**Detailed Design Specification**

**< IT\_Capstone 4905 >**

**< Budget Tracker System>**

**Team Member: Andrew Yang, Jian Ma**

**Sponsor: Diana Bergeman**

#### **Approval Signatures**

|  |  |  |
| --- | --- | --- |
| **Approved by: Business Project Leader** |  | **Approved by: IM/IT Project Leader** |
|  |  |  |
|  |  |  |
|  |  |  |
| **Prepared by: Business Project Manager** |  | **Prepared by: IM/IT Project Manager** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | **Reviewed by: Quality Assurance Manager** |

This is the section for the control of the outline planning, which teammates or sponsors can edit and update this document. The editor should provide the date and the description of the changes, in order to maintain the development and distribution of revisions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision Number** | **Date of Issue** | **Author(s)** | **Description of Change** |
|  |  |  |  |

**Table of Contents**

1. **Project Summary**
2. **Points of Contact**
3. **Introduction**
   1. **Purpose, Scope, and Objectives**
4. **Overall Description**
   1. **Functions**
   2. **Use cases**
   3. **Operating environment**
   4. **User Documentation**
   5. **Assumptions and Dependencies**
   6. **Project Constraints**
5. **Use Case Diagrams (revised from Requirements Specification)**
6. **Class Diagrams**
7. **State Diagrams**
8. **Interaction Diagrams (sequence or collaboration at your choice)**
9. **Component Diagrams**
10. **Package Diagrams**
11. **Project Summary**

Create a budget tracking system that allows users to log purchases, expenses, etc., pull up reports on any defined chart string and see pending charges, expended charges, and report to show % of the year that’s gone by, % of funds already spent, amount available to spend, etc.

1. **Points of Contact**

|  |  |  |
| --- | --- | --- |
| Role | Name | Email address |
| Team member | Andrew Yang | andrewyang@my.unt.edu |
| Team member | Jian Ma | jianma@my.unt.edu |
| Sponsor | Diana Bergeman | diana.bergeman@unt.edu |
|  | David Keathly | david.keathly@unt.edu |

1. **Introduction**

In order to analyze the budget of the College of Engineering in UNT, the sponsor needs to use Microsoft Excel to complete the task. However, there are hundreds of data needs to be processed, which makes it very complicated by using Excel. Hence, we decided to use Microsoft Access software to create a more efficient and easy method to help sponsor solve this problem.

* 1. Purpose, Scope, and Objectives
* Create a budget tracking system by using Microsoft Access.
* The product should have a variety of functions to fulfill sponsor’s requirements.

1. **Overall Description**
   1. Functions

* Allow the data entry
* Allow to create new indexes
* Has the ability to sort the data
* Running reports in any field with any or a combination of parameters identified
* calculate how much of total budget is spent
* calculate how much of my fiscal year has gone by
* System needs to be able to run based off of fiscal year time frames rather than giving me everything that ever existed
  1. Use cases

This budget tracking system is managed by the administrative coordinator of the department. Only one person needs to access this product. Therefore, this product can be local based instead of connecting to the internet.

* 1. Operating environment

This budget tracking system is based on Microsoft Access. Due to Access can only be run on Windows, users can only use this product on a Windows computer or a Mac computer which can access VMware fusion.

* 1. User Documentation

In order to show how to use the product we created, we will give a manual to the user. The manual will be given in a Word or PowerPoint document.

* 1. Assumptions and Dependencies

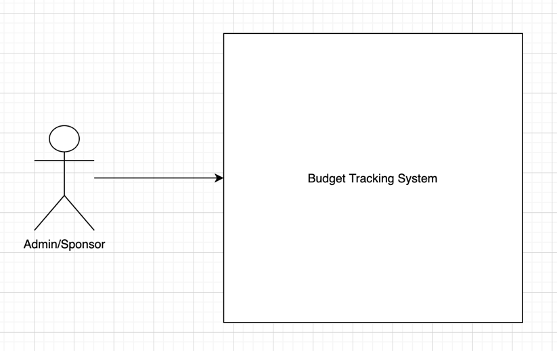
Due to both our team members’ laptops being under MacOS, we have to figure out how to access Windows on our laptop. This could be the biggest factor that will affect the requirements. And our software is locally based, so maybe only one of our team members can develop the software at the same time. Our product will be only performed using Microsoft Access, so we don’t need any other software components.

* 1. Project Constraints

The constraints for our product are that it is a local based software, so maybe it cannot access the internet for multiple users to use it at the same time. Also, users can only use this software under the Windows system.

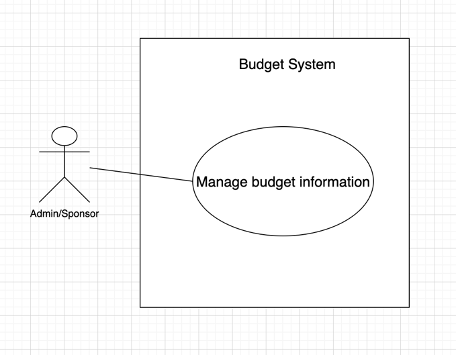
1. **Use Case Diagram (revised from Requirements Specification)**

**Context diagram:**

****

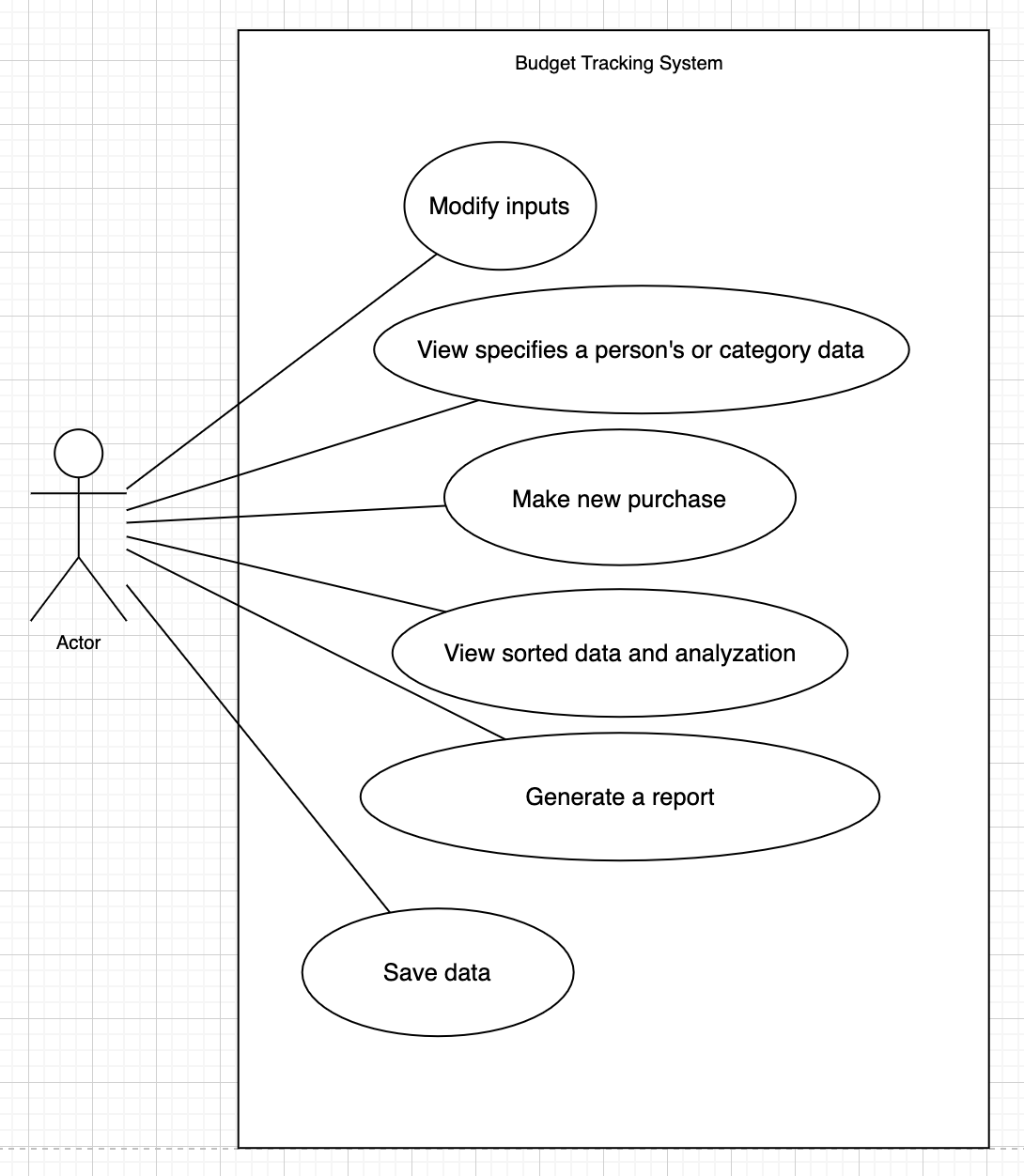
From the context diagram, we can see there will be only one role in the budget tracking system. The admin or sponsor’s main duty is using this system to keep tracking the budget of the department.

**Level 1 diagram:**

****

Manage budget information: there would be budget information management modules which would be used by the sponsor who could modify the budget data, view the data, calculate budget data, also could save data in personal PC or account.

**Level 2 diagram:**

****

1. Modify the inputs: sponsor could edit, add, or delete the budget data which can include name or budget number of the fund.

2. View specific data: the sponsor could select a range of funds and see the specific data of those funds.

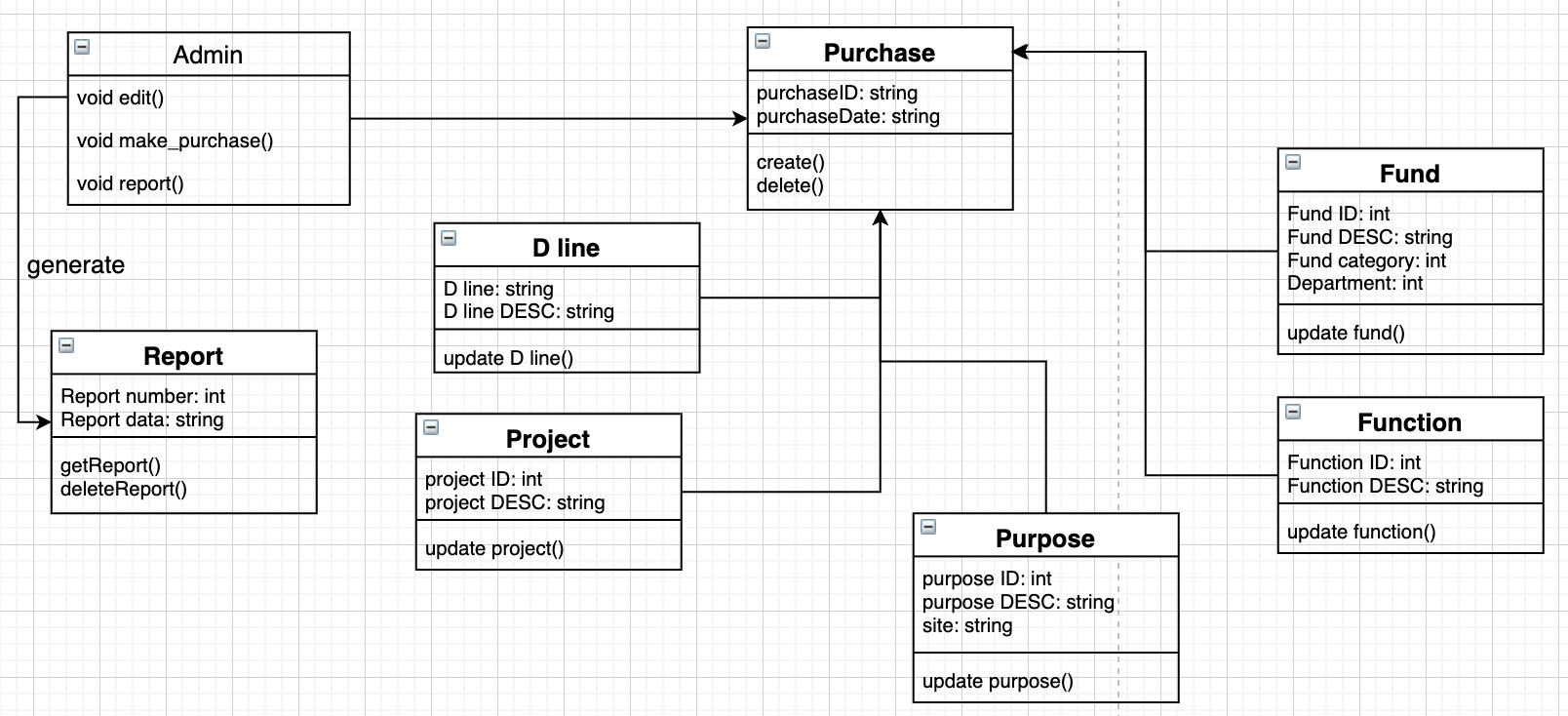
3. Make a new purchase: the sponsor could add a new purchase to the list.

4. View sorted data and analyzation: the sponsor could sort each category, which could make it easier to do some analyzation on the data.

5. Generate a report: the sponsor can generate reports from a different range of funds.

6. Save data: the sponsor could save the data into a personal PC or Microsoft account if he/she needs.

1. **Class Diagram**

****

1. Admin: the admin is the sponsor who will access to this budget tracking system

2. Purchase: this will allow sponsor to create new purchase in the budget

3. Fund: a collection class which will store the information of the fund which includes ID, description, category and so on. Also, the sponsor could modify it.

4. D line: sponsor can update information in the list of D line.

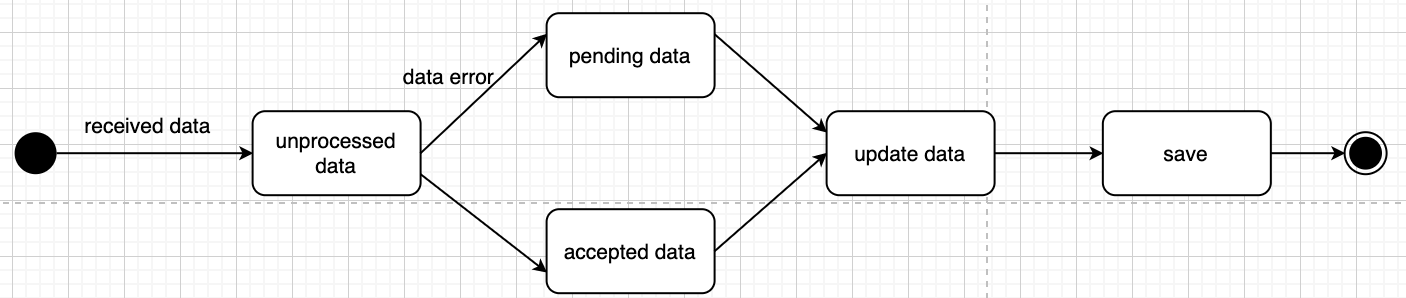
5. Report: this will allow the admin to get the report, which will include the report number and the date of the report.

6. Project: a collection class which will store the information of the project.

7. Function: this will allow the sponsor to update the function list.

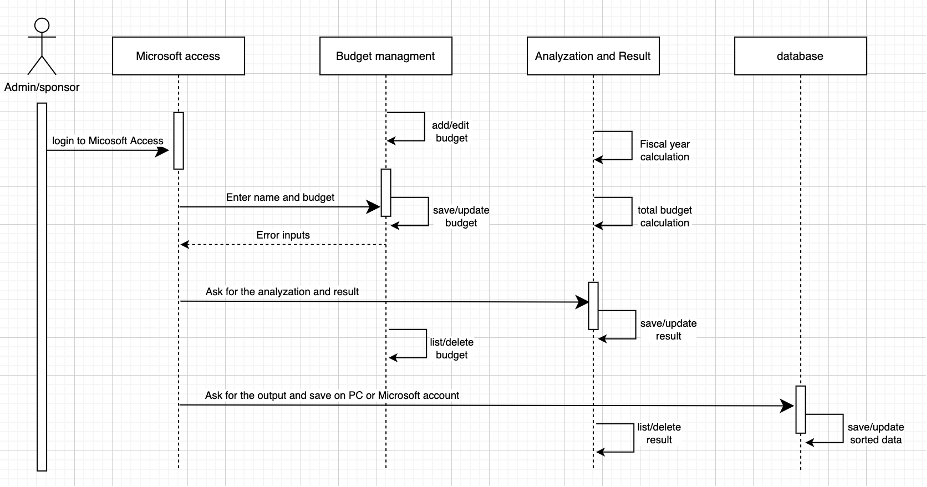
8. Purpose: sponsor can modify the information in the purpose list.

1. **State Diagram**

****

When the sponsor receives the new data, then she needs to use this system to update the information. The first state is unprocessed data. If there’s errors in the data, she will go to the pending data state to wait until the data is available to be processed. If there’s no error, she will directly update the data in the database. Finally, if the sponsor finishes her task, she will go to save state to save the data.

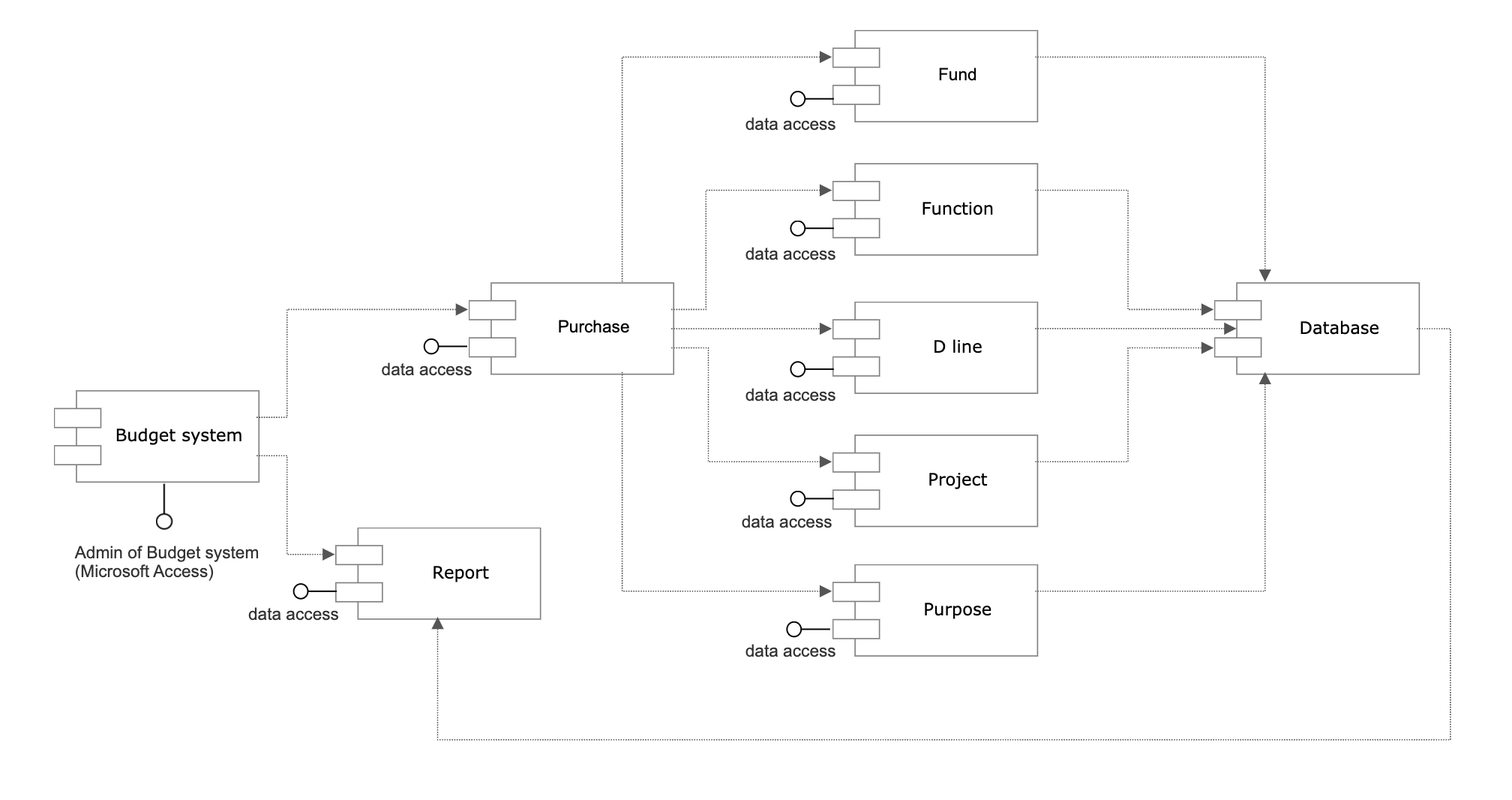
1. **Interaction Diagram (sequence or collaboration at your choice)**

****

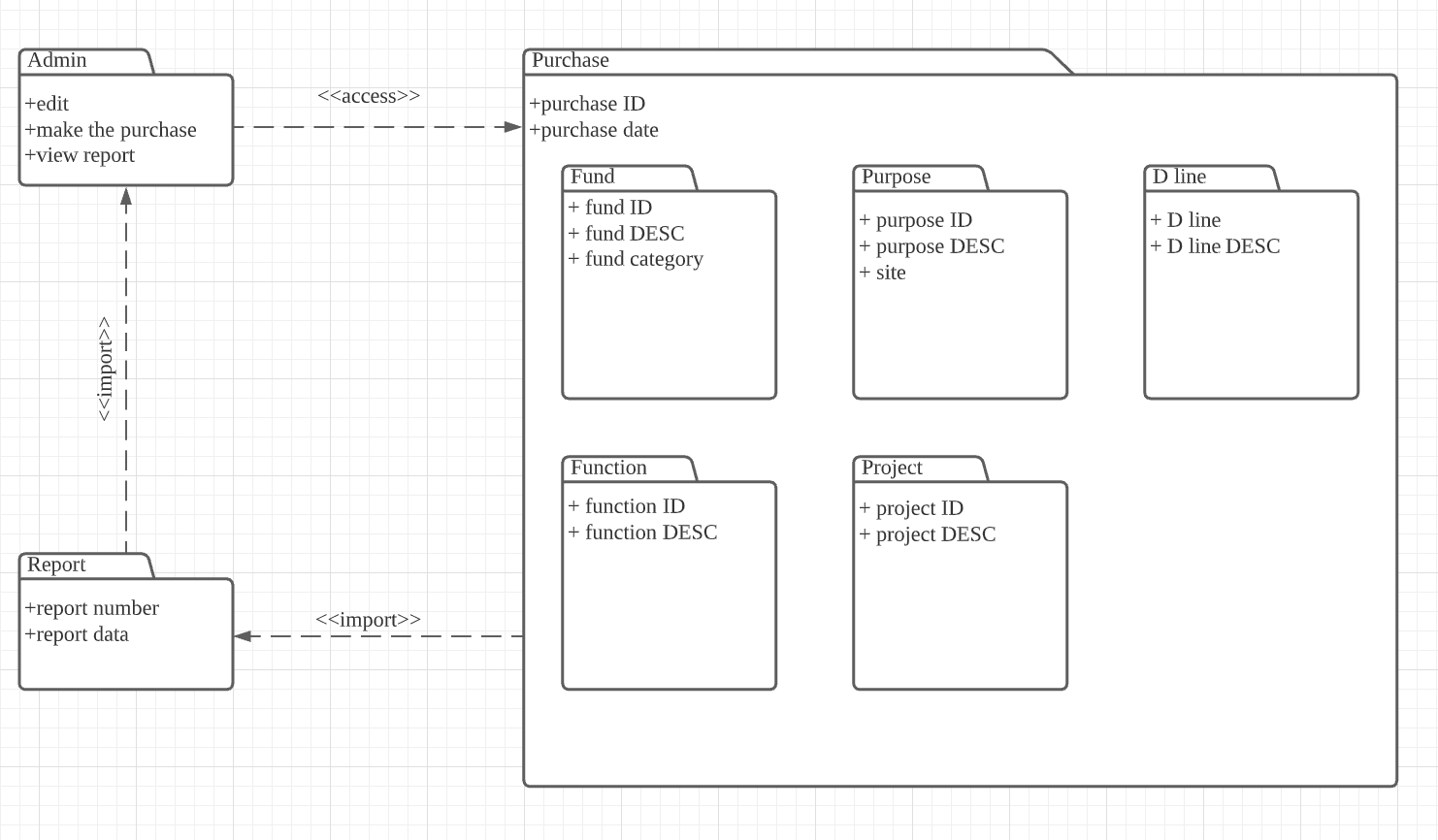
1. Enter name and budget: the sponsor can send a Modify message to add, edit, or delete the data, if the input is wrong, it will return an Error inputs message to the sponsor.

2. Ask for the analysis and result: the sponsor sends Get, or Totalbudget message to get the report of the budget. And a report will be returned.

3. Save: the sponsor will send a Save message to save the data on PC or Microsoft account.

1. **Component Diagram**The Admin uses the Microsoft Access interface to enter the budget system, then enter the data in the Purchase component. Those inputs will be divided into 5 components which are Fund, Function, D line, Project, and Purpose. After any data or sorted data will be sorted in the database and go to the report component; therefore, the admin can enter the inputs and show the specified data.

1. **Package Diagram**

****

The Admin accesses the purchase package in order to do the calculation or any action. The package will be calculated and divided into Fund, Purpose, D line, Function, and Project, which sorted the data and category and imported it to the report package. After the report part receives the data, the admin allows control and import the data output back to the admin.