Project Report – 40228592, 40226585

1 Introduction

1.1 Project Summary

1.1.1 Objectives

- Any programming language can be used to develop the system: **Python**.
- For database need to use Structured Query Languages (**SQL**).
- Avoid duplication of widget and label names.
- Support API to parse data in locally and perform CRUD operation.
- Sync CRUD operation with **Tkinter** for user interface.
- Implementation of agile philosophy (e.g.: **SCRUM** Framework).

1.2 Project Scope

1.2.1 In Scope Functionality

- Call API to perform CRUD operation into the system.
- Ability to add, update, delete one or many row together.
- Ability to clear widgets of system.
- Perform CRUD operation
- Create design patterns
- Design architecture
- Use of testing tools
- Use of refactoring strategies.

1.2.2 Out of Scope Functionality

- Login functionality.
- Security
- Support different operating system.

1.3 System Perspective

1.3.1 Assumptions

- Testing of System as of November 14.

1.3.2 Constraints

Impending changes to API data may impact data design and system.

2 System Requirements

The requirements in this document are prioritized as follows:

Value	Rating	Description
1	Critical	This requirement is critical to the success of the project. The project will not be possible without this requirement.
2	High	This requirement is high priority, but the project can be implemented at a bare minimum without this requirement.
3	Medium	This requirement is somewhat important, as it provides some value, but the project can proceed without it.
4	Low	This is a low priority requirement, or a “nice to have” feature, if time and cost allow it.
5	Future	This requirement is out of scope for this project and has been included here for a possible future release.

2.1 Functional Requirements

Req#	Priority	Description
FR-G-001	1	To call API into the system. API Link: https://dummyjson.com/users.
FR-G-002	2	Perform Crud Operation
FR-G-003	3	Design patterns for the system
FR-G-004	4	Create User interface
FR-G-005	2	Use of testing tools
FR-S-001	2	Create architectural model
FR-R-001	2	Use of refactoring strategies

2.2 Non-Functional Requirements

ID	Requirement
NFR-001	Login Functionality
NFR-002	Support difference operating system (OS)

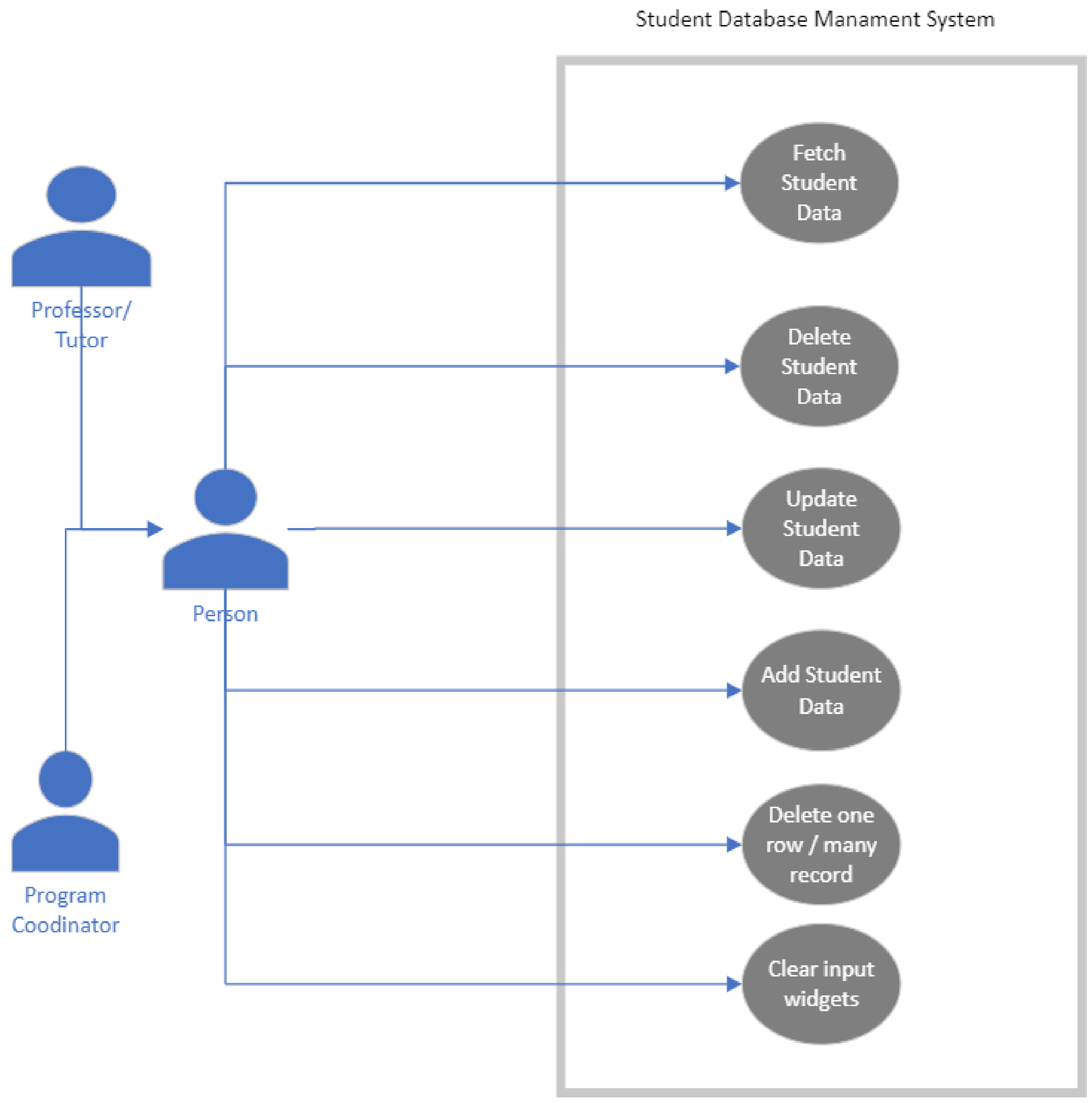
3 Process Overview

UI of Project:
https://github.com/joyal7701/app_project/blob/main/APP%20Project%20wireframe.pdf

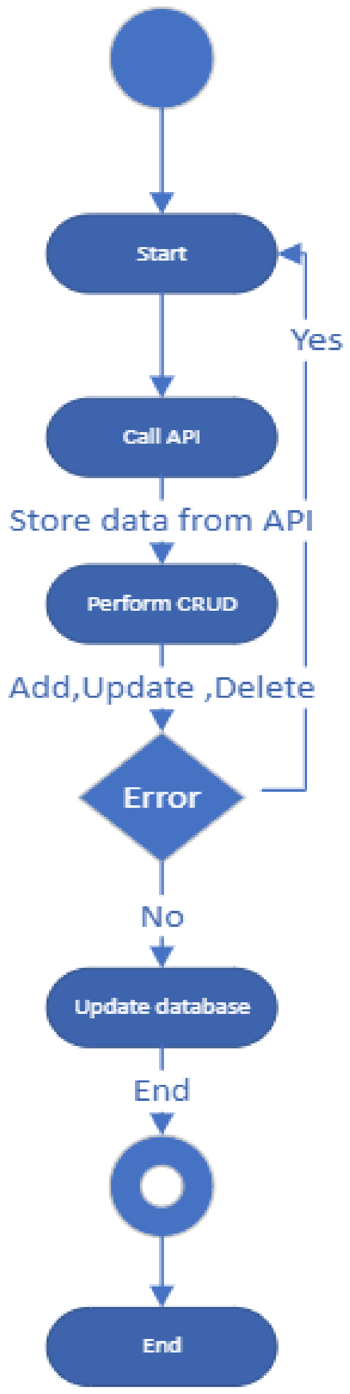
3.1 Process Overview (As-Is)

At any point the system to support activities such as API call and perform crud operation.

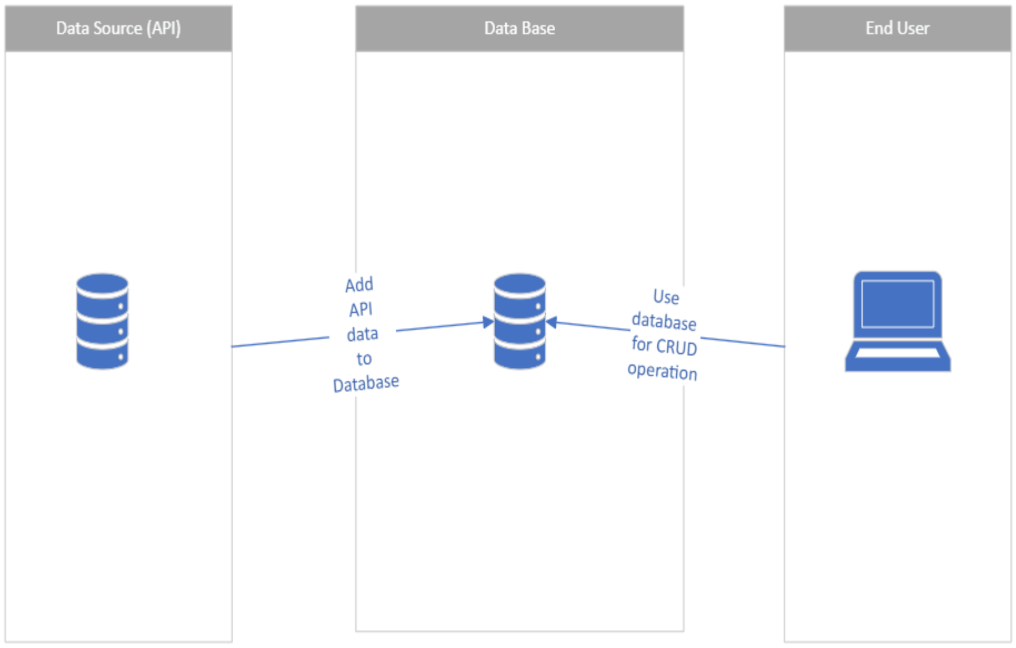
1. **Use Case Diagram:**



2. **System Activity diagram:**



3. **<u>Data Source Architecture:</u>**

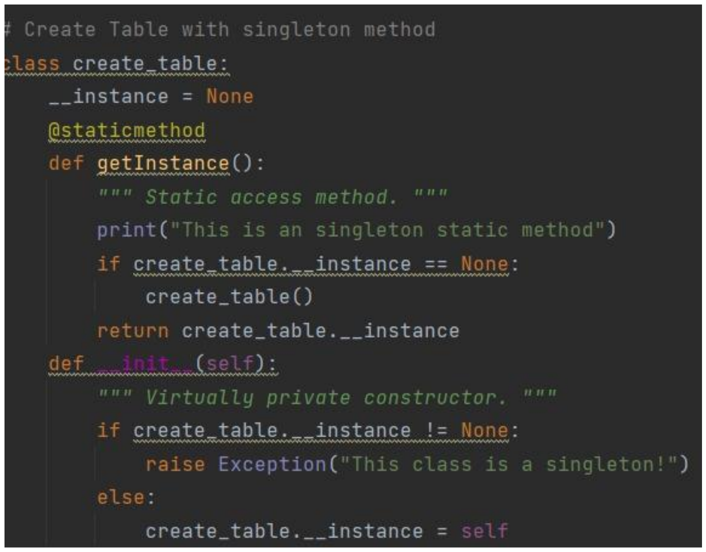


4 Implemented Coding standards

- **Import** statements are added on separate lines.
- Throughout the coding part it is ensured that all **comments** look clear and easily understandable.
- **Class names** are given by using the CapWords convention.
- **Function names** and **variable names** are in in lowercase, with words separated by underscores to improve readability of the code.
- **Dictionaries** are used while mapping of things.
- **No whitespace** is given inside brackets, braces, or parentheses.

5 Use of applicable pattern

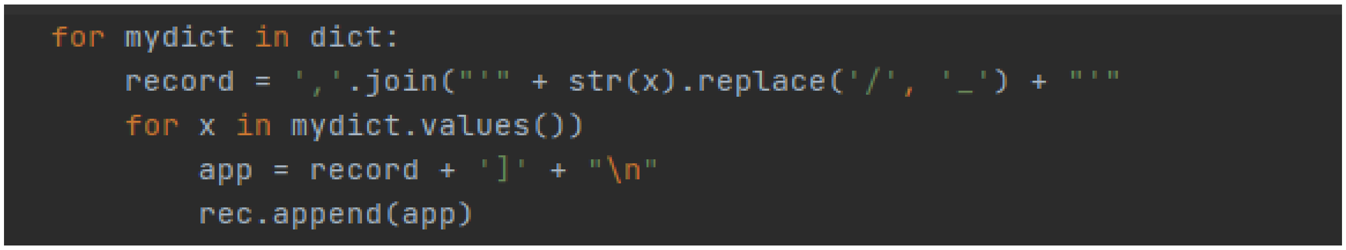
THE SINGLETON DESIGN PATTERN



```
# Create Table with singleton method
class create_table:
    __instance = None
    @staticmethod
    def getInstance():
        """ Static access method. """
        print("This is an singleton static method")
        if create_table.__instance == None:
            create_table()
        return create_table.__instance
    def __init__(self):
        """ Virtually private constructor. """
        if create_table.__instance != None:
            raise Exception("This class is a singleton!")
        else:
            create_table.__instance = self
```

6 Use of Refactoring Strategy

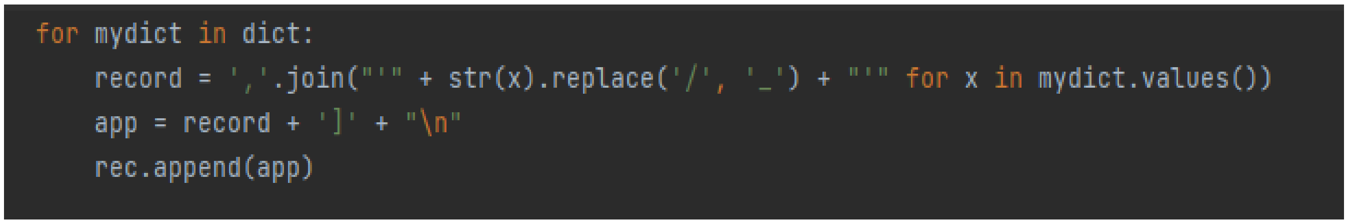
Before:



```
for mydict in dict:
    record = ','.join("'" + str(x).replace('/', '_') + "'"
    for x in mydict.values())
        app = record + ']' + "\n"
        rec.append(app)
```

<u>Refactoring nested loop with shared variables</u>

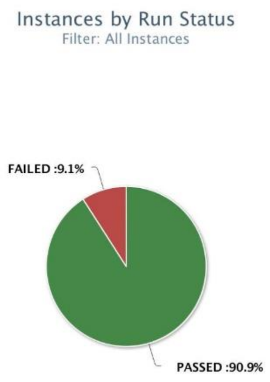
After:



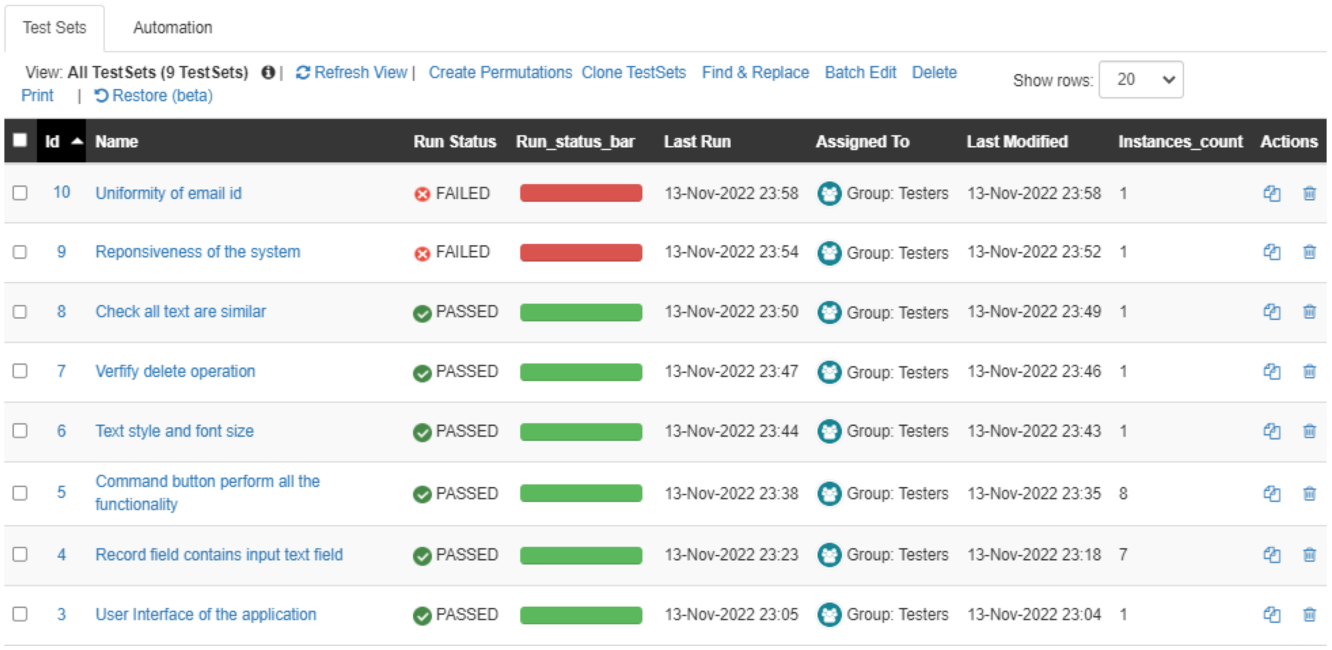
```
for mydict in dict:
    record = ','.join("'" + str(x).replace('/', '_') + "'" for x in mydict.values())
    app = record + ']' + "\n"
    rec.append(app)
```

7 Testing Tool

Manual testing has been done for the project. **PractiTest** end-to-end **Test Management tool** used in the project for test case management. It provides mapping of requirements to test cases and find log defects inside the platform. It gives clear result with good dashboard in a professional way.



Link to see, analyze and run all Test Sets*:* https://prod.practitest.com/p/24909/tests



Id	Name	Run Status	Run_status_bar	Last Run	Assigned To	Last Modified	Instances_count	Actions
10	Uniformity of email id	FAILED	🟥	13-Nov-2022 23:58	Group: Testers	13-Nov-2022 23:58	1	
9	Reponsiveness of the system	FAILED	🟥	13-Nov-2022 23:54	Group: Testers	13-Nov-2022 23:52	1	
8	Check all text are similar	PASSED	🟩	13-Nov-2022 23:50	Group: Testers	13-Nov-2022 23:49	1	
7	Verify delete operation	PASSED	🟩	13-Nov-2022 23:47	Group: Testers	13-Nov-2022 23:46	1	
6	Text style and font size	PASSED	🟩	13-Nov-2022 23:44	Group: Testers	13-Nov-2022 23:43	1	
5	Command button perform all the functionality	PASSED	🟩	13-Nov-2022 23:38	Group: Testers	13-Nov-2022 23:35	8	
4	Record field contains input text field	PASSED	🟩	13-Nov-2022 23:23	Group: Testers	13-Nov-2022 23:18	7	
3	User Interface of the application	PASSED	🟩	13-Nov-2022 23:05	Group: Testers	13-Nov-2022 23:04	1	

8 Appendices

8.1 Related Documents

GitHub: https://github.com/joyal7701/app_project

API: https://dummyjson.com/users

Sprint Planning:
https://github.com/joyal7701/app_project/blob/main/Sprint%20report%20for%20project.xlsx

Wireframe Design:
https://github.com/joyal7701/app_project/blob/main/APP%20Project%20wireframe.pdf

Jira Tool:
https://project205.atlassian.net/jira/software/projects/SP/boards/1/roadmap?assignee=unassigned

Testing Cases: https://github.com/joyal7701/app_project/blob/main/Test%20cases.xlsx

Testing Tool: https://prod.practitest.com/p/24909/tests