



Bangabandhu Sheikh Mujibur Rahman Science And Technology University, Gopalganj- 8100



Project Report on “Abdullah Chatrabash Mess Management System”

Course Name: Introduction To Programming With Python

SUBMITTED BY	SUBMITTED TO
Md. Rokonzaman Batch ID : 25B014 Batch No : 25 <i>Department of Management Studies, BSMRSTU</i>	Trainers Name: Nasif Ahmed <i>Assistant Professor, Department of CSE, BSMRSTU</i> Abu Bakar Muhammad Abdullah <i>Assistant Professor, Department of CSE, BSMRSTU</i>

Date of Submission: 3 September 2024

EDGE-BSMRSTUCSE Digital Skills Training



**ICT
DIVISION**



EDGE
ENHANCING DIGITAL GOVERNMENT AND ECONOMY

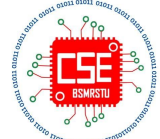


Table of Contents

1. Introduction	1
2. Features/Functions	1
2.1 Adding New Students	1
2.2 Updating Meal Counts	2
2.3 Checking Balance	2
2.4 Checking Meal Count	2
2.5 Retrieving Student Status	3
2.6 Displaying All Students	4
2.7 Update balance for a student	4
3. Implementation	5
4. Future Work	5
5. Conclusion	6
References	6

Abdullah Chatrabash Mess Management System Report

1. Introduction

The **Abdullah Chatrabash Mess Management System** is a Python-based application designed to streamline the management of student accommodations, particularly focusing on meal tracking and financial record-keeping. This system addresses the common challenges faced by student housing administrators, such as keeping track of individual student meal consumption, managing deposits, and maintaining accurate records of student information.

The project aims to digitize and automate these processes, reducing manual errors and providing a more efficient way to manage student accommodations. By offering features like adding new students, updating meal counts, checking balances, and generating student status reports, the system provides a comprehensive solution for mess management.

2. Features/Functions

The **Abdullah Chatrabash Mess Management System** offers several key features, each implemented with specific Python functions. Here's an overview of the main features along with their corresponding source code:

2.1 Adding New Students

This feature allows administrators to add new students to the system, recording their basic information and initial deposit.

```
def insert_student(student_id, name, mobile, address, deposit):  
    with open('MessData.txt', 'a') as f:  
        f.write(f"{student_id},{name},{mobile},{address},{deposit},0\n")
```

This function appends a new line to the 'MessData.txt' file with the student's information. The last value '0' represents the initial meal count.

2.2 Updating Meal Counts

Administrators can update the number of meals taken by a student.

```
def update_meals(student_id, meals):
    with open('MessData.txt', 'r') as f:
        data = f.readlines()

    updated_data = []
    for line in data:
        if line.startswith(student_id):
            parts = line.strip().split(',')
            parts[5] = str(int(parts[5]) + meals)
            updated_data.append(','.join(parts) + '\n')
        else:
            updated_data.append(line)

    with open('MessData.txt', 'w') as f:
        f.writelines(updated_data)
```

This function reads the file, updates the meal count for the specified student, and then rewrites the entire file with the updated information.

2.3 Checking Balance

Users can check the current balance of a student's account.

```
def check_balance(student_id):
    with open('MessData.txt', 'r') as f:
        data = f.readlines()

    for line in data:
        if line.startswith(student_id):
            parts = line.strip().split(',')
            return float(parts[4])

    return None
```

This function searches for the student's record and returns their current balance.

2.4 Checking Meal Count

This feature allows checking the total number of meals a student has taken.

```
def get_meals_count(student_id):
    with open('MessData.txt', 'r') as f:
        data = f.readlines()

    for line in data:
        if line.startswith(student_id):
            parts = line.strip().split(',')
            return int(parts[5])

    return None
```

Similar to the balance check, this function retrieves the meal count for a specific student.

2.5 Retrieving Student Status

This feature provides a comprehensive view of a student's information.

```
def get_student_status(student_id):
    with open('MessData.txt', 'r') as f:
        data = f.readlines()

    for line in data:
        if line.startswith(student_id):
            parts = line.strip().split(',')
            return {
                "ID": parts[0],
                "Name": parts[1],
                "Mobile": parts[2],
                "Address": parts[3],
                "Balance": parts[4],
                "Meals": parts[5]
            }

    return None
```

This function returns a dictionary containing all the information about a specific student.

2.6 Displaying All Students

This feature allows administrators to view information about all students in the system.

```
def display_all_students():  
    with open('MessData.txt', 'r') as f:  
        data = f.readlines()  
  
    for line in data:  
        parts = line.strip().split(',')  
        print(f"  
        ID: {parts[0]},  
        Name: {parts[1]},  
        Mobile: {parts[2]},  
        Address: {parts[3]},  
        Deposit: {parts[4]},  
        Meals: {parts[5]}")
```

This function reads and displays all student records in a formatted manner.

2.7 Update balance for a student

This feature allow to update balance.

```
def update_balance(student_id, amount):  
    with open('MessData.txt', 'r') as f:  
        data = f.readlines()  
  
    updated_data = []  
    for line in data:  
        if line.startswith(student_id):  
            parts = line.strip().split(',')  
            parts[4] = str(float(parts[4]) + amount)  
            updated_data.append(','.join(parts) + '\n')  
        else:  
            updated_data.append(line)  
  
    with open('MessData.txt', 'w') as f:  
        f.writelines(updated_data)
```

3. Implementation

The **Abdullah Chatrabash Mess Management System** is implemented in Python, utilizing a simple text file ('MessData.txt') as its data storage mechanism. This approach allows for easy data persistence without the need for a complex database system.

The main components of the implementation include:

- ❖ **Data Storage:** Student data is stored in 'MessData.txt', with each line representing a student record. Fields are comma-separated, following the format: ID, Name, Mobile, Address, Deposit, Meals.
- ❖ **File Handling:** The system uses Python's built-in file handling capabilities to read from and write to the data file. This includes appending new records, reading existing ones, and updating information.
- ❖ **User Interface:** A command-line interface is implemented using a while loop in the 'main()' function. This provides a menu-driven interaction for users to access various features of the system.
- ❖ **Error Handling:** Basic error handling is implemented, such as checking for the existence of a student before performing operations and providing appropriate feedback to the user.
- ❖ **Modular Design:** The system is designed with separate functions for each major operation, promoting code reusability and maintainability.

The implementation follows a procedural programming paradigm, which is suitable for the scope and complexity of this project. Each function is designed to perform a specific task, making the code easy to understand and modify.

4. Future Work

While the current implementation of the Abdullah Chatrabash Mess Management System provides a solid foundation for basic mess management, there are several areas where the system could be enhanced:

- ❖ **Database Integration:** Replace the text file storage with a proper database system (e.g., SQLite or MySQL) to improve data management, support concurrent access, and enable more complex queries.
- ❖ **Graphical User Interface (GUI):** Develop a GUI using a framework like Tkinter or PyQt to make the system more user-friendly and accessible to non-technical users.
- ❖ **Advanced Reporting:** Implement features for generating detailed reports, such as monthly summaries, financial statements, and meal consumption trends.
- ❖ **User Authentication:** Add a login system with different access levels for administrators and students.
- ❖ **Meal Planning:** Integrate a meal planning feature that allows administrators to set weekly or monthly meal schedules.

- ✧ **Payment Integration:** Implement a system for online payments and automatic balance updates.
- ✧ **Notifications:** Add a feature to send notifications to students about low balances or important announcements.
- ✧ **Data Backup and Recovery:** Implement regular data backup and recovery mechanisms to prevent data loss.
- ✧ **Mobile Application:** Develop a companion mobile app for students to check their balances, meal counts, and receive notifications.
- ✧ **API Development:** Create a RESTful API to allow integration with other systems or future front-end developments.

5. Conclusion

The **Abdullah Chatrabash Mess Management System** provides an efficient solution for managing student accommodations, particularly focusing on meal tracking and financial record-keeping. By digitizing these processes, the system reduces the potential for human error and streamlines administrative tasks.

Key achievements of the project include:

- ✧ Successful implementation of core features such as student registration, meal tracking, and balance management.
- ✧ Creation of a user-friendly command-line interface for easy interaction with the system.
- ✧ Efficient data management using a simple yet effective file-based storage system.

While the current implementation meets the basic requirements of mess management, there is significant potential for future enhancements. These improvements could transform the system into a more robust, feature-rich application capable of handling complex scenarios and integrating with other systems.

In conclusion, the **Abdullah Chatrabash Mess Management System** serves as a solid starting point for digital mess management, with ample opportunities for growth and refinement to meet evolving needs in student accommodation management.

References

1. Python Software Foundation. (2024). Python Documentation. <https://docs.python.org/3/>
2. Sweigart, A. (2019). Automate the Boring Stuff with Python: Practical Programming for Total Beginners. No Starch Press.
3. Guttag, J. V. (2016). Introduction to Computation and Programming Using Python: With Application to Understanding Data. MIT Press.
4. Lutz, M. (2013). Learning Python: Powerful Object-Oriented Programming. O'Reilly Media, Inc.
5. McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media, Inc.

6. Ramalho, L. (2021). *Fluent Python: Clear, Concise, and Effective Programming*. O'Reilly Media, Inc.
7. Summerfield, M. (2010). *Programming in Python 3: A Complete Introduction to the Python Language*. Addison-Wesley Professional.
8. Van Rossum, G., Warsaw, B., & Coghlan, N. (2001). PEP 8 -- Style Guide for Python Code. Python.org. <https://www.python.org/dev/peps/pep-0008/>
9. Alchin, M. (2010). *Pro Python*. Apress.
10. Phillips, D. (2015). *Python 3 Object-Oriented Programming*. Packt Publishing Ltd.
11. <https://www.w3schools.com/python/>
12. <https://claude.ai>