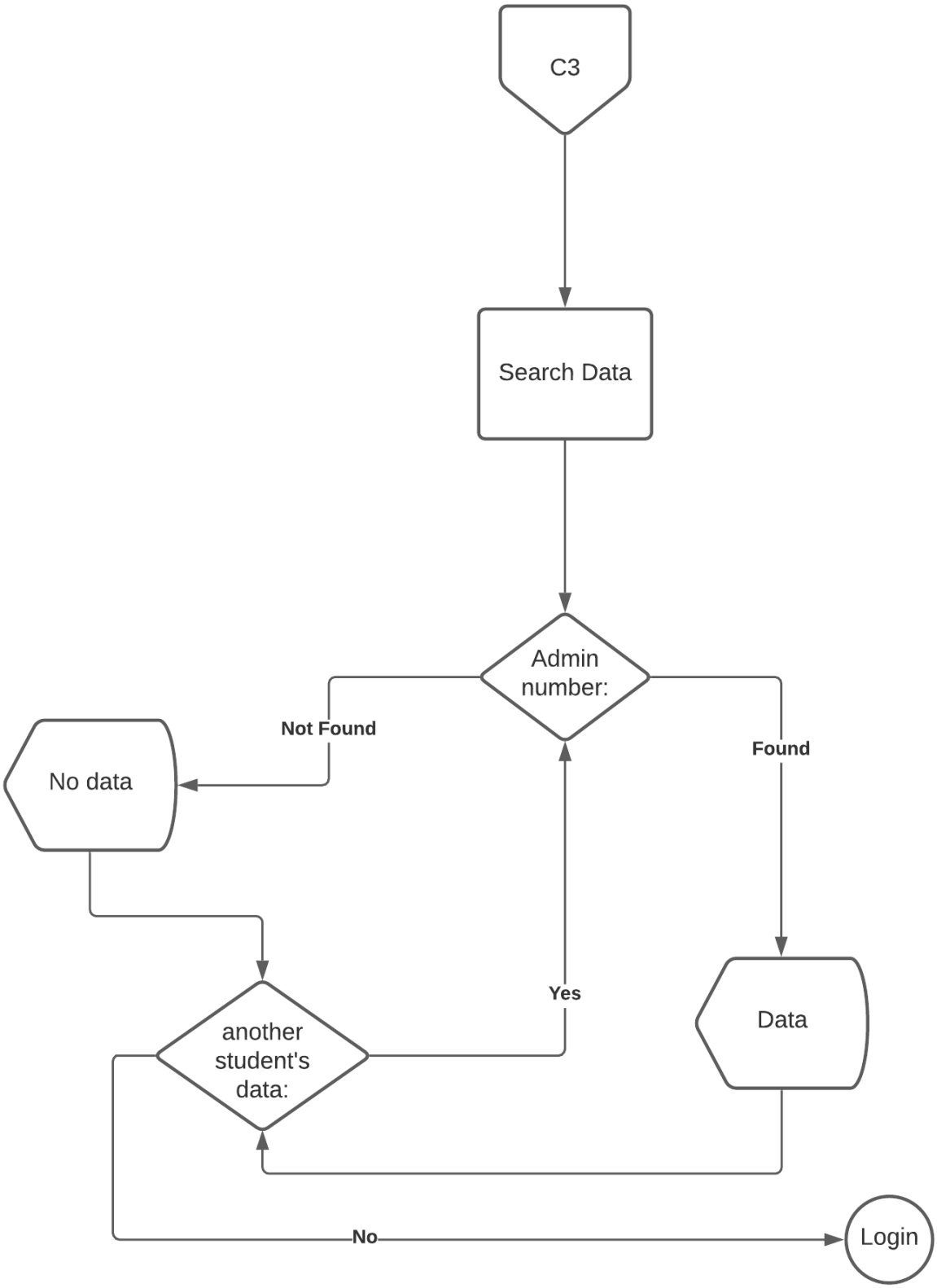
Step- 5: Stop

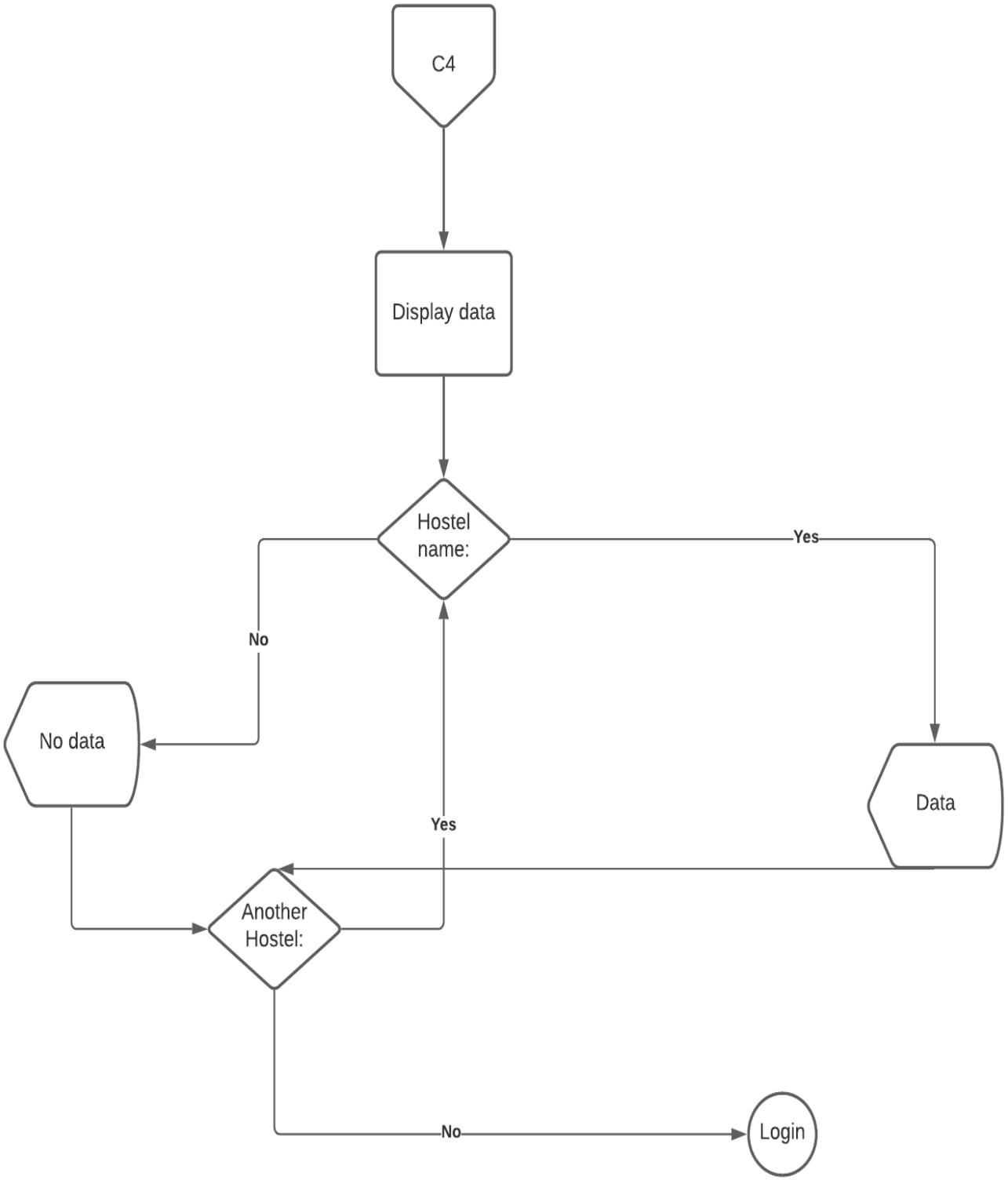
FLOW CHARTS

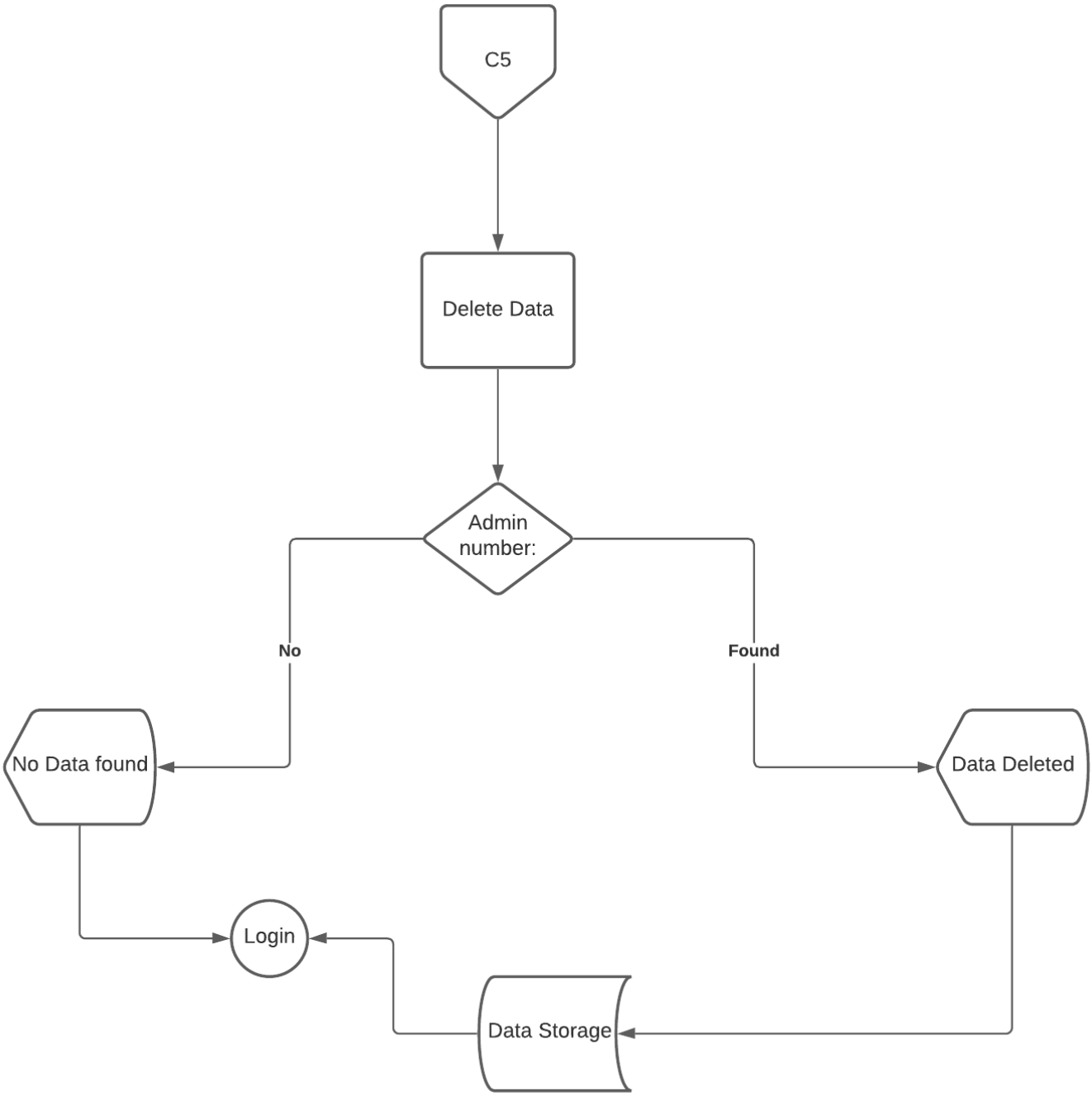

Start
Admin Login
No
Login
Add student data
C1
C3
Search data
Modify Student data
C2
Yes
C4
Display data
Delete data
C5
Logout
Stop


C1
Add Student Data
Enter student name:
Enter Admin no. :
Enter Hostel Name:
Data Storage


C2
Modify Student data
Change:
Student name
Hostel of Student
Admin
Admin number:
Student name(new):
Admin number:
Student name:
Changed Hostel name:
Can't be Changed
Data Stored


C3
Search Data
Admin number:
Not Found
Found
No data
another student's data:
Yes
Data
No
Login


C4
Display data
Hostel name:
Yes
No
No data
Data
Yes
Another Hostel:
No
Login


C5
Delete Data
Admin number:
No
Found
No Data found
Data Deleted
Login
Data Storage

SOURCE CODE

1. Backend- programs that will run in the back and user will not see

a. Startup- Main Backend Module can be used to run first time

Nameandpass.py

```
import csv
def takename_passwd():
    with open('npassformysql.csv','w',newline='') as file:
        wr = csv.writer(file)
        name = 'root'
        passwd = 'root'
        head = ['Name', 'Password']
        data = []
        data.append(name)
        data.append(passwd)
        wr.writerow(head)
        wr.writerow(data)

takename_passwd()
with open('npassformysql.csv','r') as file:
    rd = csv.reader(file)
    data = [x for x in rd]
    name = data[1][0]
    passwd = data[1][1]
```

Startup.py

```
import mysql.connector as mc

from nameandpass import name,passwd

import startup2

db = mc.connect(

    host = 'localhost',
```

```
    user = name,
    passwd = passwd
)
myc = db.cursor(buffered=True)
try:
    myc.execute('CREATE DATABASE Hostels_Management')
    startup2.run()
except Exception:
    startup2.run()
```

Startup2.py

```
import mysql.connector as mc
from nameandpass import name,passwd
import startup3
def run():
    db = mc.connect(
        host = 'localhost',
        user = name,
        passwd = passwd,
        database = 'Hostels_Management'
    )
    myc = db.cursor(buffered=True)
```

```
    def createtables():
        myc.execute(
            "CREATE TABLE Students(Adminno int PRIMARY KEY,Name varchar(50),Age int UNSIGNED,Class smallint UNSIGNED,Sex char(10),Father_name varchar(100), Mother_name varchar(100), DOB Varchar(15),Phone_number bigint UNSIGNED,Email varchar(100), Hostel_ID int)")
        myc.execute("CREATE TABLE Hostels(Hostel_ID int PRIMARY KEY, Name varchar(20))")
    def add_data_hostels():
        while True:
            hostelid = int(input("Enter the hostel id: "))
            name = input("Enter Hostel name(Please refrain from using spaces. use underscore(_) instead): ")
            myc.execute(f"INSERT INTO Hostels(Hostel_ID,Name) values(%s,%s)", (hostelid, name))
            db.commit()
            c = input('Another data(Y/n): ').lower()
            if c == 'n':
                break
    def create_tables_hostels():
        myc.execute("SELECT * from hostels")
        hostels_names = [x[1] for x in myc]
        for na in hostels_names:
```

```
            myc.execute(f"CREATE TABLE {na}(Adminno int PRIMARY
KEY,Student_name varchar(50),Age smallint ,Class smallint ,Gender
char(10),Hostel_ID int)")
        db.commit()
    def checkup():
        check = input('Enter Data changed(Y/n): ').lower()
        if check == 'y':
            startup3.run()
        else:
            print("Please change the data")
            checkup()
    try:
        createtables()
        try:
            add_data_hostels()
            create_tables_hostels()
            print(
                'Please make sure that hostel id field in the csv
file has only the numbers you have entered now. If not please change
it')
            checkup()
        except:
            try:
```

```
                create_tables_hostels()

                print(

                    'Please make sure that hostel id field in the
csv file has only the numbers you have entered now. If not please
change it')

                checkup()

            except:

                print(

                    'Please make sure that hostel id field in the
csv file has only the numbers you have entered now. If not please
change it')

                checkup()

    except Exception:

        print('Tables Already created.')

        myc.execute('SELECT Hostel_ID FROM Hostels')

        hostelids = [x[0] for x in myc]

        print(hostelids)

        print('Please make sure that hostel ids in the
studentsfakedata.csv file is in the range of number that are printed
above')

        checkup()
```

Startup3.py

```
import mysql.connector as mc
from nameandpass import name,passwd
import pandas as pd

def run():
```

```
    db = mc.connect(
        host = 'localhost',
        user = name,
        passwd = passwd,
        database = 'Hostels_Management'
    )

    myc = db.cursor(buffered=True)



    def add_data_students():
        studata =
pd.read_csv(r'Studentsfakedata.csv',index_col=False,
delimiter=',')
        studata.head()
        for i,row in studata.iterrows():
            myc.execute(f"INSERT INTO
Students(Adminno,Name,Age,Class,Sex,Father_name,Mother_name,DOB,P
hone_number,Email,Hostel_Id)
VALUES(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)",tuple(row))
            db.commit()

    def add_data_hotels_students():
        myc.execute("SELECT * from hostels")
        hostels_names = [x[1] for x in myc]
        myc.execute("SELECT * from hostels")
        hostels_id = [x[0] for x in myc]
        for i in hostels_id:
            myc.execute(f"SELECT Adminno,
Name,Age,Class,Sex,Hostel_ID FROM students where Hostel_ID =
{i}")
            name = hostels_names[i-1]
            data = myc.fetchall()
            for row in data:
                myc.execute(f"INSERT INTO
{name}(Adminno,Student_name,Age,Class,Gender,Hostel_ID)
VALUES(%s,%s,%s,%s,%s,%s)",row)
                db.commit()

    try:
        try:
            add_data_students()
            try:
                add_data_hotels_students()
                print('Data Entered')
            except:
                print('Data Already Entered')
        except:
            try:
                add_data_hotels_students()
                print('Data Entered')
```

```
            except:
                print('Data Already entered')
    except Exception:
        try:
            add_data_students()
            try:
                add_data_hotels_students()
                print('Data Entered')
            except:
                print('Data Already Entered')
        except:
            try:
                add_data_hotels_students()
                print('Data Entered')
            except:
                print('Data Already entered')
        print("Data Already Entered")

    print("Run Hostelmanagement.py.py")
```

b. Data File

Studentsfakedata.csv

Admin no.	Name	AGE	CLASS	GENDER	Father's Name	Mother's Name	DOB	ph no.	email	Hoste_Id
3678	Dean Crink	16	11	Male	Gale Crinkley	Sapphire Crinkley	12/14/2006	1978644547	gcrinkley0@devhub.com	10
2942	Dugald He	14	11	Male	Husain Henrique	Mariette Henrique	4/26/2008	4458234283	hhenrique1@networksolutions.com	7
2777	Forbes Hut	18	11	Male	Aron Hufton	Adella Hufton	9/9/2004	9429851377	ahufton2@ucla.edu	10
1873	Mic Niche	18	10	Male	Oberon Niche	Kirsti Niche	3/6/2004	8732046410	oniche3@bloomberg.com	9
1108	Fair Blackv	17	12	Female	Benjy Blackway	Dell Blackway	8/29/2005	1461912009	bblackway4@infoseek.co.jp	5
5973	Sandy Felip	17	12	Female	Hersh Felipe	Mercie Felipe	5/31/2005	2905802675	hfelipe5@freewebs.com	4
1252	Egor Prall	15	10	Male	Fred Prall	Becky Prall	9/21/2007	3944853338	fprall6@marriott.com	10
5940	Yancy Mee	16	11	Female	Archibald Meegin	Lynnette Meegin	5/16/2006	2633166406	ameegin7@cnet.com	3
2974	Wernher F	16	12	Male	Jamal Forster	Clo Forster	1/4/2006	3028090712	jforster8@vk.com	12
4022	Ezri Hoven	17	10	Female	Donalt Hovenden	Zilvia Hovenden	12/11/2005	3993547606	dhovenden9@statcounter.com	5
3339	Carny Tubr	15	11	Female	Isac Tubritt	Jolee Tubritt	4/9/2007	6072374540	itubritta@theglobeandmail.com	1
1115	Harlen Irvi	15	12	Female	Jeremiah Irvine	Riva Irvine	12/30/2007	6794425849	jirvineb@netvibes.com	5
1202	Marietta D	19	10	Female	Ailbert Duffit	Marnia Duffit	4/24/2003	7618147760	aduffitc@soundcloud.com	1
1520	Matt McN	19	10	Female	Yule McNiff	Gertrudis McNiff	6/27/2003	4593070599	ymcniffd@exblog.jp	3
2682	Chrotoem	15	10	Male	Wilfred Learmonth	Tomasina Learmonth	6/23/2007	6519339533	wlearmonthe@sciencedirect.com	7
4685	Krisha Dur	16	12	Female	Ethe Durak	Paola Durak	1/10/2006	3052838528	edurakf@flavors.me	6
3236	Read Conc	16	11	Male	Zane Conquer	Grace Conquer	4/2/2006	6092536822	zconquerg@nationalgeographic.com	7
3741	Lew Messu	16	11	Male	Clifford Messum	Kissie Messum	8/6/2006	4175228380	cmessumh@ask.com	11
5054	Dorey Seal	16	11	Male	Gaylord Sealeaf	Austin Sealeaf	2/8/2006	5002657023	gsealeafi@hc360.com	12
2429	Niko Buyer	16	12	Male	Evered Buyers	Sophronia Buyers	4/21/2006	6636844251	ebuyersj@ihg.com	10
2636	Philbert So	19	10	Female	Marven Sogg	Shanna Sogg	4/10/2003	5438037435	msoggk@businesswire.com	5
1280	Thorn Mar	17	11	Female	Craggy Mangeon	Janenna Mangeon	9/28/2005	1956841452	cmangeonl@studiopress.com	6
1385	Homer Gu	17	12	Female	Wolfgang Gumey	Ashlen Gumey	2/13/2005	2386044008	wgumeym@fastcompany.com	6
5255	Dalston Hu	15	12	Male	Steven Humbie	Raquel Humbie	6/21/2007	7059171308	shumbien@cbc.ca	12
2335	Garek Hen	15	12	Female	Ingar Henner	Kanya Henner	5/6/2007	7531200550	ihennero@fda.gov	2
2250	Anson Tre	19	10	Female	Bogey Trembey	Angelle Trembey	2/13/2003	3148417987	btrembeyp@zdnet.com	4
3709	Jedediah S	18	10	Male	Cazzie Stobo	Kylila Stobo	12/5/2004	5596634997	cstoboq@multiply.com	9
3046	Nilson Fre	19	11	Female	Conrad Freda	Janava Freda	5/13/2003	6896234699	cfredar@ning.com	1

More data like this is present

npassformysql.csv

Name	Password
root	root

2. Frontend

a. Hostel Management

```
import mysql.connector as mc
from nameandpass import name,passwd

db = mc.connect(
    host="localhost",
    user=name,
    passwd=passwd,
    database = 'hostels_Management'
)

myc = db.cursor(buffered=True)

def add_new_data():
        ask = input("Enter the name of the hostel in which you want
to add the student in: ")
        myc.execute("SELECT Hostel_ID FROM Hostels WHERE (Name =
%s) ",(ask,))
        hotelid = [x for x in myc]
        id = None
        for item in hotelid[0]:
            id = item
        adminno = int(input("Enter the admission number of the
student: "))
        name = input("Enter the name of the student: ")
        age = int(input("Enter the age of the student: "))
        classe = input("Enter the class of the student: ")
        gender    =    input("Enter    the    gender    of    the
student(Male,Female): ")
        query              =              f"INSERT              INTO
{ask}(Adminno,Student_name,Age,Class,Gender,Hostel_ID)
values(%s,%s,%s,%s,%s,%s)"
        myc.execute(query,(adminno,name,age,classe,gender,id))
        # print(id)

def delete():
    ask = input("Enter the name of the hostel you want to delete
data from: ")
    adminno= int(input("Enter the admin number of the student you
want to delete the data of: "))
    myc.execute(f"DELETE FROM {ask} WHERE adminno={adminno}")
    db.commit()

def modify_data():
    ask = input("enter the hostel name you want to change the data
of: ")
```

```
    how = int(input("How many columns do you want to modify: "))
    def modify(table_name):
        admi = int(input("Enter the admin number of the student you
want to change the data of: "))
        column_name = input("Enter the column name you want to
change the data of: ")
        data = input("What do you want to write: ")
        try:
            myc.execute(f"UPDATE {table_name} SET {column_name} =
%s WHERE adminno = {admi}",(data,))
            db.commit()
        except Exception:
            print("Wrong admin number or column name entered.
Please try again")
    for i in range(how):
        modify(ask)

def search():
    num = int(input("Enter the adminno of the student you want the
data of: "))
    try:
        myc.execute(f"SELECT Hostel_ID FROM Students WHERE Adminno
= {num}")
        id = [x for x in myc]
        query = "SELECT Name FROM Hostels WHERE (Hostel_ID = %s)"
        myc.execute(query,tuple(id[0]))
        hostel_name = [i for i in myc]
        hname = ''
        for item in hostel_name[0]:
            hname = hname + item
        def minisearch(table_name,val):
            myc.execute(f"SELECT   *   FROM   {table_name}   WHERE
(Adminno = %s)",(val,))
            header                                                =
('Adminno.','Name','Age','Class','Gender','HostelID')
            print(header)
            for y in myc:
             print(y)
        minisearch(hname,num)
    except Exception:
        print('The adminno. entered is not present')

def  display():
    head = ('HostelID','Hostel Name')
    print(head)
    myc.execute("SELECT * FROM Hostels")
    for i in myc:
        print(i)
    ask = input("Enter the name of the hostel you want to see the
data of (Don't use space use underscore(_)):  ")
    header  =  ('Adminno.',  'Name',  'Age',  'Class',  'Gender',
'HostelID')
```

```
    print(header)
    myc.execute(f"SELECT * FROM {ask}")
    for x in myc:
        print(x)


def ready():
    while True:
        ask = int(input("1. Add Student\n"
                "2. Search for a student\n"
                "3. Modify data\n"
                "4. Display data\n"
                "5. Delete data\n"
                "6.Exit\n"
                "Enter the option you want to select: "))
        if ask == 1:
            add_new_data()
        elif ask == 2:
            search()
        elif ask == 3:
            modify_data()
        elif ask == 4 :
            display()
        elif ask == 5:
            delete()
        elif ask == 6:
            break
        else:
            print("No recognizable option entered")

def login():
    aname = input("Enter the username of mysql: ")
    pd = input("Enter the password of the mysql: ")
    if (aname == name and pd == passwd):
        ready()
    else:
        print("Wrong password entered")


login()
```

OUTPUT

1. Backend

```
D:\Pranav\venv\Scripts\python.exe D:/Pranav/Startup.py
Enter the hostel id: 1
Enter Hostel name(Please refrainf from using spaces. use underscore(_) instead): North_House
Another data(Y/n): y
Enter the hostel id: 2
Enter Hostel name(Please refrainf from using spaces. use underscore(_) instead): South_House
Another data(Y/n): y
Enter the hostel id: 3
Enter Hostel name(Please refrainf from using spaces. use underscore(_) instead): East_House
Another data(Y/n): y
Enter the hostel id: 4
Enter Hostel name(Please refrainf from using spaces. use underscore(_) instead): West_House
Another data(Y/n): y
Enter the hostel id: 5
Enter Hostel name(Please refrainf from using spaces. use underscore(_) instead): RR_House
Another data(Y/n): y
Enter the hostel id: 6
Enter Hostel name(Please refrainf from using spaces. use underscore(_) instead): Gokul
Another data(Y/n): y
Enter the hostel id: 7
Enter Hostel name(Please refrainf from using spaces. use underscore(_) instead): River_Side_Hostel
Another data(Y/n): y
Enter the hostel id: 8
Enter Hostel name(Please refrainf from using spaces. use underscore(_) instead): Shanti_Vihar
Another data(Y/n): y
Enter the hostel id: 9
```

```
Enter Hostel name(Please refrainf from using spaces. use underscore(_) instead): Padma_Kutir
Another data(Y/n): y
Enter the hostel id: 10
Enter Hostel name(Please refrainf from using spaces. use underscore(_) instead): Varuna
Another data(Y/n): y
Enter the hostel id: 11
Enter Hostel name(Please refrainf from using spaces. use underscore(_) instead): Nilgiri
Another data(Y/n): y
Enter the hostel id: 12
Enter Hostel name(Please refrainf from using spaces. use underscore(_) instead): Sneh_Sadan
Another data(Y/n): n
Please make sure that hostel id field in the csv file has only the numbers you have entered now. If not please change it
Enter Data changed(Y/n): y
Data Already Entered
Run Hostelmanagement.py.py

Process finished with exit code 0
```

2. Frontend

```
D:\Pranav\venv\Scripts\python.exe D:/Pranav/Hostelmanagement.py
Enter the username of mysql: root
Enter the password of the mysql: root
1. Add Student
2. Search for a student
3. Modify data
4. Display data
5. Delete data
6.Exit
Enter the option you want to select: 4
('HostelID', 'Hostel Name')
(1, 'North_House')
(2, 'South_House')
(3, 'East_House')
(4, 'West_House')
(5, 'RR_House')
(6, 'Gokul')
(7, 'River_Side_Hostel')
(8, 'Shanti_Vihar')
(9, 'Padma_Kutir')
(10, 'Varuna')
(11, 'Nilgiri')
(12, 'Sneh_Sadan')
Enter the name of the hostel you want to see the data of (Don't use space use underscore(_)):  Varuna
('Adminno.', 'Name', 'Age', 'Class', 'Gender', 'HostelID')
(1252, 'Egor Prall', 15, 10, 'Male', 10)
```

```
(2010, 'Adriano Franiak', 16, 11, 'Male', 10)
(2313, 'Gilles Ogilvy', 14, 12, 'Male', 10)
(2429, 'Niko Buyers', 16, 12, 'Male', 10)
(2777, 'Forbes Hufton', 18, 11, 'Male', 10)
(3012, 'Maximilianus Lenoir', 14, 12, 'Male', 10)
(3180, 'Mayer Bergstrand', 16, 10, 'Male', 10)
(3254, 'Darrin Riddock', 16, 10, 'Male', 10)
(3277, 'Orlando Gerner', 19, 12, 'Male', 10)
(3602, 'Aldwin Trim', 16, 10, 'Male', 10)
(3678, 'Dean Crinkley', 16, 11, 'Male', 10)
(4038, 'Anatollo Bocken', 14, 10, 'Male', 10)
(4871, 'Harvey Forsdicke', 18, 10, 'Male', 10)
(5044, 'Rickey Kyndred', 16, 12, 'Male', 10)
1. Add Student
2. Search for a student
3. Modify data
4. Display data
5. Delete data
6.Exit
Enter the option you want to select: 1
Enter the name of the hostel in which you want to add the student in: Varuna
Enter the admission number of the student: 5045
Enter the name of the student: juli
Enter the age of the student: 17
Enter the class of the student: 12
Enter the gender of the student(Male,Female): Male
```

```
1. Add Student
2. Search for a student
3. Modify data
4. Display data
5. Delete data
6.Exit
Enter the option you want to select: 4
('HostelID', 'Hostel Name')
(1, 'North_House')
(2, 'South_House')
(3, 'East_House')
(4, 'West_House')
(5, 'RR_House')
(6, 'Gokul')
(7, 'River_Side_Hostel')
(8, 'Shanti_Vihar')
(9, 'Padma_Kutir')
(10, 'Varuna')
(11, 'Nilgiri')
(12, 'Sneh_Sadan')
Enter the name of the hostel you want to see the data of (Don't use space use underscore(_)):  Varuna
('Adminno.', 'Name', 'Age', 'Class', 'Gender', 'HostelID')
(1252, 'Egor Prall', 15, 10, 'Male', 10)
(2010, 'Adriano Franiak', 16, 11, 'Male', 10)
(2313, 'Gilles Ogilvy', 14, 12, 'Male', 10)
(2429, 'Niko Buyers', 16, 12, 'Male', 10)
(2777, 'Forbes Hufton', 18, 11, 'Male', 10)
(3012, 'Maximilianus Lenoir', 14, 12, 'Male', 10)
(3180, 'Mayer Bergstrand', 16, 10, 'Male', 10)
(3254, 'Darrin Riddock', 16, 10, 'Male', 10)
(3277, 'Orlando Gerner', 19, 12, 'Male', 10)
(3602, 'Aldwin Trim', 16, 10, 'Male', 10)
(3678, 'Dean Crinkley', 16, 11, 'Male', 10)
(4038, 'Anatollo Bocken', 14, 10, 'Male', 10)
(4871, 'Harvey Forsdicke', 18, 10, 'Male', 10)
(5044, 'Rickey Kyndred', 16, 12, 'Male', 10)
(5045, 'juli', 17, 12, 'Male', 10)
```

```
1. Add Student
2. Search for a student
3. Modify data
4. Display data
5. Delete data
6.Exit
Enter the option you want to select: 2
Enter the adminno of the student you want the data of: 5044
('Adminno.', 'Name', 'Age', 'Class', 'Gender', 'HostelID')
(5044, 'Rickey Kyndred', 16, 12, 'Male', 10)
1. Add Student
2. Search for a student
3. Modify data
4. Display data
5. Delete data
6.Exit
Enter the option you want to select: 3
enter the hostel name you want to change the data of: Varuna
How many columns do you want to modify: 1
Enter the admin number of the student you want to change the data of: 5044
Enter the column name you want to change the data of: Age
What do you want to write: 17
1. Add Student
2. Search for a student
3. Modify data
```

```
4. Display data
5. Delete data
6.Exit
Enter the option you want to select: 2
Enter the adminno of the student you want the data of: 5044
('Adminno.', 'Name', 'Age', 'Class', 'Gender', 'HostelID')
(5044, 'Rickey Kyndred', 17, 12, 'Male', 10)
1. Add Student
2. Search for a student
3. Modify data
4. Display data
5. Delete data
6.Exit
Enter the option you want to select: 5
Enter the name of the hostel you want to delete data from: Varuna
Enter the admin number of the student you want to delete the data of: 5045
1. Add Student
2. Search for a student
3. Modify data
4. Display data
5. Delete data
6.Exit
Enter the option you want to select: 4
('HostelID', 'Hostel Name')
(1, 'North_House')
(2, 'South_House')
```

```
(3, 'East_House')
(4, 'West_House')
(5, 'RR_House')
(6, 'Gokul')
(7, 'River_Side_Hostel')
(8, 'Shanti_Vihar')
(9, 'Padma_Kutir')
(10, 'Varuna')
(11, 'Nilgiri')
(12, 'Sneh_Sadan')
Enter the name of the hostel you want to see the data of (Don't use space use underscore(_)):  Varuna
('Adminno.', 'Name', 'Age', 'Class', 'Gender', 'HostelID')
(1252, 'Egor Prall', 15, 10, 'Male', 10)
(2010, 'Adriano Franiak', 16, 11, 'Male', 10)
(2313, 'Gilles Ogilvy', 14, 12, 'Male', 10)
(2429, 'Niko Buyers', 16, 12, 'Male', 10)
(2777, 'Forbes Hufton', 18, 11, 'Male', 10)
(3012, 'Maximilianus Lenoir', 14, 12, 'Male', 10)
(3180, 'Mayer Bergstrand', 16, 10, 'Male', 10)
(3254, 'Darrin Riddock', 16, 10, 'Male', 10)
(3277, 'Orlando Gerner', 19, 12, 'Male', 10)
(3602, 'Aldwin Trim', 16, 10, 'Male', 10)
(3678, 'Dean Crinkley', 16, 11, 'Male', 10)
(4038, 'Anatollo Bocken', 14, 10, 'Male', 10)
(4871, 'Harvey Forsdicke', 18, 10, 'Male', 10)
(5044, 'Rickey Kyndred', 17, 12, 'Male', 10)
```

```
1. Add Student
2. Search for a student
3. Modify data
4. Display data
5. Delete data
6.Exit
Enter the option you want to select: 6

Process finished with exit code 0
```

HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirements:

1. Intel I3 or above or Amd equivalent
2. 2GB Ram or above
3. 256 GB Storage or Above
4. No Graphic card Needed

Software Requirements:

1. Windows 7 / 8 / 8.1 / 10 / 11 or equivalent Mac or Linux
2. Python 3.9 or above
3. Mysql Community Server
4. Python Idle

Bibliography

- Stackoverflow.com
- W3schools.com
- Progate.com
- Github.com
- YouTube- codewithHarry
- YouTube – techwithtim
- Sumita Arora Class XII
- MySQL Docs
- Dev.Mysql.com
- Superuser.com
- PGT
- Mackaroo.com
- Cloud.Google.com
- PyCharm
- Visual Studio Code
- Python Docs