**Final Acceptance Test Procedure with results**

**< 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. **Purpose of ATP**
6. **Testing Requirements**
7. **System Integration Test Procedure**
8. **Use-Case (Functional) Requirements Test Procedure**
9. **Performance Requirements Test Procedure**
10. **Summary Report**
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. **Purpose of ATP**

The purpose of ATP is to test the project to verify it meets the requirements of the sponsor. It will describe some specific test requirements and procedures which can help us to ensure our project is accepted to our sponsor.

1. **Testing Requirements**

* Facilities

The tester can test in any place he/she wants, just need to download the budget tracking file and open it in Microsoft Access.

* Hardware and software

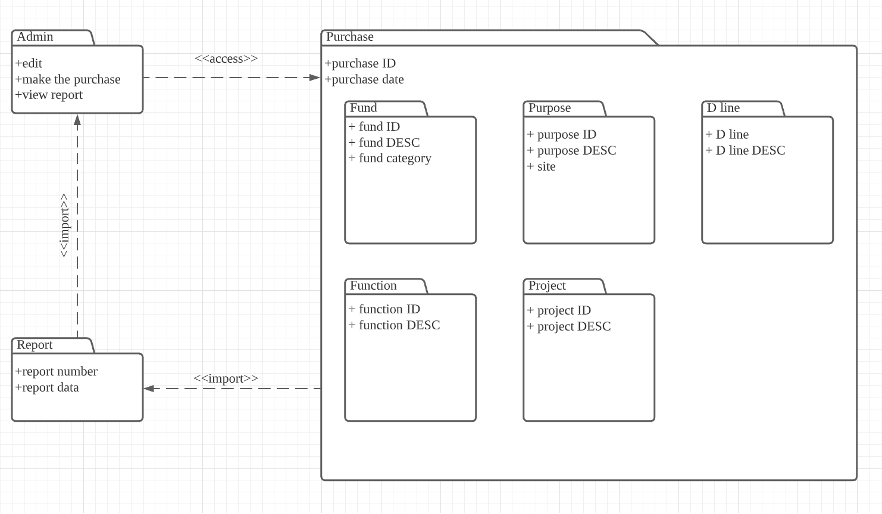
The tester needs to have a PC or laptop with the installation of Microsoft Access. It is better to use Windows PC to open this system, but he/she also can use VMware or Parallel to compile the system on a MacBook.

* Test process

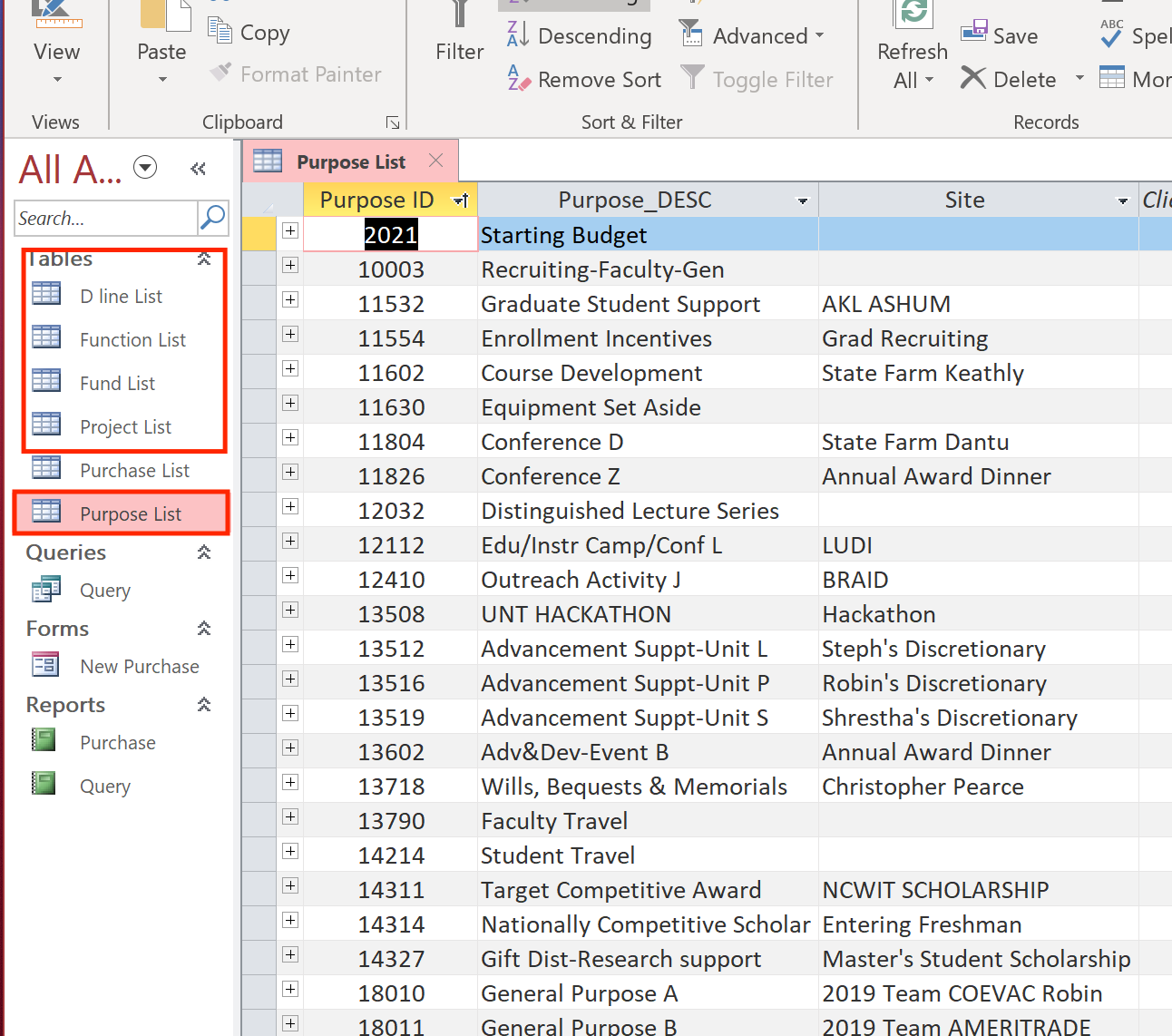
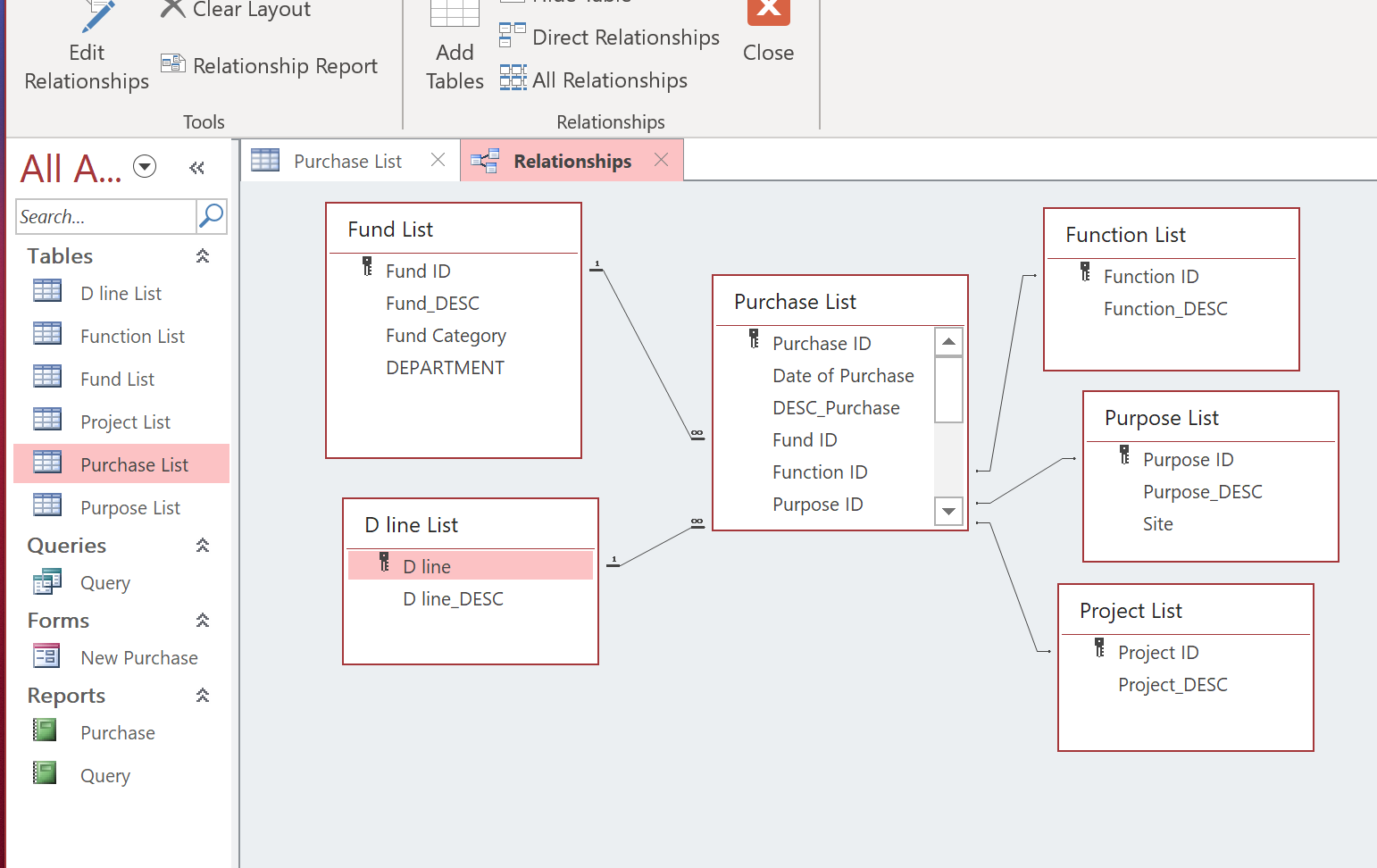
We will test the project based on the following steps to make sure all requirements are met.

* Test inputting the data into each table
* Test make new forms (fulfilling the data and check if each button works well)
* Go to Purchase table to see if the purchase has been created
* Test sorting the data
* Test querying data and add some criteria
* Running report
* Check if the calculation in the report is correct

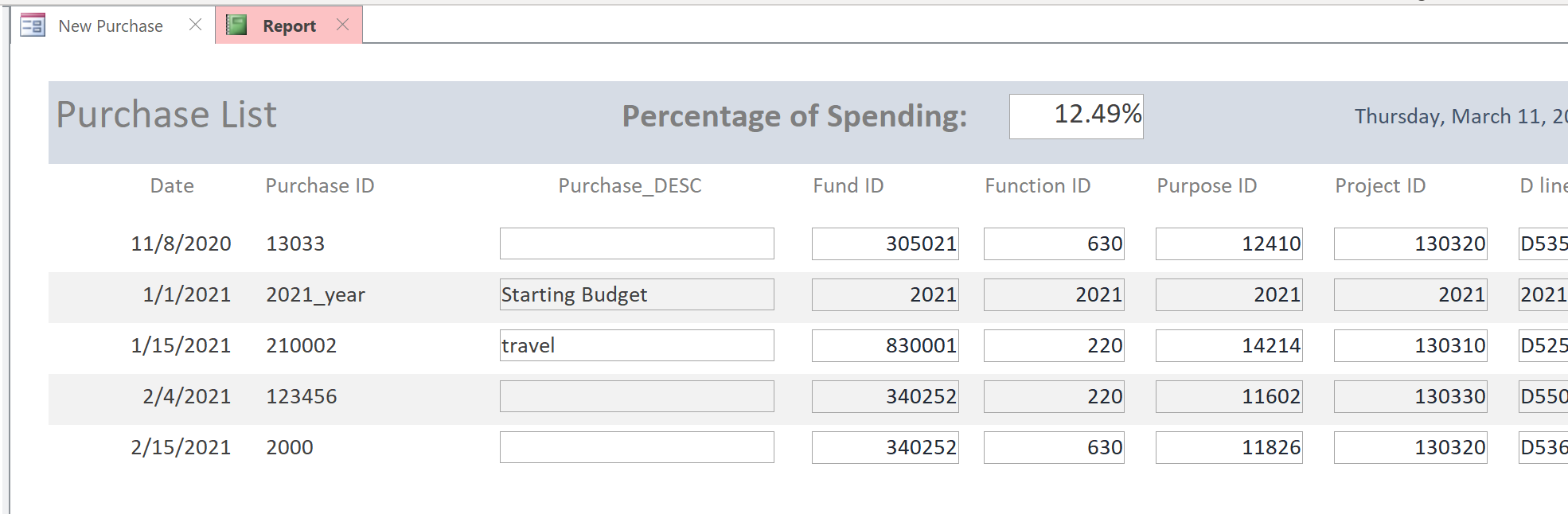
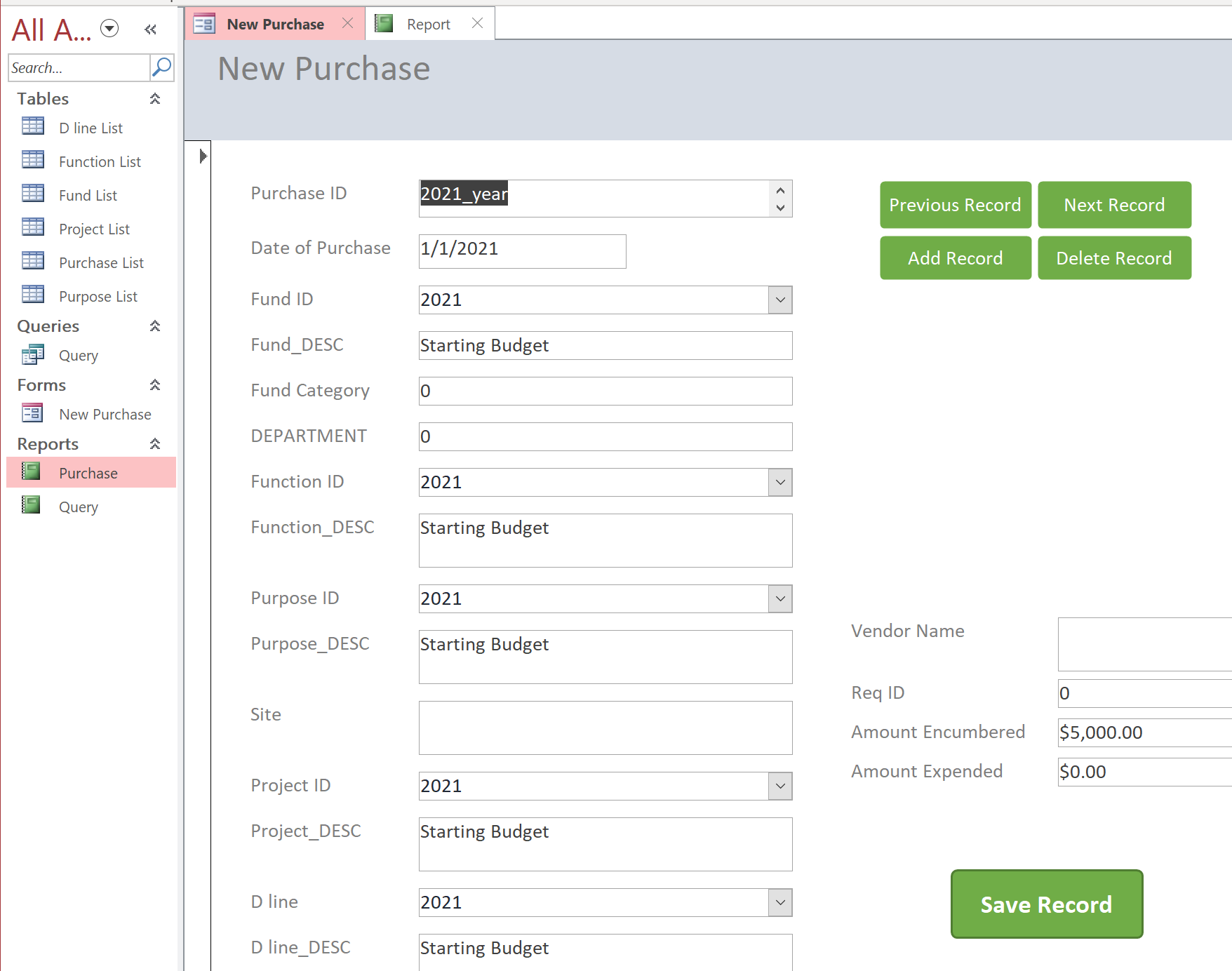
1. **System Integration Test Procedure**



Based on the diagram we created in SRS, our system has some lists which store the information of each category. D line, Function, Fund, Project, and Purpose tables are independent tables which store some sub-data. Also, the Purchase table has some relationships with other tables’ data. So, we need to make sure the Purchase table will be updated if the sub-table’s data has been modified.

For the forms, queries, and reports, we need to make sure they can display the data from the Purchase list.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | Expected result | Correction required | Comments | Results (Pass/Fail) |
| 1 | Verify D line list | Can be opened without errors |  |  | Pass |
| 2 | Verify Function list | Can be opened without errors |  |  | Pass |
| 3 | Verify Fund list | Can be opened without errors |  |  | Pass |
| 4 | Verify Project list | Can be opened without errors |  |  | Pass |
| 5 | Verify Purpose list | Can be opened without errors |  |  | Pass |
| 6 | Check Purchase List | Data is related to sub-tables |  |  | Pass |
| 7 | Check forms | Can display the data from Purchase List |  |  | Pass |
| 8 | Check queries | Can display the data from Purchase List |  |  | Pass |
| 9 | Check reports | Can display the data from Purchase List |  |  | Pass |

1. **Use-Case (Functional) Requirements Test Procedure**

Based on the functional requirements, we create the following procedure to test our project:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | Expected result | Correction required | Comments | Results (Pass/Fail) |
| 1 | Data input | Store data in correct format in each table |  |  | Pass |
| 2 | Make new form | Each button works, drop down list working |  |  | Pass |
| 3 | Purchase table updated | After performing each form, this table will be updated |  |  | Pass |
| 4 | Sort, filter data | Works well for each criterion in each table |  |  | Pass |
| 5 | Query data | Query data under different criteria |  |  | Pass |
| 6 | Generate report | Displays the results and calculated data |  |  | Pass |

1. **Performance Requirements Test Procedure**

Since our project is a Microsoft Access file, we don’t need to care more about the performance stuff. However, we need to check if each functionality works well when it contains plenty of data. Also, we need to check if the error message pops up if entering the wrong things…

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | Expected result | Correction required | Comments | Results (Pass/Fail) |
| 1 | Tables | Error message when entering wrong data format |  |  | Pass |
| 2 | Forms | Each button does the right stuff |  |  | Pass |
| 3 | Report | Shows the right data, and the calculation of percentage should be 100% correct |  |  | Pass |
| 4 | Speed | When plenty of data stored, the response of tables, forms, queries, and reports should still be less than 1 second |  |  | Pass |

1. **Summary Report**

After we tested the project and made corrections on defects we found over and over again, we finally got these results for each step and record it on the above tables. Since the results of overall test procedures and processes are Pass, we claim that our project as Pass.