Exploring Low-Code by Creating a Content Management System

Ahmet Uygun

A Capstone Project Submitted to the

University of North Carolina Wilmington in Partial Fulfillment

of the Requirements for the Degree of

Master of Science

Department of Computer Science

Congdon School of Supply Chain, Business Analytics, and Information Systems

University of North Carolina Wilmington

2022

Approved by

Advisory Committee

Ron Vetter _____

Yao Shi _____

Douglas Kline

_____

Chair

Accepted By

_____

Dean, Graduate School

Table Of Contents

Abstract……iii

CHAPTER 1: INTRODUCTION……4

CHAPTER 2: BACKGROUND……5

History of Low Code……5

PowerApps……6

CHAPTER 3: JUSTIFICATION……9

CHAPTER 4: USE CASES……11

Manage Person……11

Manage Student……12

Manage Professor……13

Manage Capstones……14

Manage Capstone Recognition……15

Capstone Wizard……16

CHAPTER 5: EXAMPLE APPLICATION……17

CHAPTER 7: DISCUSSION……25

Shadow IT……31

CHAPTER 8: LIMITATIONS AND FUTURE UPDATES……34

Limitations……34

Future Updates……35

CHAPTER 9: CONCLUSION……36

REFERENCES……37

APPENDIX A……40

ABSTRACT

We live in an era of rapid business transformation made possible by the internet and digital technologies. Companies exist solely to add more value to their customers in the most efficient way possible. Low-code platform enables the rapid generation and delivery of business applications with minimal effort to write in a coding language and minimal effort for environment installation and configuration, training, and implementation. With an ever-increasing number of businesses, low-code solutions can be a significant step forward in developing critical business applications. This paper describes the experience of developing a content management app using the Microsoft PowerApps low-code platform.

Keywords: Low-code, PowerApps, Power Platform, Power Automate, Canvas App.

CHAPTER 1: INTRODUCTION

Since the introduction of smartphones, apps have become an integral part of our daily lives. We have apps for everything from turning on a light switch to remembering a coworker's birthday. People will require twice as many apps now because the world is moving much faster than a few decades ago. Many businesses are considering moving forward and entering the world of technology due to the growing demand for all apps. A few years ago, organizations discovered a great way to solve their problems more straightforwardly, low-code. Some low-code and no-code platforms, but not all, are intended to enable self-service programming by non-technical users, such as those looking to build simple marketing or sales apps.

Low code has been a hot topic in the IT circle and organizations since it presented an environment to build a business solution application without coding. But there are lots of arguments like: "It is time-efficient but is it functional enough?", "Does use low-code solutions make you sacrifice the quality of the application? "How is the security?". This research will examine Microsoft's low-code app development platform, PowerApps, to explore low-code development's positive and negative values to create an "Annals Management System" that will help the Annals manager manage the capstone processes. We will share findings from the experiment in this paper with research about the subject and answer following questions.

1. Which option is better to create multi-table operations; Creating Automations by using the Stored procedures & Views or, Creating Collections and using LookUp to make the joins?
2. In what way Low-Code have advantages over full code?
3. In what way Low-Code have disadvantages over full code?
4. How did using other Power Platform components affect the overall developing experience? (i.e., Using Power Automate to create flows)

CHAPTER 2: BACKGROUND

1. History of low-code

The history of low-code is a little bit complicated. Although it appears in the last decade, we can scale its history to the 1950s. One could argue that the first high-level programming languages, including Fortran and COBOL, were something approximating low-code platforms. They saved developers from having to write arcane machine code to write applications. Of course, most programmers today would not consider languages like Fortran and COBOL to be low code. To program in these languages, one must write much code. However, when these languages first appeared in the 1950s, developers perceived them as low-code solutions compared to what had come before. (Tozzi, 2021)

Apple's release of HyperCard in 1987 was the next significant step forward in low-code and no-code programming. HyperCard was a development platform and framework that allowed programmers to create complex applications without writing code. They could use graphical menus to define interfaces and functionality. HyperCard also supported scripting for developers who desired code-level control, though it was not required. Indeed, Bill Atkinson, the creator of HyperCard, described it as a platform that "non-programmers" could use to "put together interactive information" without requiring advanced technical skills. (What is HyperCard? - Definition from Techopedia, 2022)

Although HyperCard was successful in its period, Apple discontinued development in 1998. Today's generation of developers barely remembers the platform. That history forgetfulness has created an opportunity for development platform vendors to resurrect low-code and no-code premise. That trend was underway by 2014 when Forrester coined the term "low code." It is unclear when "no code" became a standard part of technical terminology, but it was widespread use three or four years ago.

Nowdays, there are many great platforms for low-code development like Appian, Mendix, KiSSFLOW, etc. Although they resemble their ancestors like HyperCard, they offer a lot more with less effort for the user. Between those many options of low-code platforms, PowerApps is the most suited to use in this project since our university provides us with all Microsoft products. (Tozzi, 2021)

2. PowerApps

Power Apps is a suite of apps, services, connectors, and a data platform that provides a rapid development environment to build custom apps for your business needs. Using Power Apps, you can quickly create custom business apps that connect to your data stored either in the underlying data platform (i.e., Microsoft Dataverse) or various online and on-premises data sources. (e.g., SharePoint, Microsoft 365, Dynamics 365, SQL Server). (What is PowerApps? - Power Apps, 2022)

Microsoft promotes PowerApps as a drag-and-drop app builder that does not require users to have any coding experience. With the feature, Power Apps platform allows to create two types of apps: canvas apps and model-driven apps. To some extent, Canvas and Model-driven Power Apps are very similiar. Both are business apps that non-developers can create and, share similar components. The distinction is between user control and use cases. Some application scenarios are better suited for Canvas apps, while others are better suited for Model-driven apps.

On top of these two options, PowerApps includes a third development option, Portals, which allows organizations to create their websites that users outside their organizations can access. Users from outside their organizations can sign in using a variety of identities, create and view data in Microsoft Dataverse, and even browse content anonymously, all with the features and ease of a low code application.

a. Canvas Apps

The name comes from the visual canvas that greets customers when they choose to create a Canvas Power App. It resembles an empty PowerPoint slide and serves as a canvas for dragging and dropping various items (controls, actions, images, objects). As a result, Canvas apps are truly no-code apps. They do not necessitate any coding knowledge. Understanding Excel procedures and having some expertise with web and application surfaces, on the other hand, is beneficial. (Start building apps - Power Apps, 2022)

Every aspect of the final app, including the size and formatting of its components, is entirely under the users' discretion. Because Canvas apps allow great creativity and flexibility, no two Canvas apps are alike. Mobile devices are the primary platform for Canvas apps. They are not designed to be responsive, and there are just two layout options: portrait or landscape. This form of PowerApp is not bound by a single data source. Around 200 connections are available to seamlessly combine existing data, including SharePoint, Microsoft 365, SQL Server, and Dynamics 365. Canvas business apps may easily be shared with the rest of the company by the owner once they are Published.

b. Model-Driving Apps

Model-driven apps, on the other hand, start with the data. They can construct data models; thus, they are suitable for more complicated and sophisticated business applications. They do, however, necessitate more technical knowledge than Canvas apps. A data model, pre-existing in Dynamics 365 or the Common Data Service (CDS) or generated from scratch, is always at the heart of a model-driven app. Model-driven apps run in Dynamics 365 and are based on CDS. The data and information entered into the app control it more than the user who created it. (Start building apps - Power Apps, 2022)

To put it another way, what you put in impacts what you get out of it. Compared to Canvas apps, users have less control over appearance and functionality. Instead of the app conforming to the data, the data conforms to the app. As a result, the user interface components are prepared and selected based on the underlying data, resulting in a more rigid design. Individual elements, on the other hand, can be edited by users. Model-driven Power Apps are built to be responsive. Model-driven Power Apps provide developers with several capabilities that allow them to tailor the application to their specific needs, which is impossible with out-of-the-box solutions. With JavaScript, model-driven Power Apps may be tweaked and adapted.

c. Portals

Portals allow users to quickly create web applications as the similar operations in model-driven apps but can be accessed by users who do not have a PowerApps or Dynamics license.

Power Apps can also support external-facing websites, which enable users outside of their organizations to sign in with a variety of identities, create and view data in Microsoft Dataverse, and even browse content anonymously. Dynamics 365 Portals' full functionality, which was previously only available as an add-on to customer engagement apps (Dynamics 365 Sales, Dynamics 365 Customer Service, Dynamics 365 Field Service, Dynamics 365 Marketing, and Dynamics 365 Project Service Automation), is now available standalone in Power Apps. (Portals - Power Apps, 2022)

There is a learning curve, but you can build web applications quickly once you know how to use them. JavaScript can be used to customize them further.

In finding these answers, I will use Power Platform's PowerApps and Power Automate to create a complete app with all the CRUD activities. The app is going to be a Canvas App introduced in PowerApps.

CHAPTER 3: JUSTIFICATION

As previously stated, we intend to develop an app to assist our department's Annals Manager in his/her duties of adding people and capstones to the database, which will then push all of these capstones to our annals. When a student begins his/her capstone project, there is much paperwork to keep track of. The records are then inserted into SQL Server by one of our professors to be published on our website. The process itself is not complicated, but it is not safe to entrust the entire database to a graduate assistant due to the sensitive nature of the information. As a result, a professor's availability is limited due to a manual job that only requires some coding knowledge.

We thought it would be useful to create an Annals Management App that would allow us to easily read all of the information in the data stored, insert new Students, Faculty, and Capstones, update them, and delete them on both the computer and mobile app now that we were testing the PowerApps' Low-Code environment.

We can use power apps to create screens that look like wizards to help with data entry. Annals Manager must first create many people in the current system and then categorize them as students, faculty, or non-faculty before inserting them into a capstone project as the author, chairperson, and committee members with their IDs. This app allows Annals Manager or the delegate to add people and capstones to the system with the help of these wizards without dealing with IDs and with a user-friendly interface. Also, with our app's live data connection, it is possible to create "Galleries" that display a list of your choice. For example, a user can see a list of students and click/touch on a student to navigate to a page with detailed information about the student. We have gone over the capabilities in greater detail in the Use Cases chapter.

While doing so, our goal is to investigate the power of low-code, what the myths are, and what the reality is in our opinion, share our findings with the public, and work toward a future in which civil developers can easily create apps with no-code platforms.

CHAPTER 4: USE CASES

Use Case Name	Manage Person	
Use Case Description	Adding a person, updating it and deleting.	
Frequency		
Actors	Annals Manager, or delegate	
Related Use Cases	Manage Student, Manage Professor	
Stakeholders		
Happy Pathway	Read create update and delete record	
Preconditions	Program connected to the database	
Post-Conditions	A new record created, record updated, record deleted	
Flow of Events	**Actor**	**System**
	1. User clicked the "People" button on the homepage.	2. Navigates to People screen that reads and filter non-Faculty and non-Student people from the Person table.
	3. User clicks to the "Plus" icon on top-right-corner of the screen to add new person.	4. Navigates to "addPerson" screen that has three text boxes to fill and a button to save.
	5. Enters First Name, Last Name and e-mail for the person and clicks the "Save" button to save the changes.	6. Navigates to confirmation page that shows the user entry as a confirmation.
	7. When the data entered correct clicks to the "Confirm" button to confirm. If not can choose to click "Back" button to go back to Step 5.	8. Saves the data to the database and navigates to Success page.
	9. If user wants to add another person, can choose to click "Add Another" button to go back to Step 5 and repeat the steps between. If not can click to "Back" button to go back to Step 2 to do something else.	11. Navigates to "PersonDetails" screen that shows "PersonID, First Name, Last Name, e-mail if the person is not a Student or Professor. An edit and a trash can icons as update and delete buttons.
	10. Chooses a person from the list of people.	13. Updates the Person selected accordingly.
	12. Update the name or e-mail directly from this page and click Update to save the changes.	15. Deletes the selected person
	14. Click to trash can icon to delete the person selected.	
Alternate Pathways		
Exception Conditions	Step 15- Person doesn't have any capstones as committee member.	

Use Case table 1: Manage Person

Use Case Name	Manage Student	
Use Case Description	Adding a student, updating it, and deleting.	
Frequency		
Actors	Annals Manager, or delegate	
Related Use Cases	Manage Person, Manage Faculty, Manage Capstone	
Stakeholders		
Happy Pathway	Read create update and delete record	
Preconditions	Program connected to the database	
Post-Conditions	A new record created, record updated, record deleted	
Flow of Events	**Actor**	**System**
	1. User clicked the "Student" button on the homepage.	2. Navigates to "Student" screen that reads and filter Students from the Person table.
	3. User clicks to the "Plus" icon on top-right-corner of the screen to add new person.	4. Navigates to "addStudent" screen that has three text boxes to fill and a button to save.
	5. Enters First Name, Last Name and e-mail for the student and clicks the "Save" button to save the changes.	6. Navigates to confirmation page that shows the user entry as a confirmation.
	7. When the data entered correct clicks to the "Confirm" button to confirm. If not can choose to click "Back" button to go back to Step 5.	8. Saves the data to the database and navigates to Success page.
	8. If user wants to add another person, can choose to click "Add Another" button to go back to Step 5 and repeat the steps between. If not can click to "Back" button to go back to Step 2 to do something else.	10. Navigates to "StudentDetails" screen that shows "PersonID, First Name, Last Name, e-mail and title of the Capstone.An edit and a trash can icons as update and delete buttons.
	9. Chooses a student from the list of people. And click edit icon.	11. Updates the Person selected from Student and Person tables.
	10. Update the name or e-mail directly from this page and click Update to save the changes.	13. Deletes the selected person from Student and Person tables
	12. Click to trash can icon to delete the person selected.	
Alternate Pathways		
Exception Conditions	Step 15- Student doesn't have a capstone project yet.	

Use Case Table 2: Manage Student

Use Case Name	Manage Professor	
Use Case Description	Adding a Faculty, updating it, and deleting.	
Frequency		
Actors	Annals Manager, or delegate	
Related Use Cases	Manage Faculty	
Stakeholders		
Happy Pathway	Read create update and delete record	
Preconditions	Program connected to the database	
Post-Conditions	A new record created, record updated, record deleted	
Flow of Events	**Actor**	**System**
	1. User clicked the "Professor" button on the homepage.	2. Navigates to "Professor" screen that reads and filter faculty members from the Person table.
	3. User clicks to the "Plus" icon on top-right-corner of the screen to add new person.	4. Navigates to "addProfessor" screen that has three text boxes and two dropdowns for department and discipline to fill and a button to save.
	5. Enters First Name, Last Name and e-mail for the faculty and select the department ad discipline from the dropdowns and clicks the "Save" button to save the changes.	6. Pops the confirmation message that shows the user entry as a confirmation.
	7. When the data entered correct clicks to the "Confirm" button to confirm. If not can choose to click "Back" button to go back to Step 5.	8. Saves the data to the necessary database tables and navigates to Success page.
	9. If user wants to add another person, can choose to click "Add Another" button to go back to Step 5 and repeat the steps between. If not can click to "Back" button to go back to Step 2 to do something else.	11. Navigates to "ProfessorDetails" screen that shows "PersonID, First Name, Last Name, e-mail, department, url and titles of the capstones involved. An edit and a trash can icons as update and delete buttons.
	10. Chooses a faculty from the list of people.	13. Updates the Person selected accordingly from person and faculty tables.
	12. Update the name, e-mail, or department/discipline by clicking edit icon and click Update to save the changes.	15. Deletes the selected person from Faculty and Person tables
	14. Click to trash can icon to delete the person selected.	
Alternate Pathways		
Exception Conditions	Step 15- Faculty doesn't have a capstone project involved.	

Use Case Table 3: Manage Professor

Use Case Name	Manage Capstones	
Use Case Description	Adding a Capstone, updating it, and deleting.	
Frequency		
Actors	Annals Manager, or delegate	
Related Use Cases	Manage Person, Manage Student, Manage Faculty, Manage Capstone Recognition	
Stakeholders		
Happy Pathway	Read create update	
Preconditions	Program connected to the database	
Post-Conditions	A new record created, record updated	
Flow of Events	**Actor**	**System**
	1. User clicked the "Capstones" button on the homepage.	2. Navigates to "Capstone" screen that reads and shows Capstone titles with their author's name.
	3. User clicks to the "Plus" icon on top-right-corner of the screen to add new capstone.	4. Navigates to "addCapstone" screen that has ten text boxes and one date picker to fill and a button to save.
	5. Picks the People of the capstone according to their roles and clicks the Check button to see the role of the person and clicks the "Next" button to go to the next Page.	7. Navigates to confirmation page that shows the user entry as a confirmation.
	6. Fills the Title, Abstract, PDFurl, and annalsPaperNumber and picks the proposal date and defense date then clicks next.	9. Runs the Power Automate and adjusts some variables
	8. Reviews the input on the page and clicks Publish if evereything is okay.	10. Navigates to CapstoneSuccess page that shows the Title of the Capstone and has the Hompage button
	11. Click Capstones button on the screen and choose a capstone from the capstones list	12. Shows the Capstone Selected, displays Title, Abstract, defence date, proposal date, student, chairperson,committee members and annals volume/issue/paper numbers
	13. If user wants to edit the Capstone Information, can just click to the edit button in the capstone page and edit. If the user wants to edit committee member can use the proper buttons to insert update or delete a committee member	
Alternate Pathways		
Exception Conditions	9.1. Title of the capstone is not in the titles 9.2. Student doesn't already has a capstone	

Use Case Table 4: Manage Capstone

Use Case Name	Manage Capstone Recognition	
Use Case Description	Adding a capstone recognition, updating it and deleting.	
Frequency		
Actors	Annals Manager, or delegate	
Related Use Cases	Manage Capstone	
Stakeholders		
Happy Pathway	Read create update and delete record	
Preconditions	Capstone exist	
Post-Conditions	A new record created, record updated, record deleted	
Flow of Events	**Actor**	**System**
	1. User clicked the "Capstone" button on the homepage. And Selects the capstone that will have the recognition.	2. Navigates to "CapstoneList" screen and navigates to Capstone Details that shows capstone Information
	3. User clicks to the "Recognition" button to see the recognitions and add new recognition.	4. Navigates to "Recognitions" screen that has a gallerry and four text boxes to fill and a button to save.
	5. Enters First Name, Last Name, text and url for the recognition and clicks the "Save" button to save the changes.	6. Prompts a message to confirm the user entry.
	7. When the data entered correct clicks to the "Confirm" button to confirm. If not can choose to click "Back" button to go back to Step 5.	8. Saves the data to the database and navigates to Success page.
	9. If user wants to add another recognition, can choose to click "Add Another" button to go back to Step 5 and repeat the steps between. If not can click to "Back" button to go back to Step 2 to do something else.	11. Navigates to "RecognitionDetails" screen that shows "Title of the Capstone, text of recognition and URL. An Update button to make changes and a trash can icon as delete button
	10. Chooses a Capstone from the list.	14. Updates the Person selected accordingly.
	12. Click Update button and edit, text of recognition and URL. Click save button.	15. Deletes the selected recognition
	14. Click to trash can icon to delete the recognition selected.	

Use Case Table 5: Manage CapstoneRecognition

Use Case Name	Capstone Wizard	
Use Case Description	Adding a Student, Faculty, Person and Capstone	
Frequency		
Actors	Annals Manager, or delegate	
Related Use Cases	Manage Capstone, Manage Student, Manage Faculty, Manage Person	
Stakeholders		
Happy Pathway	Student, faculty and committee person already in the system	
Preconditions	Program connected to the database	
Post-Conditions	New records created	
Flow of Events	**Actor**	**System**
	1. Clicks " Wizard" from the middle of the page	3. Navigates to "Wizard_Page1" that has a dropdown list and two buttons
	3. User can select the student from the dropdown list and click to the arrow button next to it. Or can click to "Continue with a new student" and type first name and last name of the student and click to continue. Then Click Next	4. System uses a few variables to adjust the student's ID after user clicked to Contionue and shows the next button that navigates the user to Chairperson page
	5. User can select the professor from the dropdown list and click to the arrow button next to it. Or can click to "Continue with a new professor" and type first name and last name of the professor and click to continue. Then Click Next	6. System uses a few variables to adjust the professor's ID after user clicked to Contionue and shows the next button that navigates the user to Committee Members page
	7. User can select the committee from the dropdown list and click to the arrow button next to it. Or can click to "Continue with a new person" and type first name and last name of the committee and click to continue. And proceed with the same cycle for the 2nd committee.Then Click Next	8. System uses a few vairables to adjust the Committee members. And proceed to Capstone Information page
	9. User repeats the same cycle for 3rd, 4th and 5th committee (user can leave them empty if not necessary) and click Next	10. System uses a few vairables to adjust the Committee members. And proceed Committee Members 2 page
	11. User enters the Title, Abstract, PDFurl, annalsPaperNumber and picks the defense date and proposal date from the datepickers then click Next to Overview	12. System uses the variables to correctly fill the Capstone Members and Capstone Information
	13. User views and confirms the information and clicks Publish Button to save.	14. System runs the Power Automate and insert the capstone to the system then navigates to the Success Page.
Alternate Pathways		
Exception Conditions	14.1 Student Doesn't already have a capstone 14.2 Another Capstone with the same title doesn't exist in the system.	

Use Case Table 6: Capstone Wizard

CHAPTER 5: EXAMPLE APPLICATION

This chapter will discuss the good/bad, simple/complex parts of a low-code platform to give people an idea who want to learn how to use a low code platform. The example is our experience of creating the Annals Management App, which has a kind of complicated normalized SQL Server database working as the data source in the background. I started to create this app as a complete Novice to the Power Platform and improved my understanding of PowerApps and Power Automate. During the process, I understood how hard it is to learn low code development and how valuable the product will be if one learns appropriately.

While creating the app, I have bumped into many blockages that lost me quite a few hours of research every time. Nevertheless, the best part is that those blockages are not to be overestimated. It is primarily a slight lack of information or understanding about the Platform itself and its language. For PowerApps, using the community blogs is a huge help, even for professionals. Creating content with PowerApps is easier than you think, but the crucial part is "Do you know how to create it?".

Our app is named UNCW Annals Manager and has 28 Screens with different purposes; we can categorize them according to CRUD capabilities.

	CREATE	READ	UPDATE	DELETE
StudentsList		X		
StudentDetails		X		X
PeopleList		X		
PersonDetail		X		X
ProfessorsList		X		
ProfessorDetail		X		X
AddNewPerson	X			
UpdatePerson			X	
Success				
CapstoneList		X		
CapstoneDetails		X		X
InsertCapstone1		X		
InsertCapstone2				
InsertCapstone3	X			
UpdateCapstone			X	
InsertCapstoneSuccess				
AddCommittee	X	X		
UpdateCommittee		X	X	
Wizard_addStudent	X	X		
Wizard_addChair	X	X		
Wizard_addCommittee	X	X		
Wizard_addCommittee_1	X	X		
Wizard_addCapstone				
WizardOverview	X			
CapstoneRecognition	X	X	X	X

Table 7: CRUD table for application's Pages

As a first impression, PowerApps looks like a complicated interface to the eye. However, it is simple to create useful apps even on the first try with just a piece of basic knowledge about the platform. There are some good and bad parts about using low code to create your apps, and it all depends on the needs of your apps. If you have an, excel that you keep track of your records. However, it is too troublesome to write the same thing repeatedly to different files. It is a great platform to create a small interface that will read and write your excel files all over the place according to the developers' arrangements; then, this is your place.

Nevertheless, Suppose you need an app that will collect the data of an energy provider company and be used by thousands of people simultaneously. In that case, it might be a bit more complicated and

troublesome. However, our app is neither too complicated and overloaded nor straightforward to test things. We have a normalized database with tables with one-to-many relationships on SQL Server. We have six main tables we used to create the app.

1. Person
2. Student
3. Professor
4. Capstone
5. CommitteeMembers
6. CapstoneRecognition

The hardest part to create was the Student and Professors list for me for some reason. Creating joined tables is a tricky feat to accomplish; the best option for when you need the joins is to use Views created in SQL Server to have easier access to the data you need. Nevertheless, you do not always users have access to affect the databases directly, so it is not the end of the road if you cannot create a View of your taste. It is possible to write a Lookup formula to look through two different tables and show data accordingly. In the "Screen Shot 2" and "Screen Shot 3" the lists created are created through the lookup formula.

```
Sort(Filter(
          ForAll(Student As _student,
   With(LookUp(Person, ID = _student.personID),
      {
         personID: ID,
         firstName: firstName,
         lastName: lastName,
         displayValue: firstName & " " & lastName
      }
   )
), StartsWith(displayValue,SearchStudentTextbox.Text)),lastName)
```

When we do Insert, Update or Delete, the operation works instantly on the database but showing the results in the lists takes time to update; this is happening due to the Lookup formula that I have used to create the multi-table connection for the gallery. The downside of this slowness is not that apparent in using the app, but it primarily affects the start time of the app. Because the lookup formula takes a few minutes to screen and filter all the data, when users start the app, it takes up to five minutes to send back the student and professor lists. However, there are no problems with other lists in the app since they don't use the lookup formula to display multi-table content.

Creating dynamic content was a refreshing activity in the project. When creating the addNewPerson Page, I wanted to use the same page for all roles of people. With the help of variables, which can be used to make arrangements in the app by changing their values with a button click by assigning different values to different buttons and changing the page's visible and invisible components accordingly, we can keep the app smaller and faster, even if only by a small margin. Also, we are using the variables in Power Automate flows when the value needed has to be an integer, getting the value from the textbox and converting it to a variable to use in the flow. Here is the list of Variables used in the app

1. Addingstatus	22. S_ID	43. nm2
2. whichStudent	23. S_FirstName	44. NCM1_ID
3. name	24. S_LastName	45. NCM2_ID
4. lname	25. ns	46. isNew_CM3
5. operation	26. WStudentID	47. CM3_ID
6. status	27. NS_ID	48. CM3_FirstName
7. whichCapstone	28. isNew_Professor	49. CM3_LastName
8. CapstoneTitle	29. C_ID	50. isNew_CM4
9. APN	30. C_FirstName	51. CM4_ID
10. addMemberOrder	31. C_LastName	52. CM4_FirstName
11. delete	32. Nc	53. CM4_LastName
12. next1	33. NC_ID	54. isNew_CM5
13. next1_1	34. isNew_CM1	55. CM5_ID
14. next2	35. CM1_ID	56. CM5_FirstName
15. next2_1	36. CM1_FirstName	57. CM5_LastName
16. next3	37. CM1_LastName	58. nm3
17. next3_1	38. isNew_CM2	59. nm4
18. CM2	39. CM2_ID	60. nm5
19. CM4	40. CM2_FirstName	61. NCM3_ID
20. CM5	41. CM2_LastName	62. NCM4_ID
21. isNew_Student	42. nm1	63. NCM5_ID

Tables 8: Variables

Furthermore, one of the good uses of the variables is having condition-based variables that automatically set their value to decrease the human errors that can be made. A small example of this is the "addMemberOrder" variable that sets in an if condition code to decide the value, which is the member order of the committee members that will be added after the Capstone is created already. Typically users should enter the order of the member by hand, but it is effortless to make mistakes in this situation. Although it does not affect much in the system much, it is small ease of use trick that can be used.

The crucial parts of the app are Capstones and Wizard-related pages, which is the main reason for the project. It was a bit complicated to get the information for every detail in the Capstone details. Since our app has multiple paths to go to the Capstone Details, we need to use variables to change the source according to the page we came to.

```
If(
   whichCapstone = "Student",
   StudentCapstone.Selected.abstract,
   If(
      whichCapstone = "Professor",
      ChairedCapstones.Selected.abstract,
      CapstoneCollection.Selected.abstract
   ))
```

Variable whichCapstone is setting the user's path from the path taken. For example, suppose the user clicks to the Capstone from the StudentsDetails page. In that case, the variable will be set as "Student" and the Capstone information will come from the StudentCapstone Gallery to the labels on the CapstoneDetails Page. This might seem not very easy, but it is easier to create by comparing this method to the full-code platforms.

Another tricky part was on the InsertCapstone1 page when selecting the committee members. I could not create a dropdown that shows faculty and non-Faculty people, so I used the whole Person Table; the downside is that now we have every person on the list, including the students. So, I decided to use another If condition to give a warning if the user selects a student as a committee member. It is not preventing the user from selecting one, but at least it gives the warning to prevent the damage to some extent.

I think the easiest to do in Power Platform is to do automation. Automation that would take tens of lines of code to program in any language is only a few clicks away from completion with the Power Automate Flows. With the help of variables, which can be used to make arrangements in the app by changing their values with a button click by assigning different values to different buttons and changing the page's visible and invisible components accordingly, we can keep the app smaller and faster, even if only by a small margin. Automate is the part we can reuse indefinitely across the apps we will create and created; these are the predefined functions or a kind of Stored Procedures for Power Apps. We have 23 Automated Flows in our app that we used:

1) Update	13) UpdateCapstone
2) CapstoneTrial	14) PromoteToProfessor
3) DeleteProfessor	15) AddCommitteeMember
4) DeleteCapstone	16) spf.addStudent
5) Spf.addCommittee	17) Spf.CapstoneInsert
6) Spf.addProfessor	18) DeletePerson
7) InsertCapstone	19) AddPerson
8) DeleteCommitteeMember	20) WizardNewStudent
9) DeleteStudent	21) UpdateCommittee
10) WizardNewCommittee	22) UpdateFirstName and LastName
11) WizardNewProfessor	23) AddStudent
12) PowerApp -> Insert row (V2),Insert row (V2) 2,Insert row (V2) 3,Res...	

Table 9: Power Automate Flows

Some of them are not being used in the final product, but all were used in a version of the app; after improving the project's requirements by discussing them with my professors, I created new flows according to the new needs. Another good part of these flows is that you can use a stored procedure from SQL Server or create your stored procedures by selecting the tables you want to make a transaction and selecting the columns you want to fill.

After completing the app, I inserted the capstones of two students who finished their capstone projects last semester; I checked the annals website and confirmed the upload; I made some edit tests and got successful returns every time. With Microsoft's security layer, now our faculty have a secure app to upload these capstones to our annals. It can be used from a smartphone, tablet, or computer. With a few clicks, you can change ownership, give, or take permission from anyone within the organization. Furthermore, it is always live, so you can fix it yourself or add new features when you realize that you need some modifications.

"Microsoft Power Fx is a low-code language for expressing logic across the Microsoft Power Platform." It has based on Microsoft Excel and powers today's Microsoft Power Apps canvas apps. It enables diverse teams to communicate and save time and money by allowing development throughout

the entire spectrum, from "no-code" to "pro code," with no cliffs in between. Moreover, since practically every business person in the industry knows how to use Excel functions freely, more users can design apps with just a basic understanding of PowerApps Canvas Apps because they already know the Fx language. Of course, this does not mean that Fx is just the excel function; it is much more capable than that.

PowerApps is not only for non-developers; it is designed to get more businesspeople involved in the programming cycle, but it is also a fantastic tool for professional developers. PowerApps offers you everything as a low code platform, and if you want to improve and maintain the system, you can always use "pro-code" in Visual Studio to add the finishing touches. Developers can use low code to make the design more accessible, faster, and less costly. If that is not enough, they can always employ "the language tooling that unpacks a canvas app into constituent parts that can be edited with Visual Studio Code or Visual Studio." The tool itself is relatively new, having been introduced in January 2021. I believe that this tool will be more appealing to developers who believe that low-code platforms are not as helpful because they cannot get complex code some features with PowerApps. Professional developers can now create apps with low code and use pro-code only when necessary. This will significantly increase productivity because professional developers will be able to create most apps in a timeframe that is at least half that of complex code development.

To summarize, PowerApps empowers users with no-code features to do the essential things, and if you want to take it a step further and have better and more complex features, you must step on low-code part to write some code to enforce your app. Finally, if you are a professional with extensive experience, you can combine all three steps to create even more profound apps more efficiently. We will ignore the pro-code section for this project because we want to test low-code platforms for civil development. We want to see how useful it is to everyone if we do not use any other languages or platforms to perform.

CHAPTER 7: DISCUSSION

Creating content with low-code development platforms is as easy as creating a worksheet in excel with some extra steps. Most features are self-explanatory and usually need little to no code. Nevertheless, assuming that everything you can create is limited to these would be far from the truth. It is possible to create complex structures by using the Fx Language. In this chapter, we will discuss and answer the questions that we asked before:

1. Which option is better to create multi-table operations? Creating Automations by using the Stored procedures & Views or, Creating Collections and using LookUp to make the joins?

In PowerApps, there are no built-in functions to create a multi-table gallery to show the records in your database, but this does not mean it is impossible to have those tables Join and show together. There are three ways to create a gallery that would help you display joined tables:

a) Using the Views stored in the database created by the DB people.

b) Using the LookUp formula to create a pseudo join

c) Using the PowerApps' Collections to make an in-memory collection

For good and bad, using the Views created on SQL level is the most reliable and fast solution Since the data coming from the database is not processed by PowerApps, it works as fast as a single table connection. However, if you lack the necessary permissions or expertise, it is not all bad; using Lookup or Collections works excellent, but the only problem with them is that they require processing data from the database in the PowerApps, which slows down as the number of data increases.

In the beginning, I was assuming that the collections would be working a lot faster compared to the lookup formula. However, their display times are almost identical, with only collections working a

blink of an eye faster than the lookup. The reason should be that both functions are needed to get the records and make arrangements using the PowerApps data processor.

So, we can say that working with a complex data structure can slow down the civil developers more compared to the IT people who already have access to the whole structure.

2. In what way Low-Code have advantages over full code? In what way Low-Code have disadvantages over full code?

Full code platforms allow the user to create and change everything according to the needs and liking of the developer, and it can be said that they empower the developer to the fullest to give the freedom in the creation process. This freedom is the strong point of the full code over low-code platforms. However, full code platforms need the corresponding skills and knowledge to use this unlimited amount of power in return. Although one does not need to be an expert in the language they use in the platform, there needs to be an educated and experienced developer to create an app that's useful to an organization. In most cases, there are a group of people working on the same app, with some working on UI and some working on connections or security. There is a lengthy period for planning, creating, and deploying the product.

On the other hand, Low code platforms need minimal coding experience, just enough to understand the logic and some basic level knowledge about the platform they use. It can be said that in a few days of education, most businesspeople can create a helpful app with low code. Although low code platforms do not give the developer much freedom, with out-of-the-box functions and components, it returns the loss of freedom in the form of time.

If we were to speak about the PowerApps platform, it is possible to say that creating the UI is most of the work. After the UI is done, interior connections are the only thing left. As for the data source connections? It is a built-in function that you can complete with just a few clicks. Furthermore, the security

is managed by Microsoft, the creator of the app can give access to anyone in the organization, and your app is safe between Microsoft's firewalls. The app created can only be accessed with your organization's e-mail account and with permission from the owner, which adds a layer of security to the product.

3. How did using other Power Platform components affect the overall developing experience? (i.e., Using Power Automate to create flows)

Thinking about the Power Platform Components as separate programs would be a mistake. All Power Platform parts work like a machine's different parts; PowerApps creates the app, Power BI uses the data source affected by the PowerApps and can create visuals, charts, and reports, while Power Automate works as an automation process. The usefulness of the Power Automate has no bounds, and it is possible to create flows for every repetitive task like sending e-mails about a job done, setting alarms for situations, creating a flow for inserting rows to multiple tables or data sources.

As for our project, almost every Insert, Update, and Delete operation are done by Power Automates Flows. We used stored procedures used in the SQL level and tables to insert update or delete in those flows. Although it is possible to do this work without using Power Automate, it is probably the best way to do it since it can show the stats of the flows created. With those stats, it is possible to see the result of the flow when it runs, how much time it took to finish, during the process, which parts return with an error, and everything you want to know about an automation process.

A collection is a way to organize your data in PowerApps, a versatile tool for creating and managing a wide range of digital content. Collections have the following features: they can include folders, smart collections, and collections that contain multiple types of content. Collections are commonly thought of as a foundational component of a PowerApps solution. (Chatterjee, 2022) We are using the collection for the Student Table to show information from Person table. When creating the multi-table

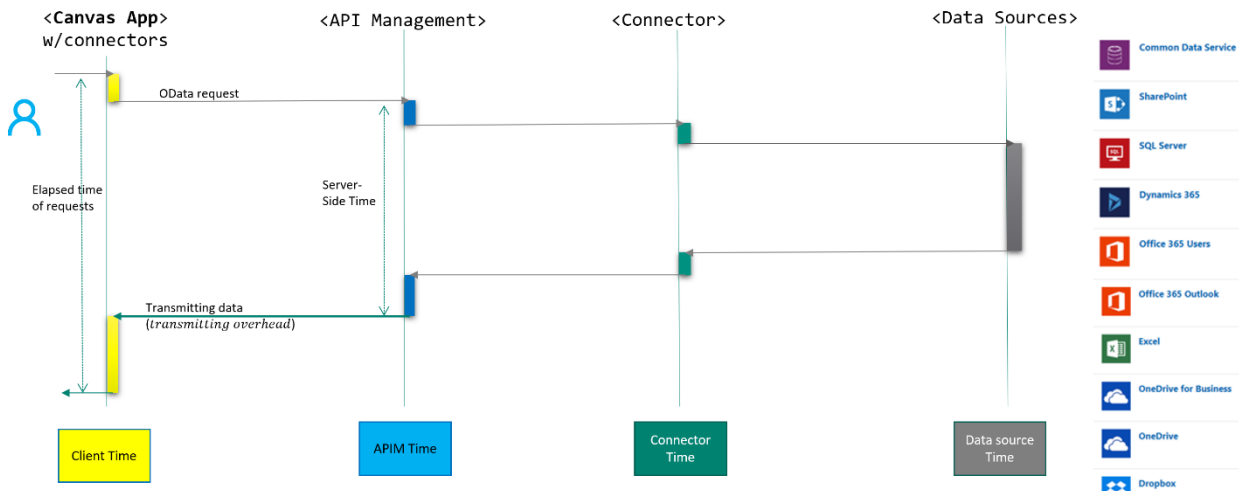
collection, we are using the Lookup formula to define the foreign key for creating the relationship between two tables.

As defined in Microsoft, The LookUp function finds the first record in a table that satisfies a formula. Use LookUp to find a single record that matches one or more criteria. But in our case, we are using it to create a join between two tables. We are using Lookup as a roundabout way to find a solution for being able to imitate a Join between two tables. The solution we used is not an optimal way to imitate the joins, but even so it is undeniable that it works, with the cost of a few minutes. (Filter, Search, and LookUp functions in Power Apps - Power Apps, 2022)

For creating a multi table collection we still need to use the Lookup formula. When I tried to think of a reason for why collection works slow it is probably because of the lookup formula we used to define the relationship between tables.

The Canvas app can access data from on-premises SQL via an on-premises data gateway. Once the on-premises data gateway is configured, the Power Apps canvas app can manage data with various on-premises data sources such as SQL, Oracle, and SharePoint on on-premises networks.

However, there is a chance that access to on-premises data sources will be slow due to the common causes listed below. Despite the fact that this article is primarily about SQL on-premises. They are still applicable to on-premises data sources. (Ahn, 2022)



$$UX\ timetaken = \sum_{i=1}^{n}(Layer\ Time) + \ transmitting\ Time$$

Figure 1 - How OData calls travel in Power Apps via connectors, except for the Dataverse connector

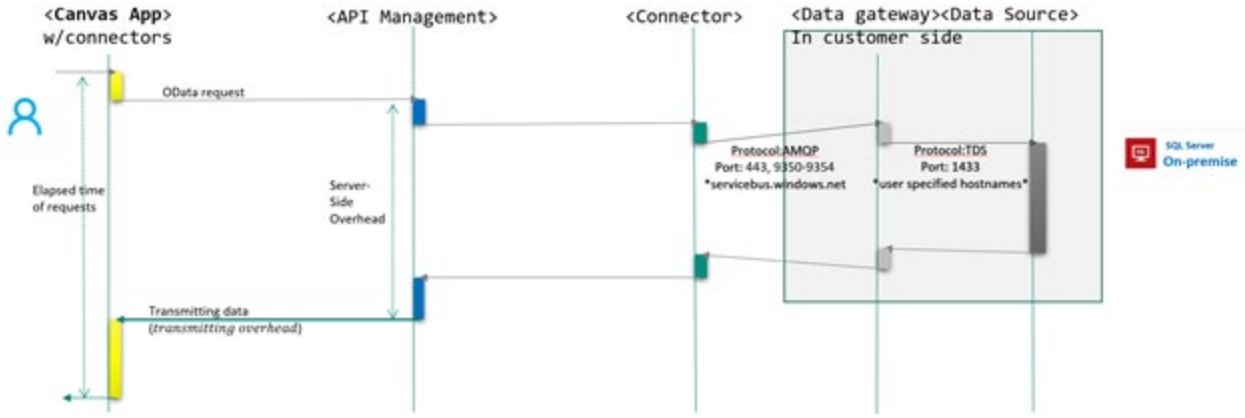


Figure 2 - How OData calls travel to on-premises SQL data source via an on-premises Data Gateway

Thick client or excessive requests: some canvas apps created a formula to perform client-side Group By, Filter By, and JOIN operations. Although the canvas app can perform such operations, they will require CPU and memory resources from the client device. Depending on the size of the data, these operations require additional scripting time on the client side, in addition to increasing the client's JS heap size. Keep in mind that each lookup data call must also travel to the data source via the data gateway. In this case, the number of data calls is crucial.

Unhealthy on-premises data gateway: Because organizations can define multiple on-premises data gateway nodes, all configured nodes must be healthy, and the on-premises data gateway service must be operational. If one of the nodes was unavailable, data requests to that node would not return a satisfactory result in a reasonable amount of time, but would instead return a 'unreachable' error message after some time.

The location of the on-premises data gateway: In order to interpret OData requests, the data gateway must make some network calls to on-premises data sources. For example, a data gateway must understand the data entity schema in order to convert OData requests into SQL DML (data manipulation language) statements. However, if the data gateway is configured on another continent with a high network latency between the data gateway and the SQL instance, it will generate additional overhead.

Scalability: A high volume of data access onto the on-premises data gateway is expected in some enterprises. In this case, a single node of the on-premises data gateway could be a bottleneck when dealing with a high volume of requests. One node of the on-premises data gateway can handle up to 200 concurrent connections. If all of these concurrent connections are actively running queries, other requests will be waiting for an available connection.

- Instead of doing Group By, Filter By, JOIN operations at the PowerApps client-side, use the View object in the SQL database for such operations. Maker or DBA (Database administrator) can create view(s) with only the columns needed by the canvas app. Then, in the canvas app, use the view entity.
- Ascertain that all on-premises data gateway nodes are in good health and that network latency between the nodes and the SQL instance is reasonable.
- Ascertain that all on-premises data gateway nodes are in good health and that network latency between the nodes and the SQL instance is reasonable.

A view is a named database query used to display data from Power Apps. A view typically joins multiple tables and can return subsets of columns. It has clear advantages, including:

- SQL Server's ability to retrieve data in the quickest and most efficient manner possible by utilizing indexes and other database optimizations.
- Better delegation support/the ability to search across multiple tables - Without views, we must join data from Power Apps by calling the Lookup function. If we need to filter data by content across multiple tables, the formulas we use are typically non-delegable and will not return all of the expected data.
- Ease of use - It can be difficult to construct the correct formula to join multiple tables with Lookup from Power Apps, especially when traversing three or more tables. Furthermore, related tables frequently share the same column names, necessitating a thorough understanding of disambiguation syntax.

When we need to join tables and display details from related tables, the most efficient way to do so is with a view. This post explained how to display related data on a gallery control and how to configure an app to update a record that a user selects via a gallery control that is linked to a view.

Shadow IT

Shadow IT refers to the use of IT-related hardware or software by a department or individual without the knowledge of the organization's IT or security groups. Cloud services, software, and hardware are all possible inclusions. (What Is Shadow IT?, 2022)

Today's main source of concern is the rapid adoption of cloud-based services. The rise of shadow IT has been accelerated by the consumerization of information technology. Users have become accustomed to downloading and utilizing cloud-based apps and services to help them with their work.

What is the rise in shadow IT trying to tell us?

1. The need for highly customizable software, flexible enough to be tweaked to any specific workflow
2. IT departments should be able to provide solutions quicker
3. There is high demand for technology that responds to market changes & user needs

Low-Code and No-Code platforms have made it easier than ever for anyone to create an application that solves a business problem without considering the implications for the entire organization.

“With simple subscription billing that can be tucked into Opex budgets, low-code platforms can enable non-IT teams to build their own solutions,” says MacQuarrie. “Depending on the ease-of-use of the platform, these teams could be far outside the official IT app-dev process, meaning they may not be considering important technical, procedural and industry-specific policies. When IT loses visibility and control we have Shadow IT. At best it’s annoying and at worst there are financial repercussions.”

Low-code platforms accelerate enterprise solution development. A low-code platform that has been approved by an organization's central IT department can be critical in combating shadow IT.

However it would be wrong to label low-code practices as a promoting tool for Shadow IT, it can also help the IT departments to decrease the impact of the shadow IT practices.

Central IT can find a platform that meets an enterprise's requirements after a thorough evaluation of leading low-code platforms. The platform can then be used by business users to build tools that comply with enterprise regulations, bridging the gap with IT. (Sasi, 2022)

Modern low-code platforms can aid in the fight against shadow IT by providing:

1. A digital sandbox environment in which you can safely develop and test business applications.
2. A development platform with a graphical programming interface that is simple to learn and use for users of any technical skill level.
3. You can use drag-and-drop graphics to create robust, customizable databases and optimized workflows.
4. Applications that can be updated and modified in real time to meet the needs of the user
5. Separate development, testing, and production environments, as well as straightforward deployment
6. Scalability allows applications to be extended from personal use to departmental use to organization-wide use.
7. APIs and integration options allow you to seamlessly access data from other enterprise-wide systems.
8. Security and authentication out-of-the-box
9. Enterprises have both on-premises and cloud options, allowing you to select the solution best suited to your office infrastructure.

CHAPTER 8: LIMITATIONS AND FUTURE UPDATES

1. Limitations

There are a few limitations to creating an application on PowerApps: (Bordoli, 2022)

- Power Apps enables users to create simple forms with its low-code services quickly. Forms with complex business logic or forms that evolve, on the other hand, can cause issues due to a limited ability to manage the code base and track changes.
- The Power Platform's IDE is web-based. Users create all forms in a web browser rather than a desktop application, so they cannot create forms offline or when they are not connected to the internet. Web-based IDEs generally lack the sophistication of desktop IDE environments.
- Users of Power Apps must create multiple versions of their apps. Responsive forms can scale to some extent, but they necessitate a trade-off. An app optimized for phones and tablets, for example, necessitates two versions, one for each device.
- Power Apps has a 2,000 item restriction from a connected data source like SharePoint, SQL, or Oracle.
- Throughput varies depending on the connection. Attempting to read or write hundreds of items from SharePoint lists, SQL databases, or Excel workbooks, on the other hand, can exceed allowed thresholds and result in failures. In the per-app licensing plan, Power Apps, for example, allows 1,000 connector requests per 24-hour period. The throughput limits of each connector vary and can be complicated.
- Microsoft restricts Power Apps attachment control to SharePoint or Dataverse on the back end, so platforms such as OneDrive or SQL cannot be used as the target document store. Furthermore, the maximum upload size is 50 MB.

- Each app in Power Apps where users execute business logic, such as field validation or field calculations, necessitates creating and maintaining the same logic across apps. Larger, more complex forms with many lines of embedded programming logic may result in unpredictability when users add or change program lines to meet changing business needs.

However, most of above limitations did not limit our project, which means the boundaries of the limitations are significant enough to hold a medium-scale Content Management app. However, even though these limitations did not affect us much, there were other limitations for our app, like not being able to create a join between two tables with a build-in function in PowerApps. Although this limitation does not prevent us from doing what we want to do, it makes things slower than they should be; whether you use collections or the LookUp formula to create your gallery, it takes time to respond.

Another limitation for me was the location problem; since our database runs in-campus, I need to use a VPN connection to be able to access the database, which would further slowdown the connection reactions

2. Future Updates

Further updates can be made to improve the quality of the app and/or expand the scale of the product. With PowerApps's Model-Driven Apps, it is possible to create a great app that contains multiple apps as its component. Our app is created only to insert the Capstone Annals of the Computer Science Department. It is possible and easy to expand this scale to other departments and even other schools with the capabilities of the database. New flows for sending reports to the Department head, such as e-mail or yearly reports of the faculties about how many capstones they were chaired or advised can be created.

CHAPTER 9: CONCLUSION

We live in a time of rapid change. The unprecedented acceleration of digital and business transformation – only fueled by the pandemic – has put enormous strain on almost all organizations worldwide.

Low-code and no-code technologies appear to answer at least some of today's challenges. The economic consequences of the COVID-19 pandemic have validated the value proposition of low-code. Low-code capabilities that support remote work functions, workflow automation, and rapid app development are increasingly crucial for the survival of organizations of all sizes.

As a result of this research, it is possible to conclude that creating content with low code will significantly reduce the time and effort required by organizations while also reducing the workload on employees. Most tasks in organizations are repetitive and use the same input for different environments to keep track. Businesspeople can create their task solution apps with the help of low code platforms to make their jobs easier. Also, while creating content with low code platforms is simple for most businesspeople, making good use of these platforms requires some education and training.

Even professional developers would agree that using low code platforms to create content is an excellent way to serve customers. Reducing the resources required to create a product in a shorter period allows developers to be more productive and efficient.

Finally, low code development will advance even further than it is now, and it will be possible to create any content using low code in the future. So, it is critical to start familiarizing our future businesspeople and developers with low code as it is critical for the next-generation development.

REFERENCES

Ahn, J., 2022. Considerations for optimized performance in Power Apps. [online] Powerapps.microsoft.com. Available at: <https://powerapps.microsoft.com/en-us/blog/considerations-for-optimized-performance-in-power-apps/> [Accessed 14 April 2022].

Bordoli, J., 2022. What are the top Microsoft Power Apps limitations?. [online] SearchContentManagement. Available at: <https://www.techtarget.com/searchcontentmanagement/tip/What-are-the-top-Microsoft-Power-Apps-limitations> [Accessed 26 February 2022].

Chatterjee, S., 2022. Introduction to Collections in PowerApps and Their Uses. [online] EPC Group. Available at: <https://www-epcgroup-net.translate.goog/what-is-collections-in-powerapps/?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=op,sc> [Accessed 14 April 2022].

Cisco. 2022. What Is Shadow IT?. [online] Available at: <https://www.cisco.com/c/en/us/products/security/what-is-shadow-it.html> [Accessed 14 April 2022].

Docs.microsoft.com. 2022. Filter, Search, and LookUp functions in Power Apps (contains video) - Power Apps. [online] Available at: <https://docs.microsoft.com/en-us/power-apps/maker/canvas-apps/functions/function-filter-lookup> [Accessed 14 April 2022].

Docs.microsoft.com. 2022. Start building apps - Power Apps. [online] Available at: <https://docs.microsoft.com/en-us/power-apps/maker/> [Accessed 13 April 2022].

Docs.microsoft.com. 2022. What is Power Apps? - Power Apps. [online] Available at: <https://docs.microsoft.com/en-us/power-apps/powerapps-overview> [Accessed 13 March 2022].

Person, D., 2022. Council Post: How Low-Code And No-Code Tools Can Help Stave Off Shadow IT And The Great Resignation. [online] Forbes. Available at: <https://www.forbes.com/sites/forbestechcouncil/2022/02/09/how-low-code-and-no-code-tools-can-help-stave-off-shadow-it-and-the-great-resignation/?sh=284ab9352106> [Accessed 14 April 2022].

Sasi, R., 2022. Say goodbye to shadow IT with low-code | Decode - A publication by Zoho Creator. [online] Decode - A publication by Zoho Creator. Available at: <https://www.zoho.com/creator/decode/say-goodbye-to-shadow-it-with-low-code> [Accessed 14 April 2022].

Stangarone, J., 2022. Pros and cons of low-code development platforms - mrc's Cup of Joe Blog. [online] mrc's Cup of Joe Blog. Available at: <https://www.mrc-productivity.com/blog/2019/03/pros-and-cons-of-low-code-development-platforms/> [Accessed 14 April 2022].

Techopedia.com. 2022. What is HyperCard? - Definition from Techopedia. [online] Available at: <https://www.techopedia.com/definition/1209/hypercard#:~:text=HyperCard%20was%20released%20in%201987,and%20the%20relationships%20between%20them.> [Accessed 13 March 2022].

Tozzi, C., 2022. The Evolution of Low-Code/No-Code Development. [online] ITPro Today: IT News, How-Tos, Trends, Case Studies, Career Tips, More. Available at: <https://www.itprotoday.com/no-codelow-code/evolution-low-codeno-code-development> [Accessed 13 April 2022].

Yokoten.co. 2022. YOKOTEN. [online] Available at: <https://www.yokoten.co/post/pros-and-cons-of-low-code-no-code-platforms> [Accessed 14 April 2022].

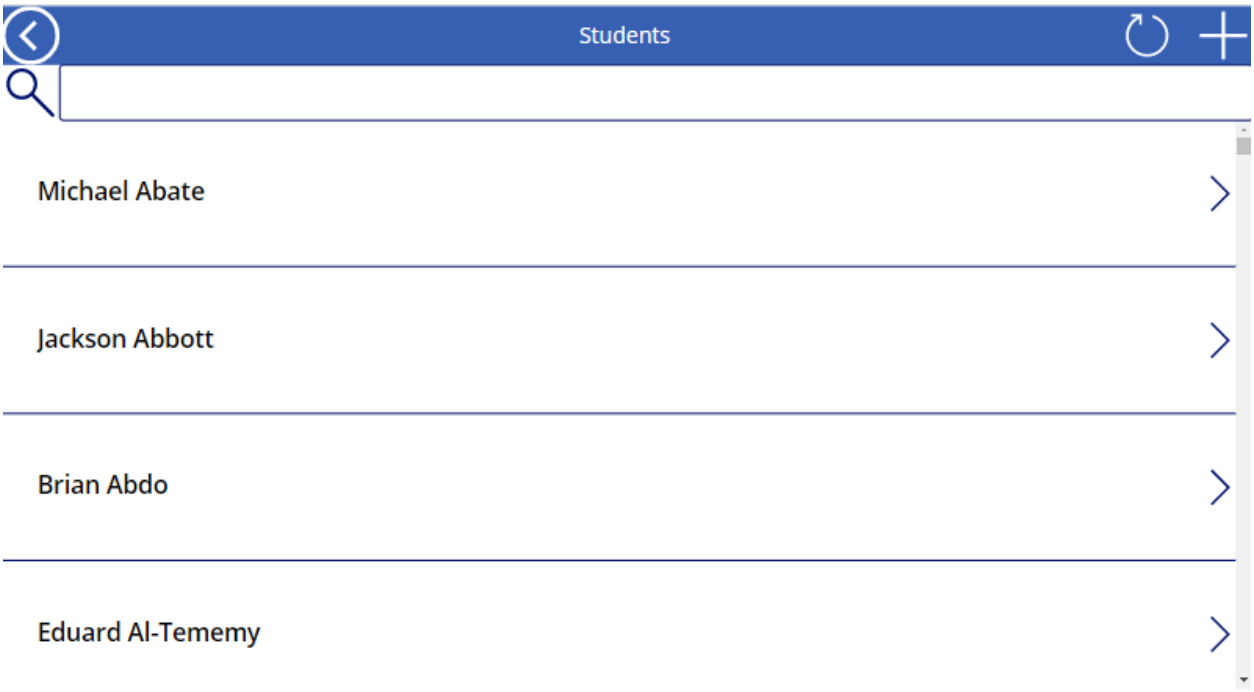
Powerappsguide.com. 2022. [online] Available at: <http://powerappsguide.com/blog/post/lookup-related-values-with-sql-views> [Accessed 14 April 2022].

APPENDIX A.

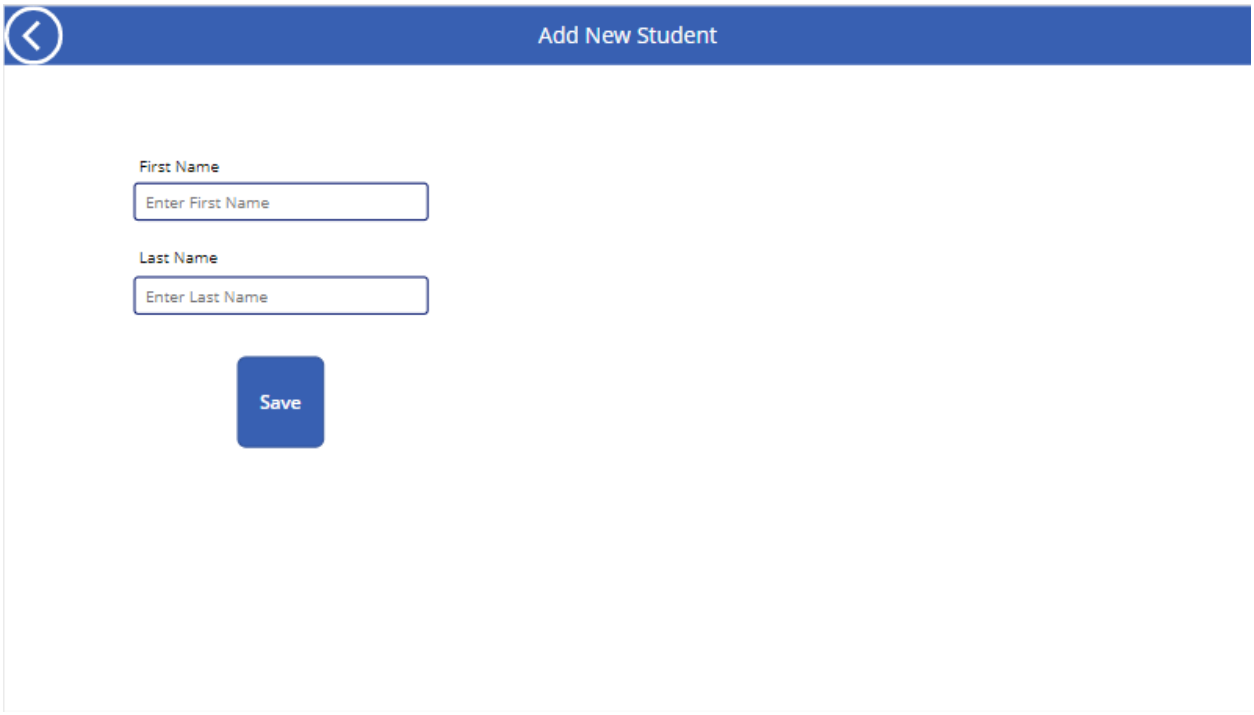
Screen Shot 1.1 - Homepage – Navigation page

This is the main page of our site. Here we have our navigation buttons to go to the pages we want to. The good part of using low code development is that there is no need for coding; we can navigate through pages with just a basic function.

Screen Shot 2.1 - Students Page – List of students

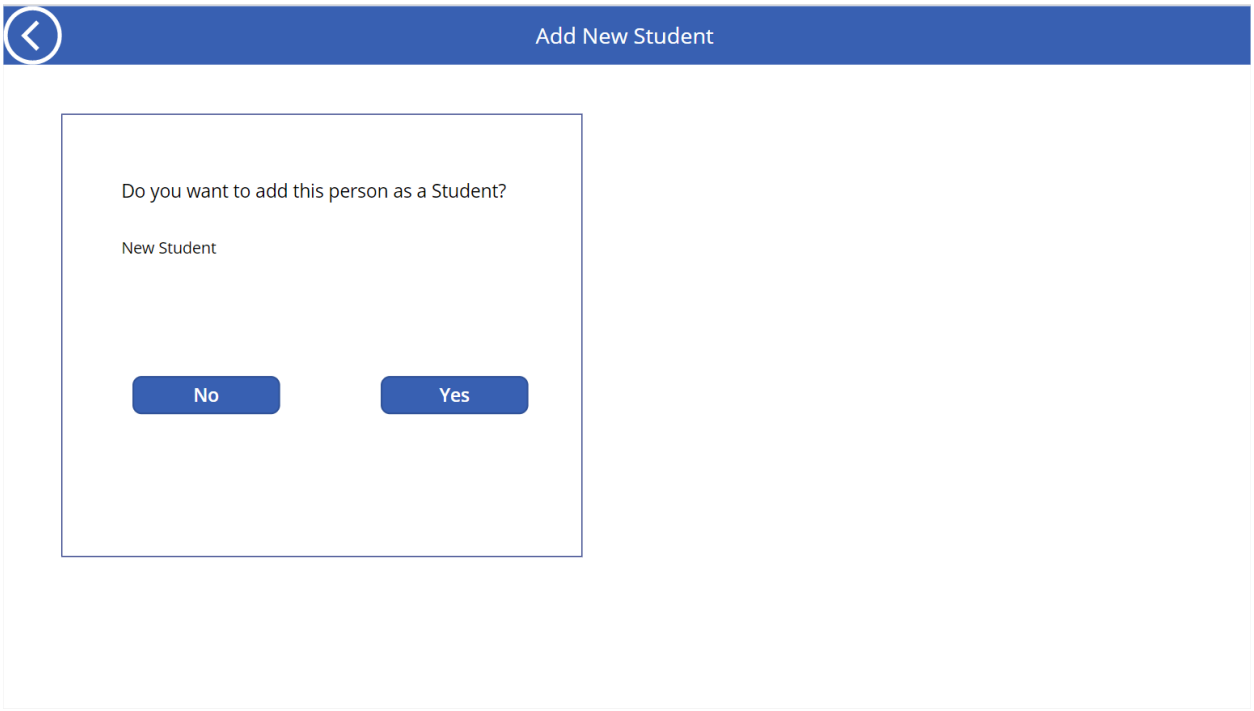


Screen Shot 2.2 - Add New Person Page



This page allows us to insert new people into the system with the help of Power Automate flows. According to the page you came from app changes the person's role and asks for different information about them. To insert a student user, enter the student's first name and last name and hit the save button. On this page, we have an if condition that looks into some variables to decide the role of the person inserted and act accordingly.

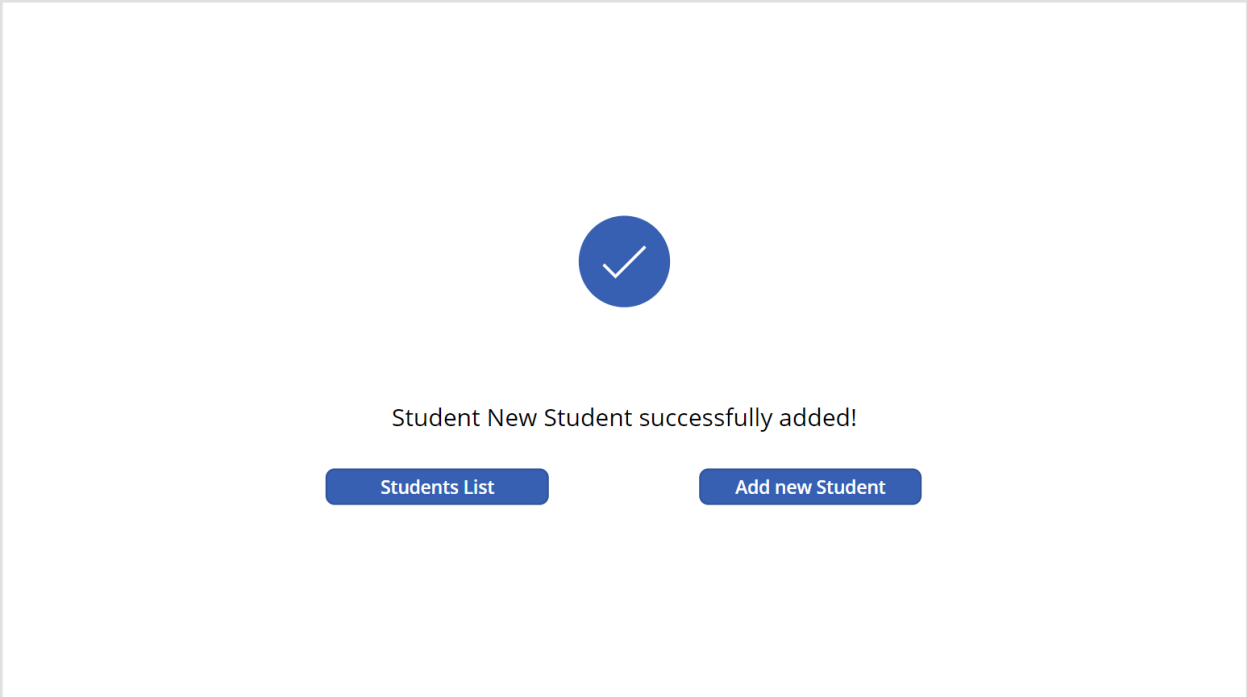
Screen Shot 2.3 - Add New Person Page



Runs the Automate to insert the Student, Professor, or Person according to the path taken

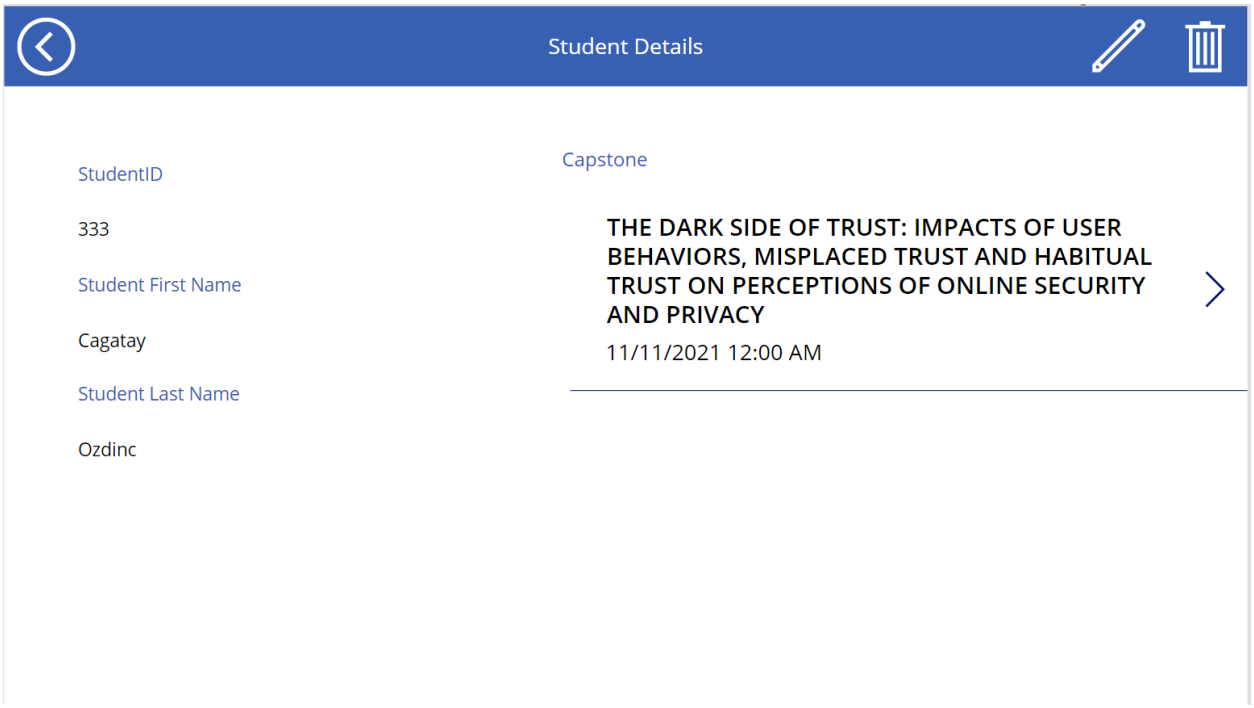
After the user hits the save button, the ordinarily invisible box becomes visible. When the user hits the yes, the student will get pushed to the database by the flow, and some variables for the next page are set.

Screen Shot 2.4 - Success Page – Insert Student



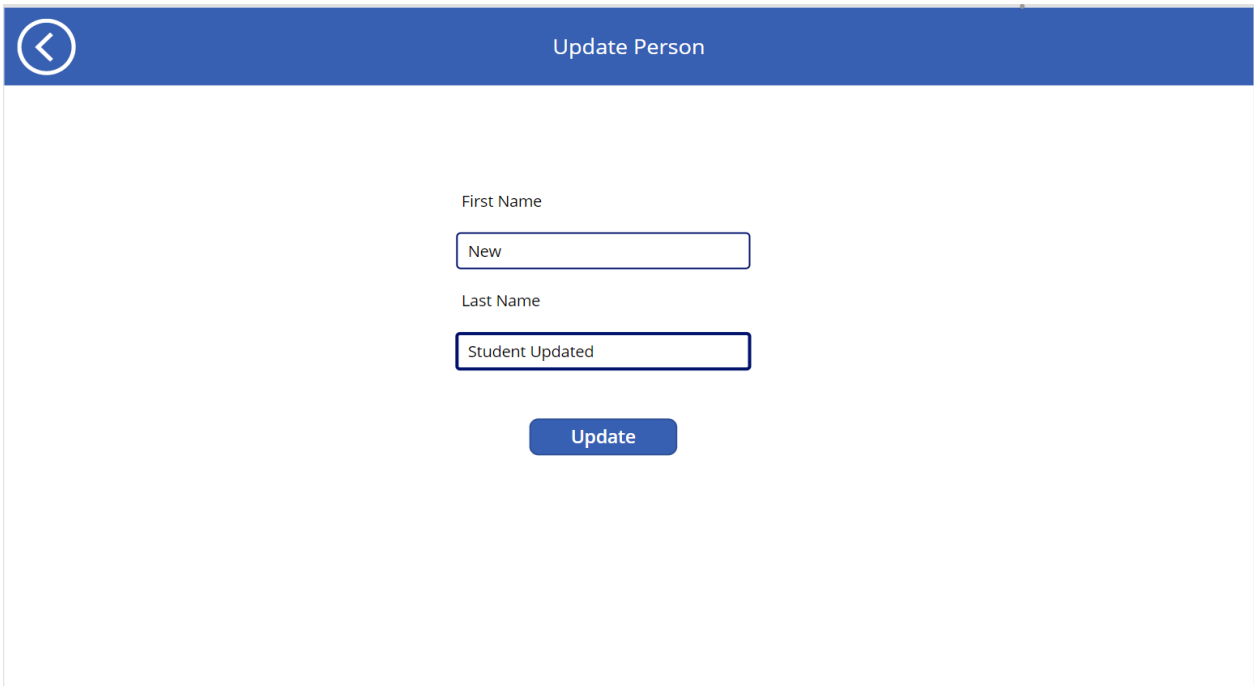
Works dynamically according to the operation, the status of the person (student, professor, person), and the name with the variables set on the previous page.

Screen Shot 2.5 - Student Details Page – Seeing the details

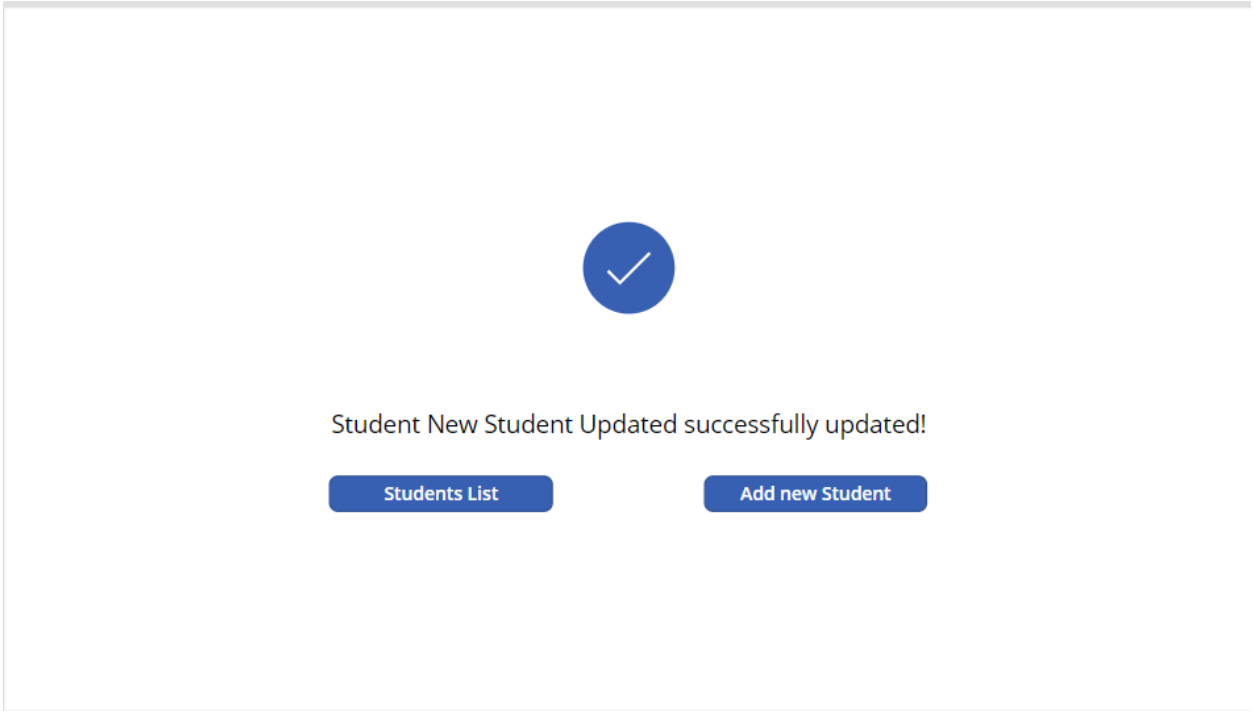


On the details page, we can see the Capstone authored by the student; with the help of the Filter function, we can click and navigate to the Capstone Detail page directly from this gallery.

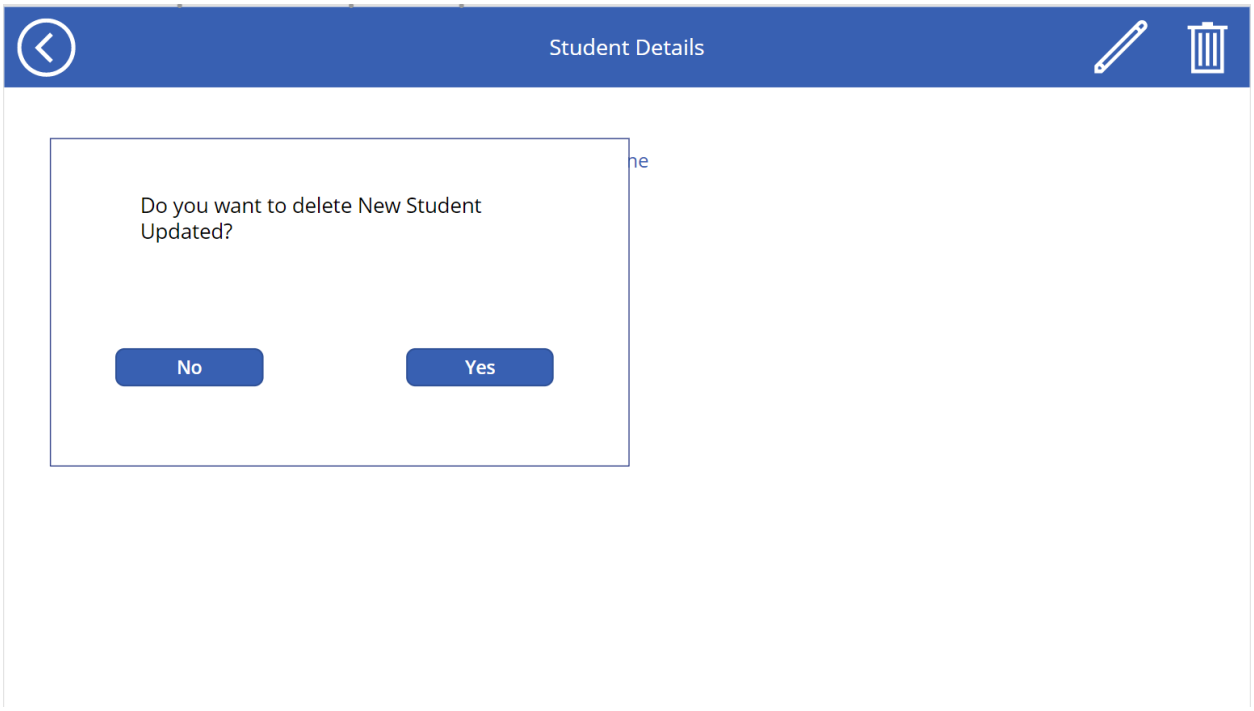
Screen Shot 2.6 - Update Person Page



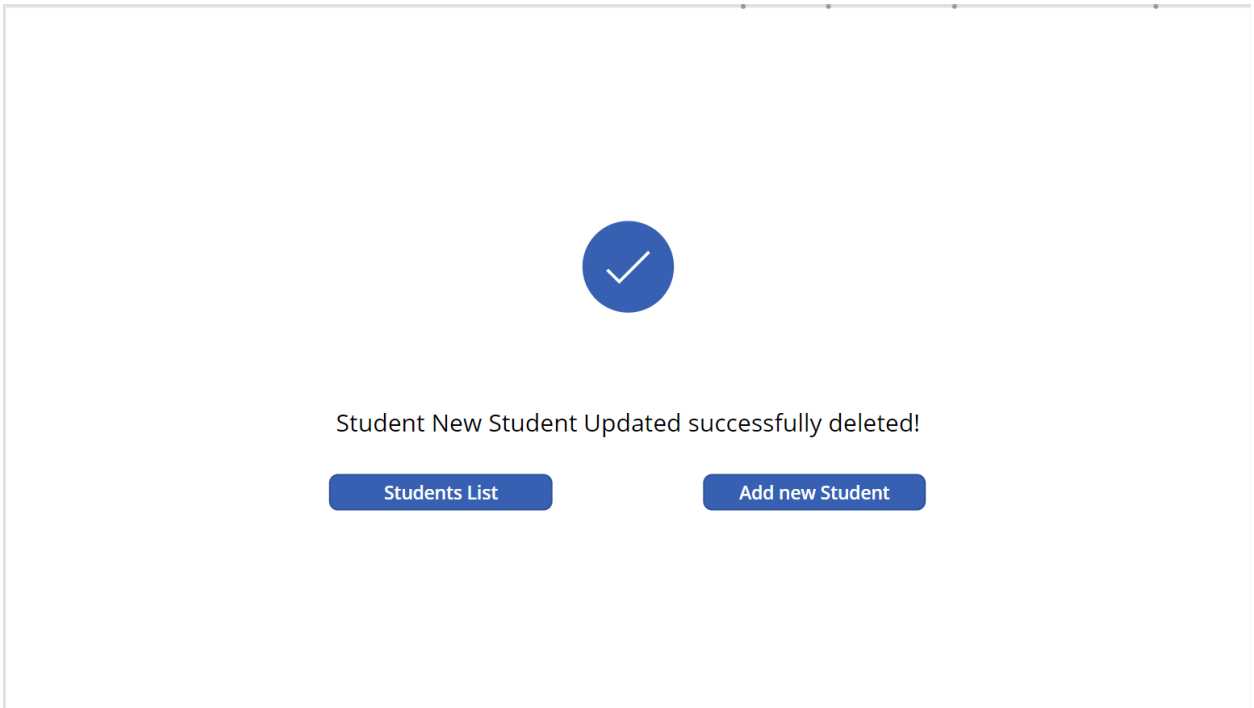
Screen Shot 2.7 - Success Page – Update Student



Screen Shot 2.8 - Student Detail – Delete student

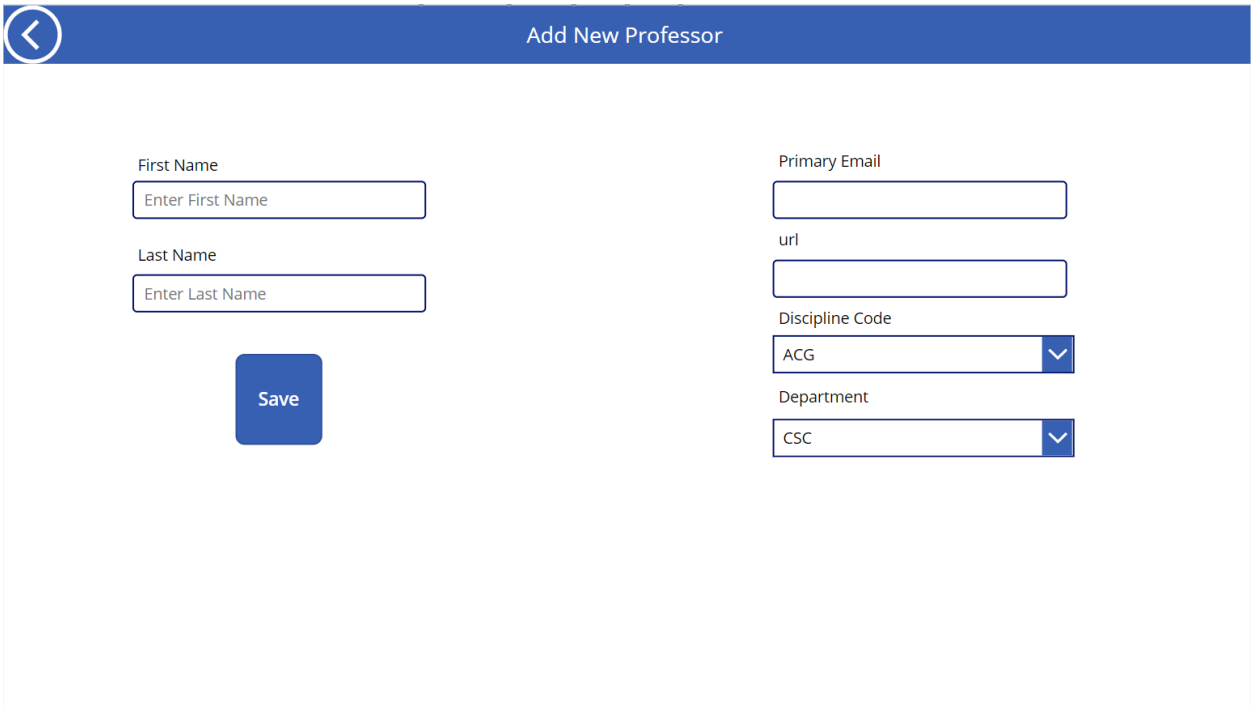


Screen Shot 2.9 - Success Page – Delete Student



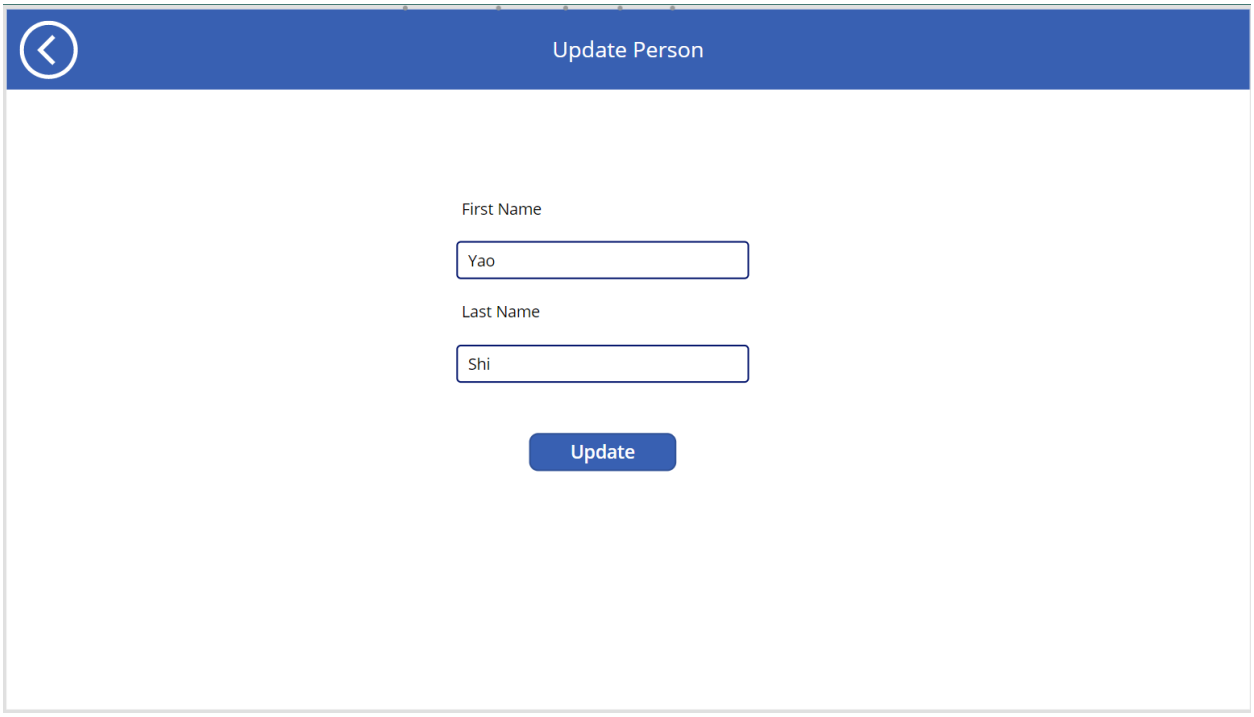
Screen Shot 3.1 - Professors List

Screen Shot 3.2 - Add New Professor

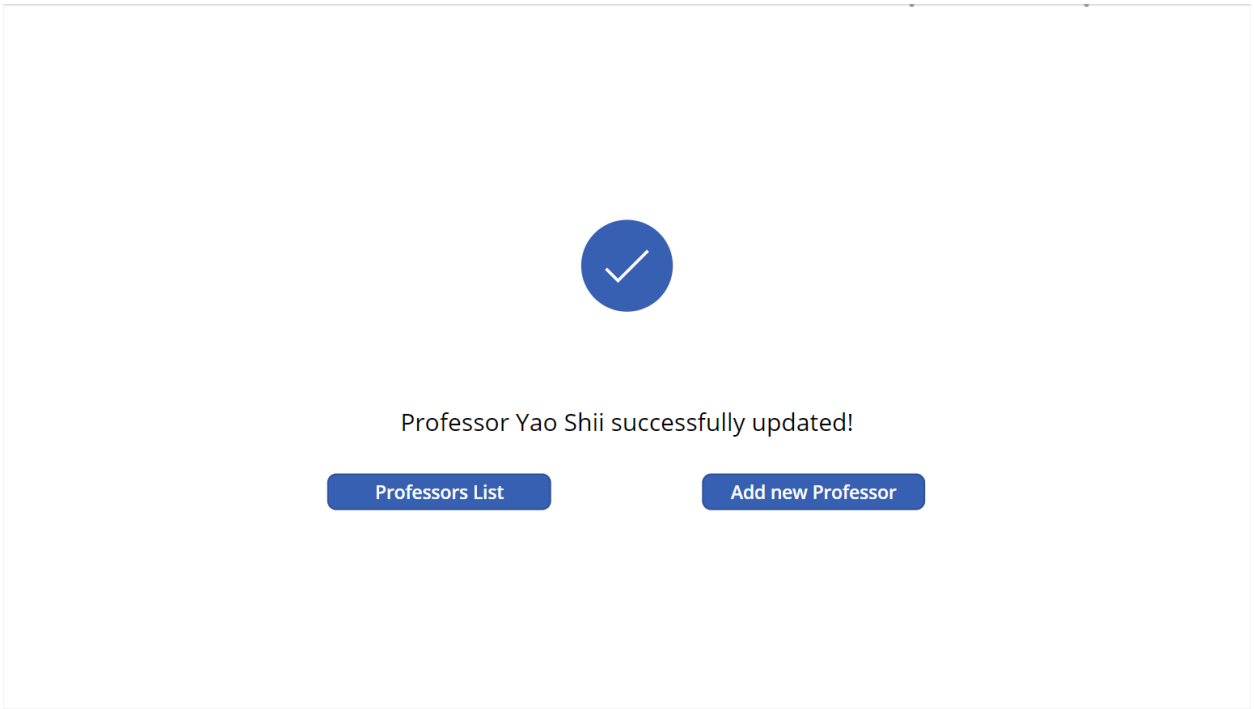


This page is the same page that we inserted the student, but now we have some extra text boxes to fill now that the role has changed to the professor.

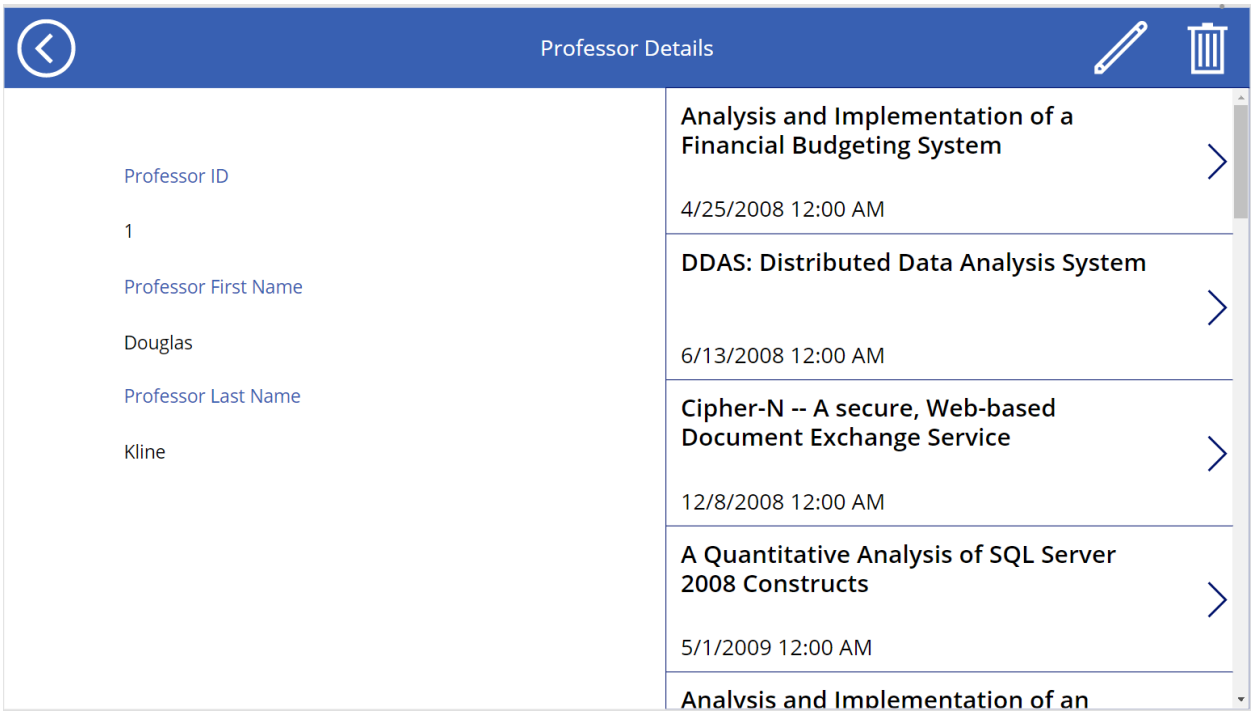
Screen Shot 3.3 - Update Professor



Screen Shot 3.4 - Success Page – Update Professor

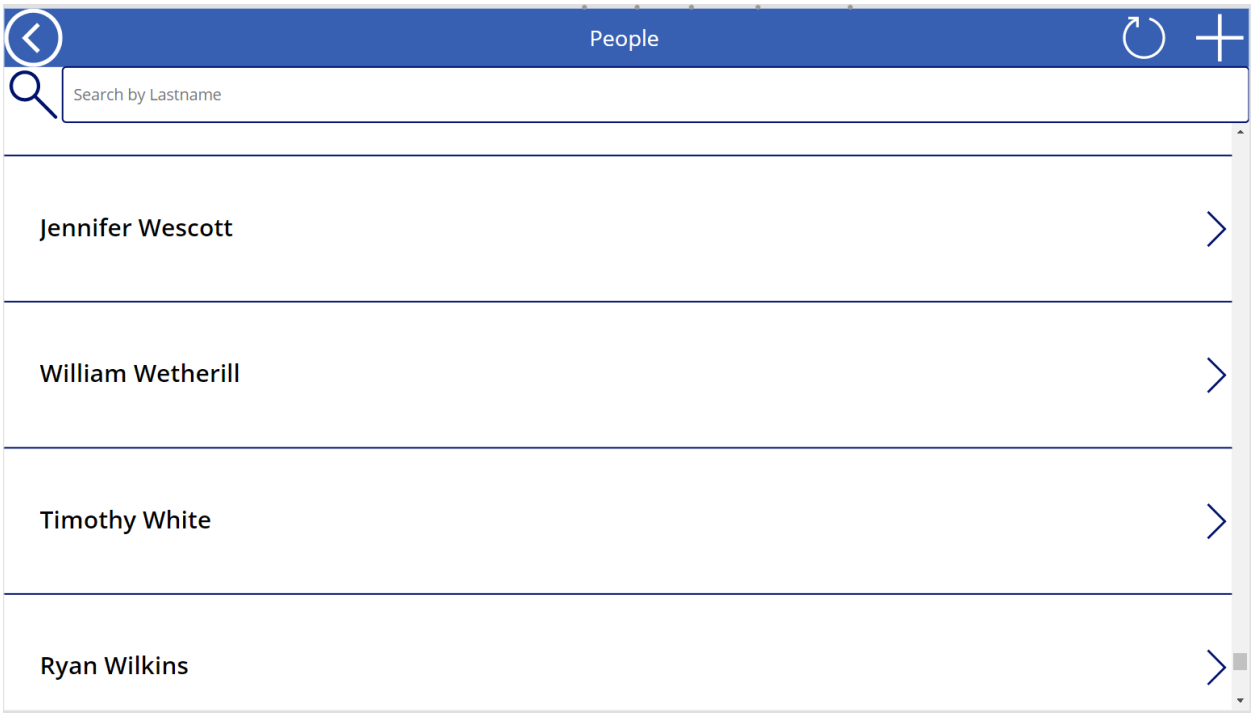


Screen Shot 3.6 - Professor Details

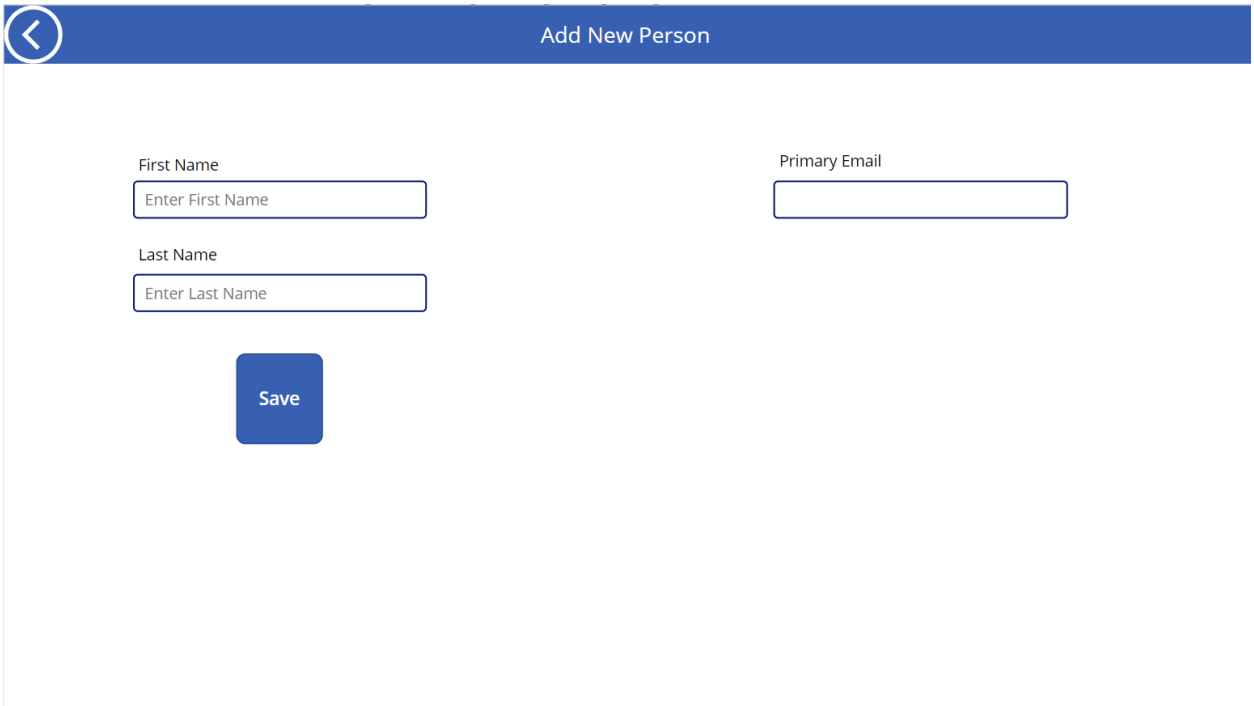


This page works like the student details page with the filter for the capstones that shows the capstones chaired by the professor.

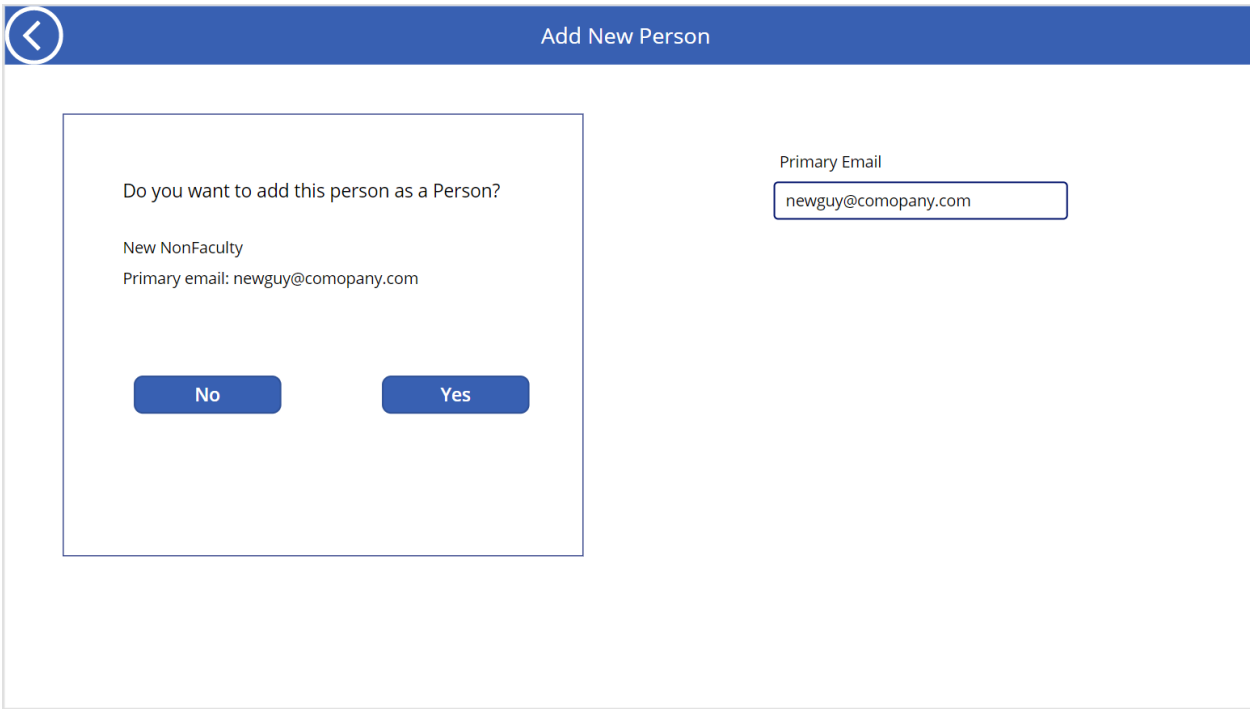
Screen Shot 4.1 - People List



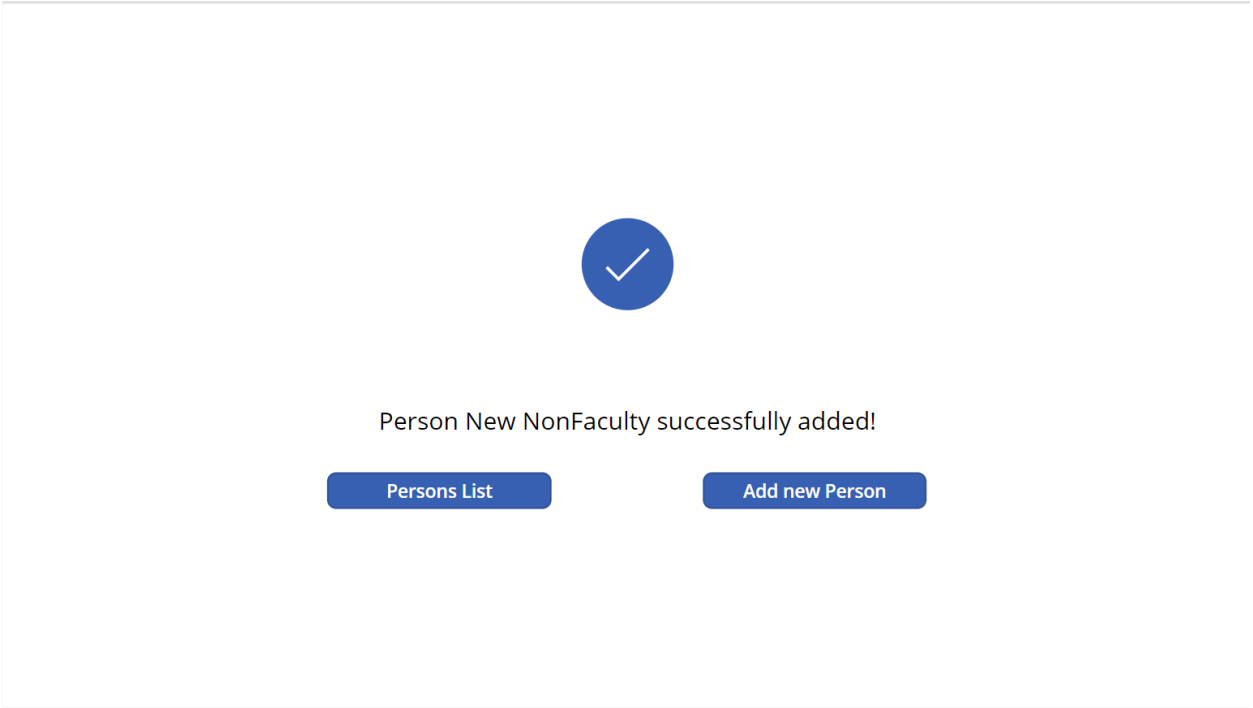
Screen Shot 4.2 - Insert Person



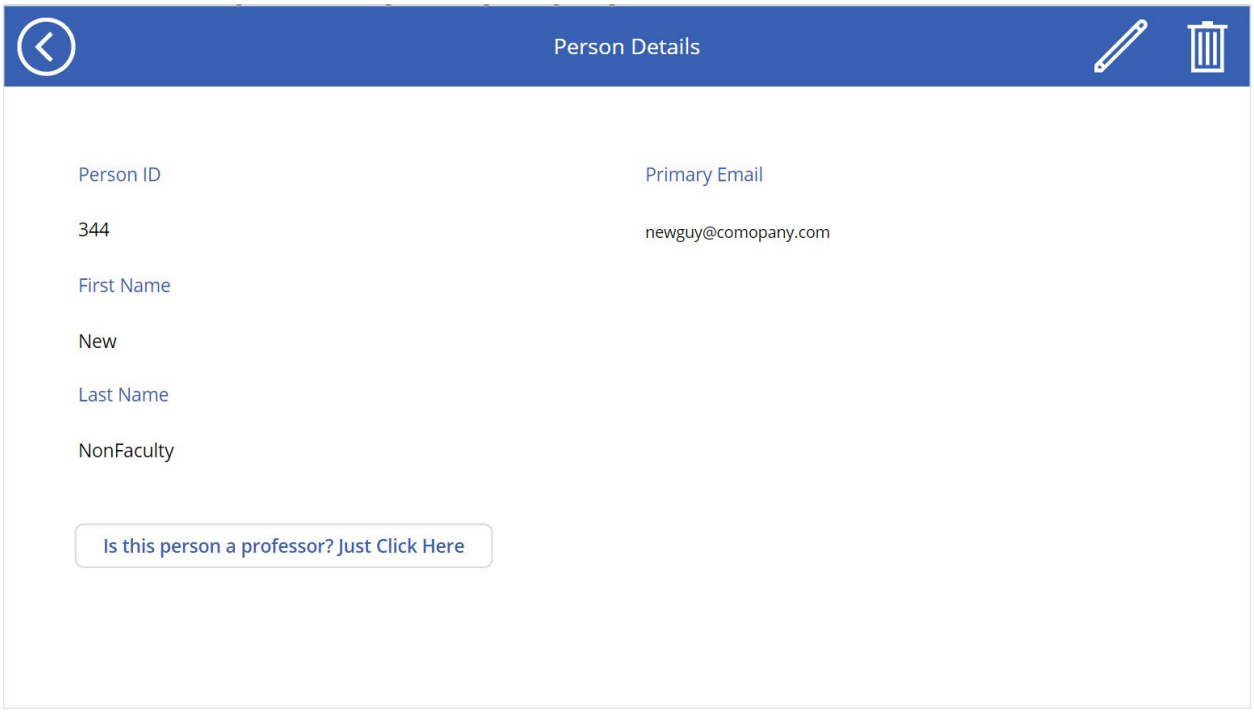
Screen Shot 4.3 - Insert Person



Screen Shot 4.4 - Insert Person Success

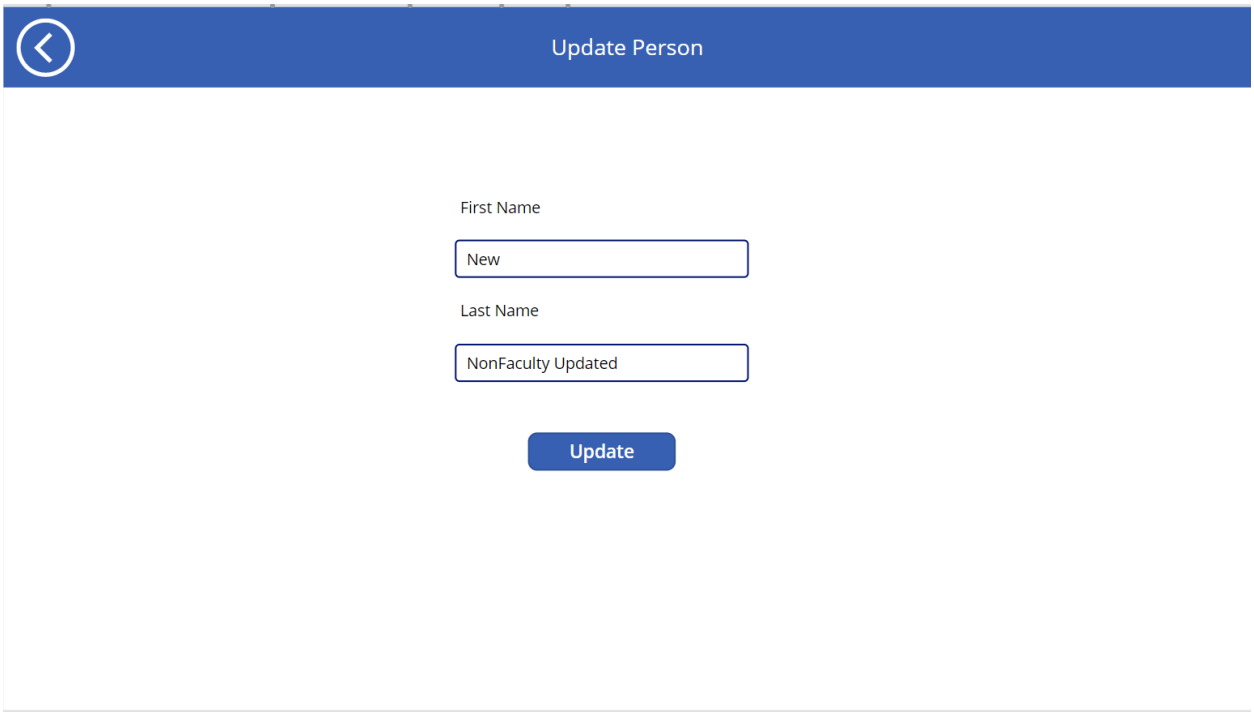


Screen Shot 4.5 - Person Details

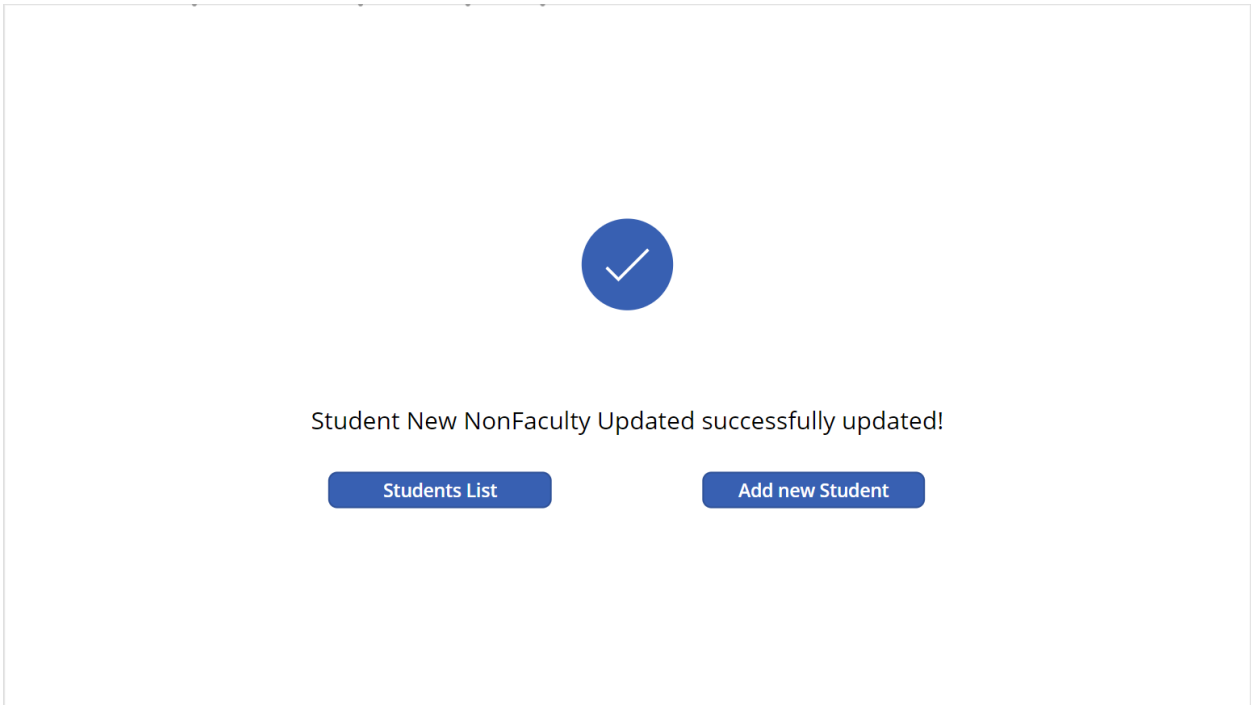


On the person details page, we have a button to change the person's role to the professor. Because it is possible to have a non-faculty committee member, it may be possible for the person to become a faculty or may be added as a non-faculty by mistake. So with just a click, we can fix this issue.

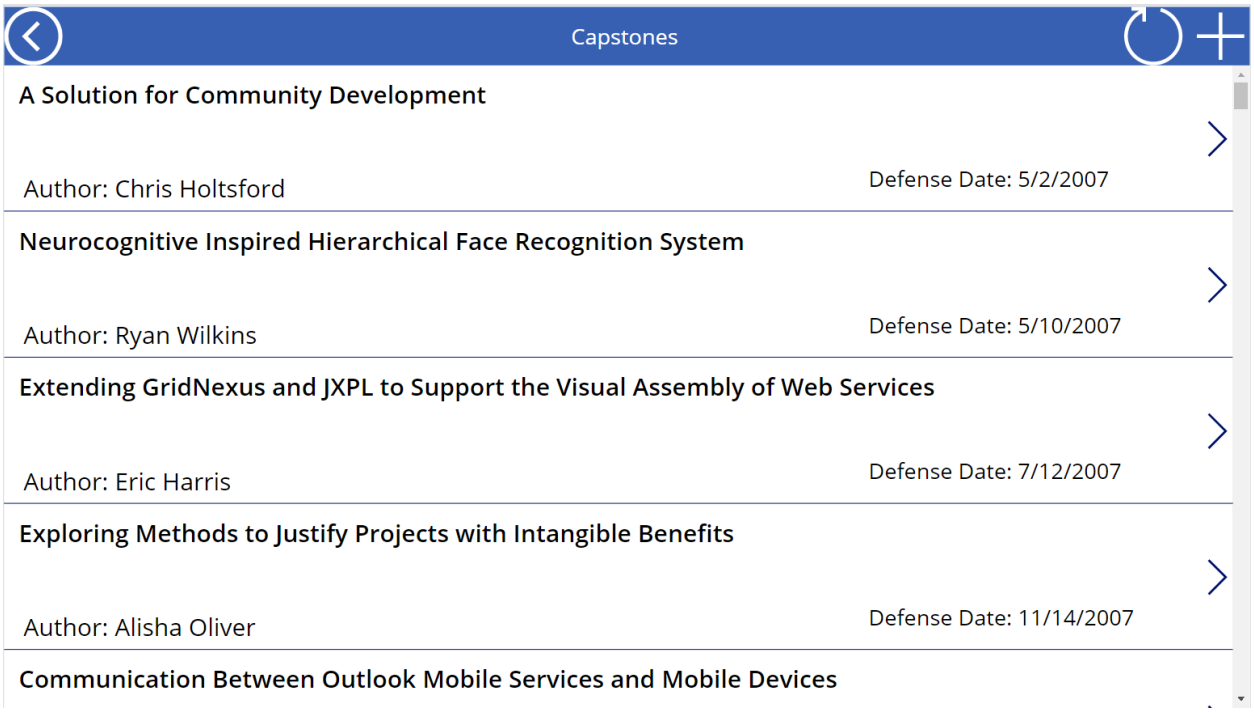
Screen Shot 4.6 - Update Person



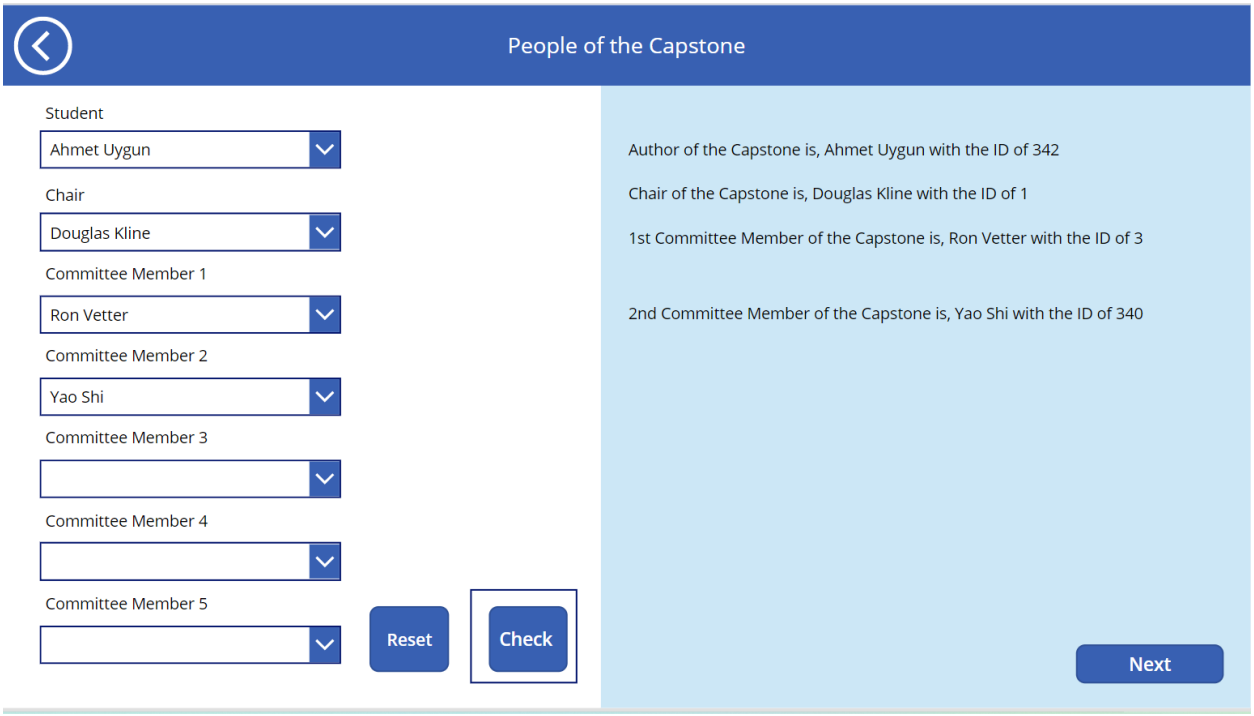
Screen Shot 4.7 - Person Update Success



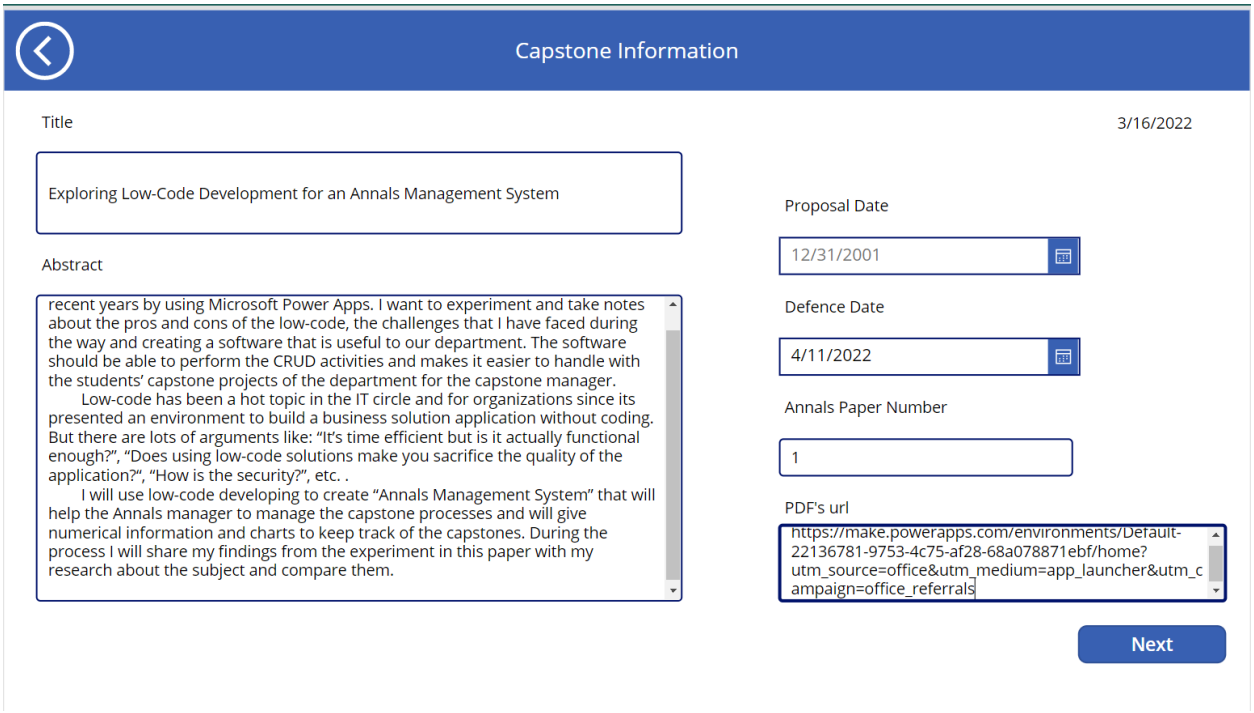
Screen Shot 5.1 - Capstones List



Screen Shot 5.2 - Insert Capstone

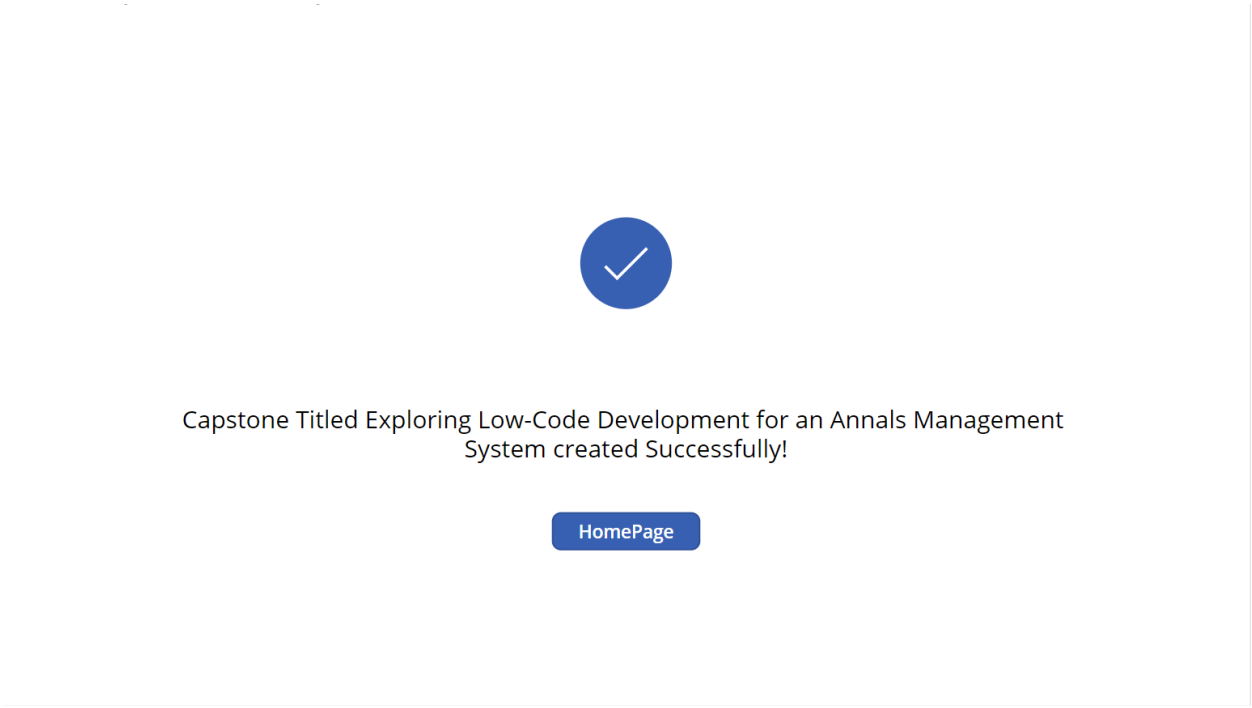


Screen Shot 5.3 – Insert Capstone2

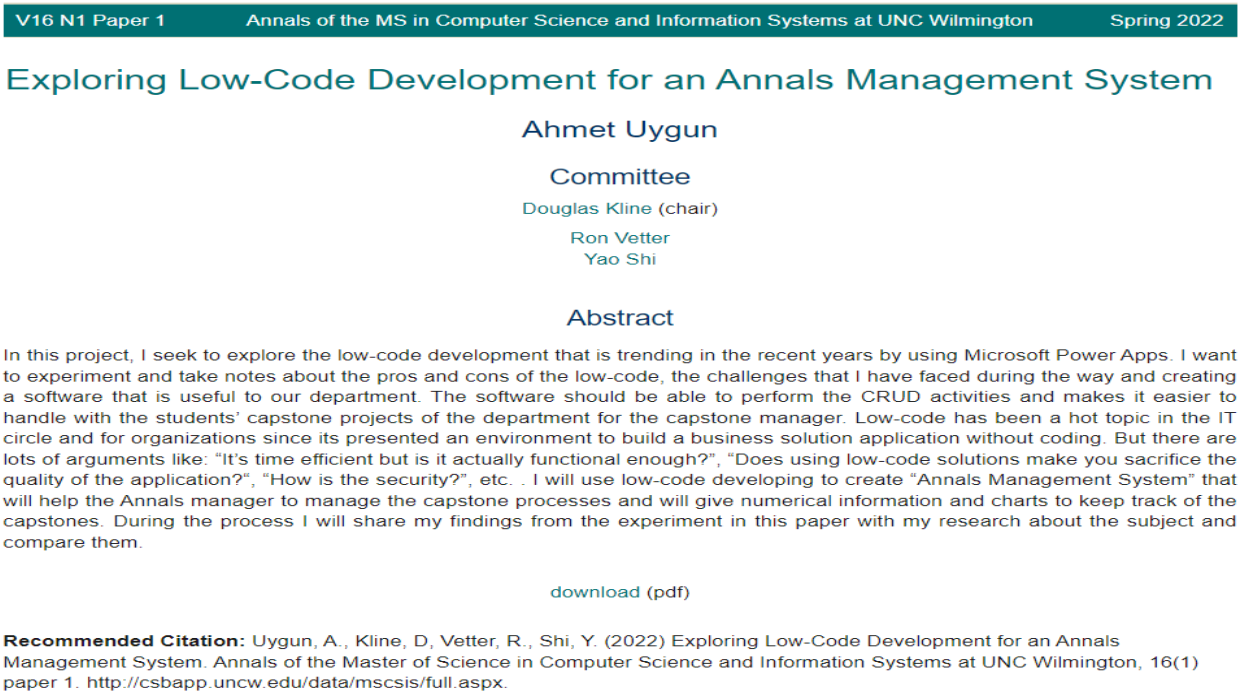


Screen Shot 5.4 – InsertCapstone3

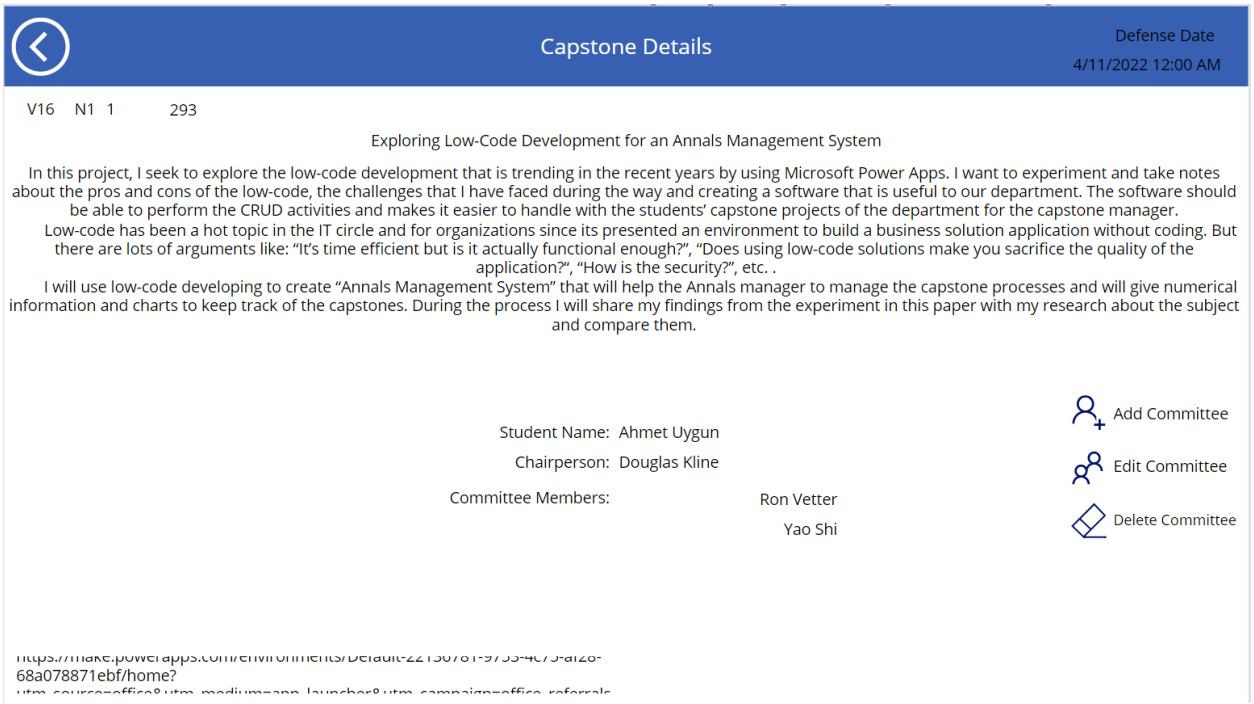
Screen Shot 5.5 - InsertCapstoneSuccess



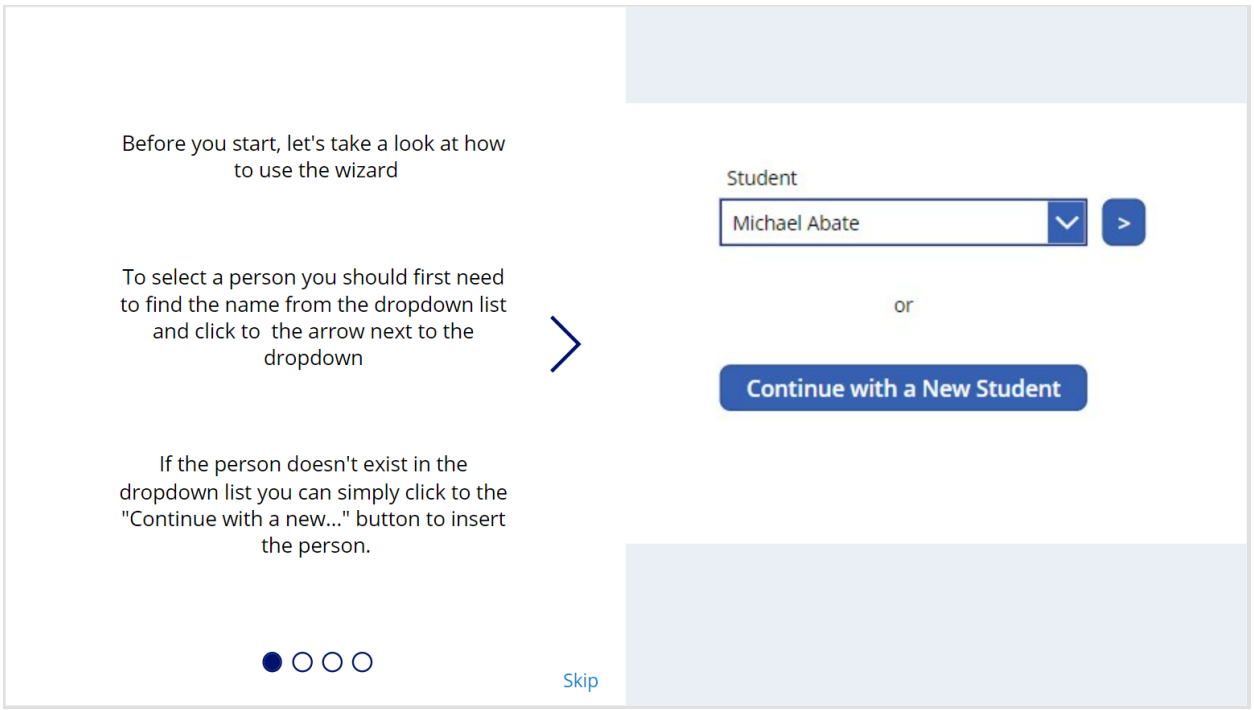
Screen Shot 5.6 – Annals Website



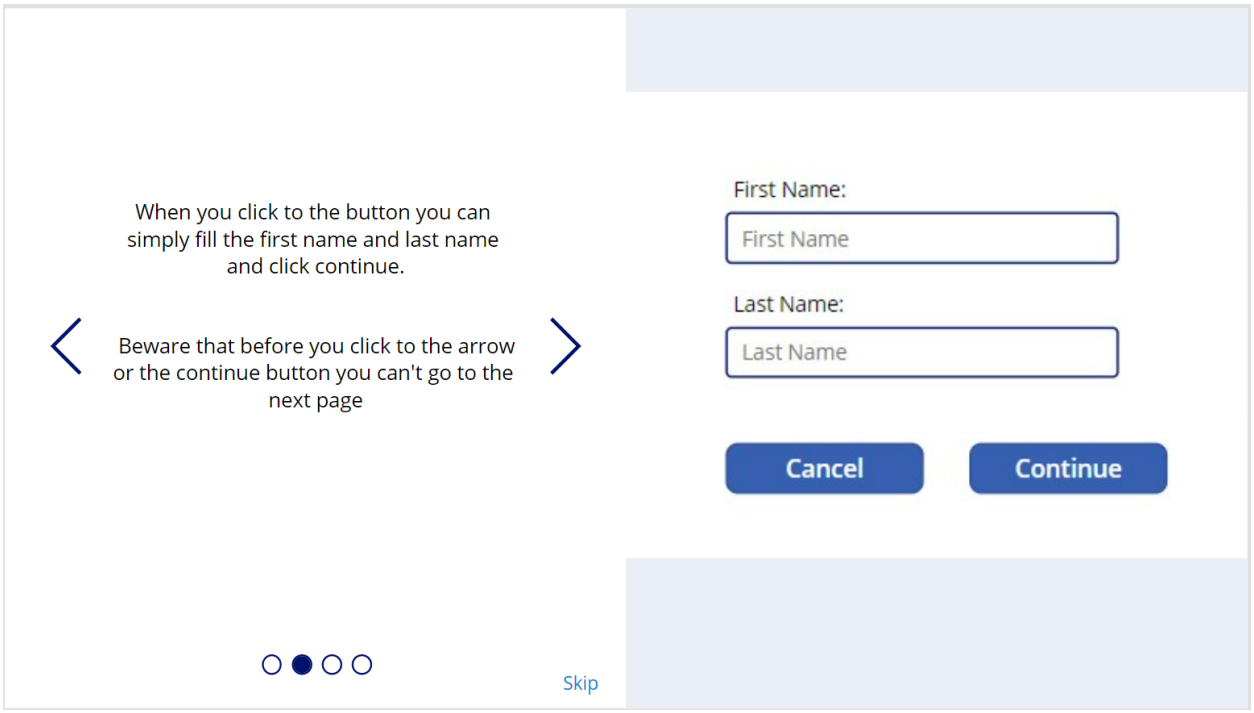
Screen Shot 5.7 – Capstone Details



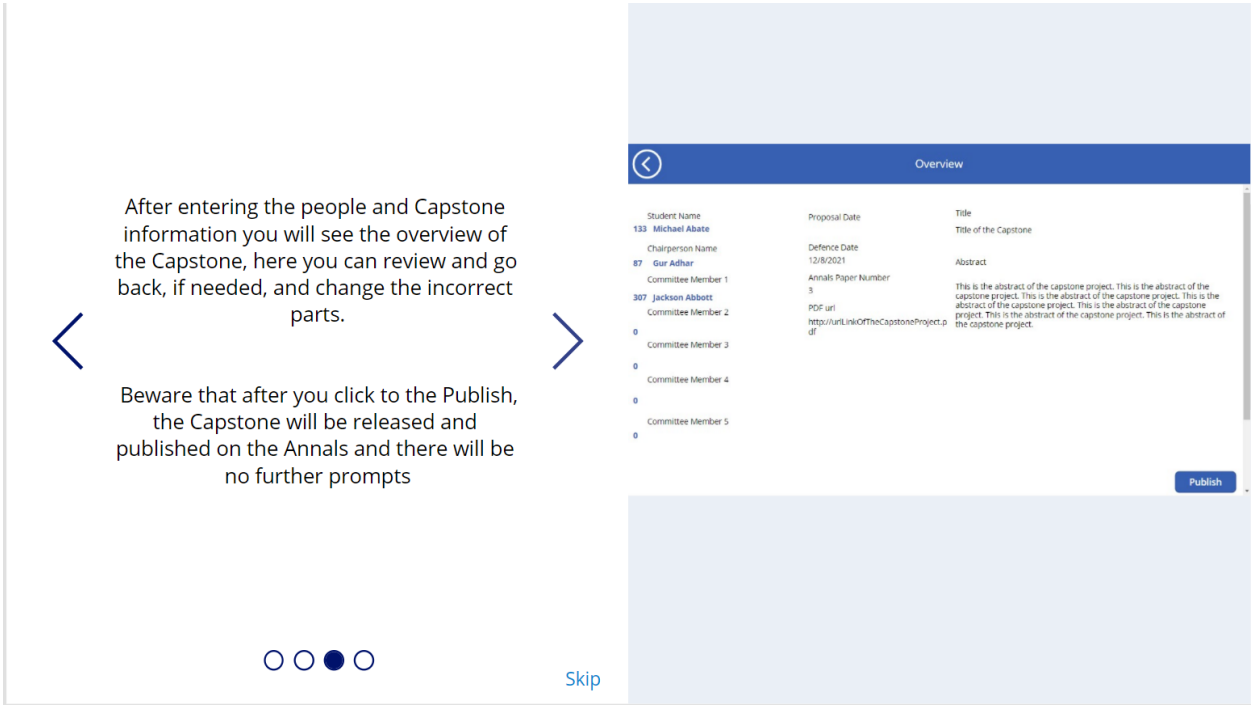
Screen Shot 6.1 – Wizard Tutorial



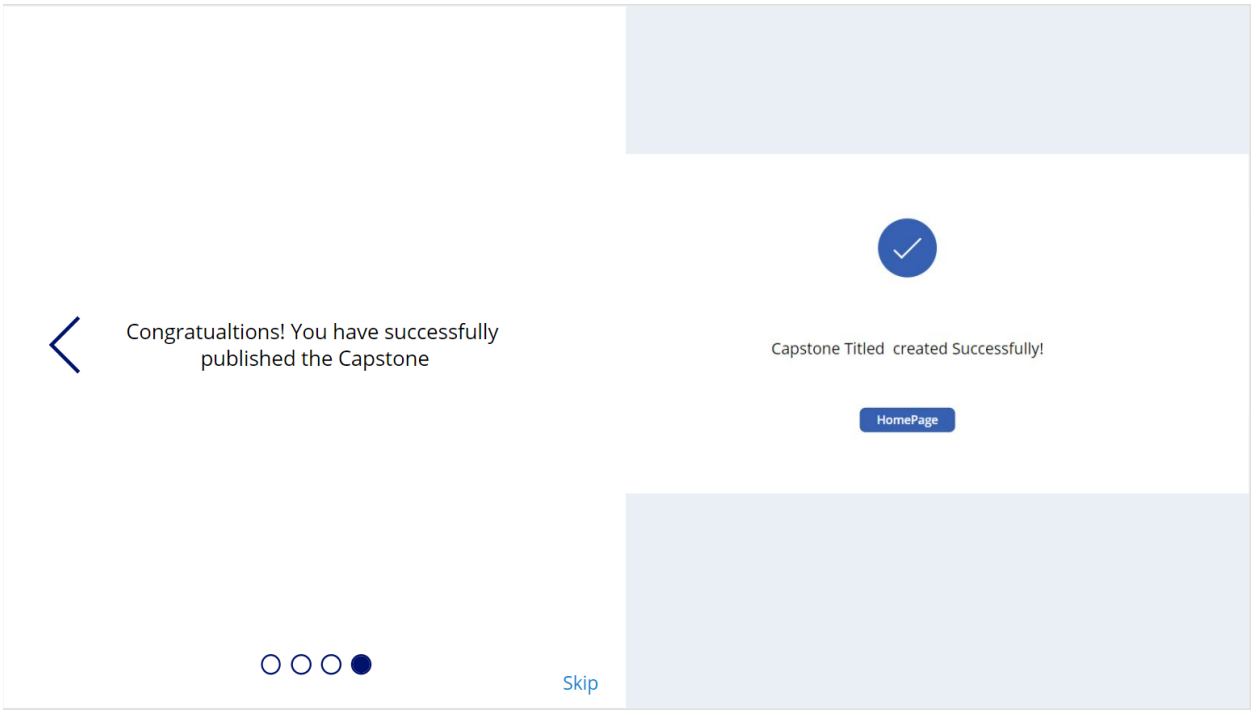
Screen Shot 6.2 – Wizard Tutorial 2



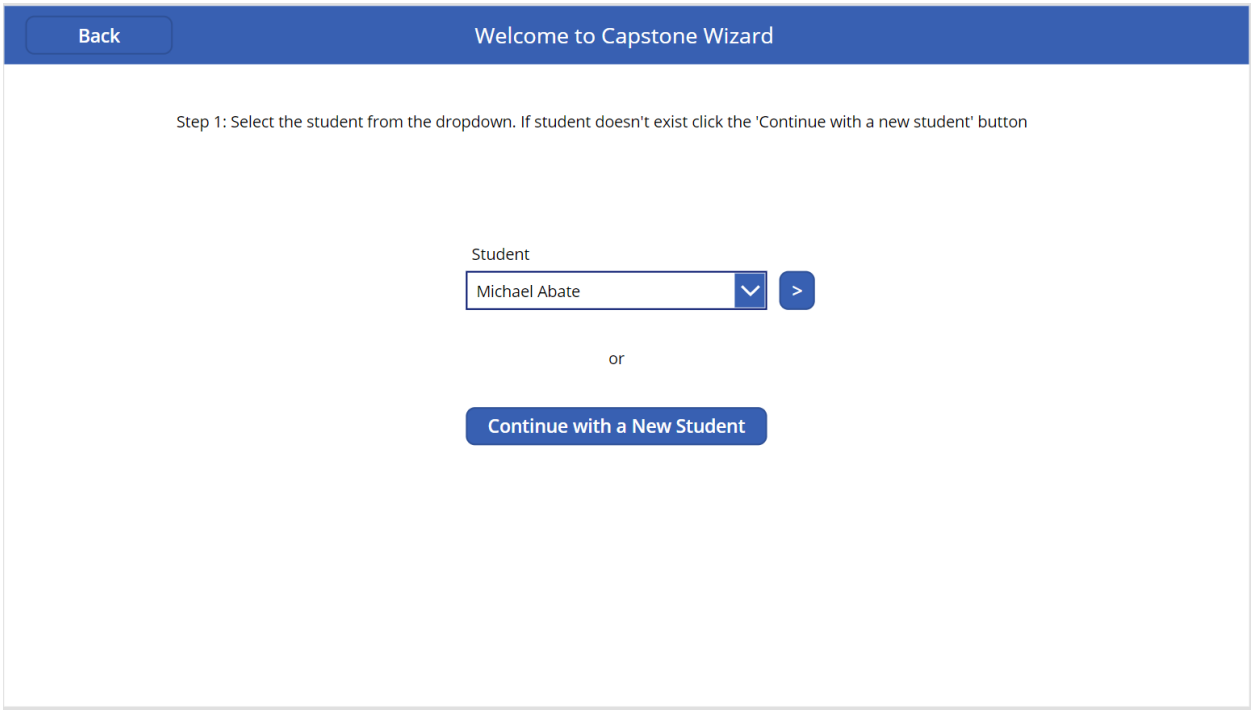
Screen Shot 6.3 – Wizard Tutorial 3



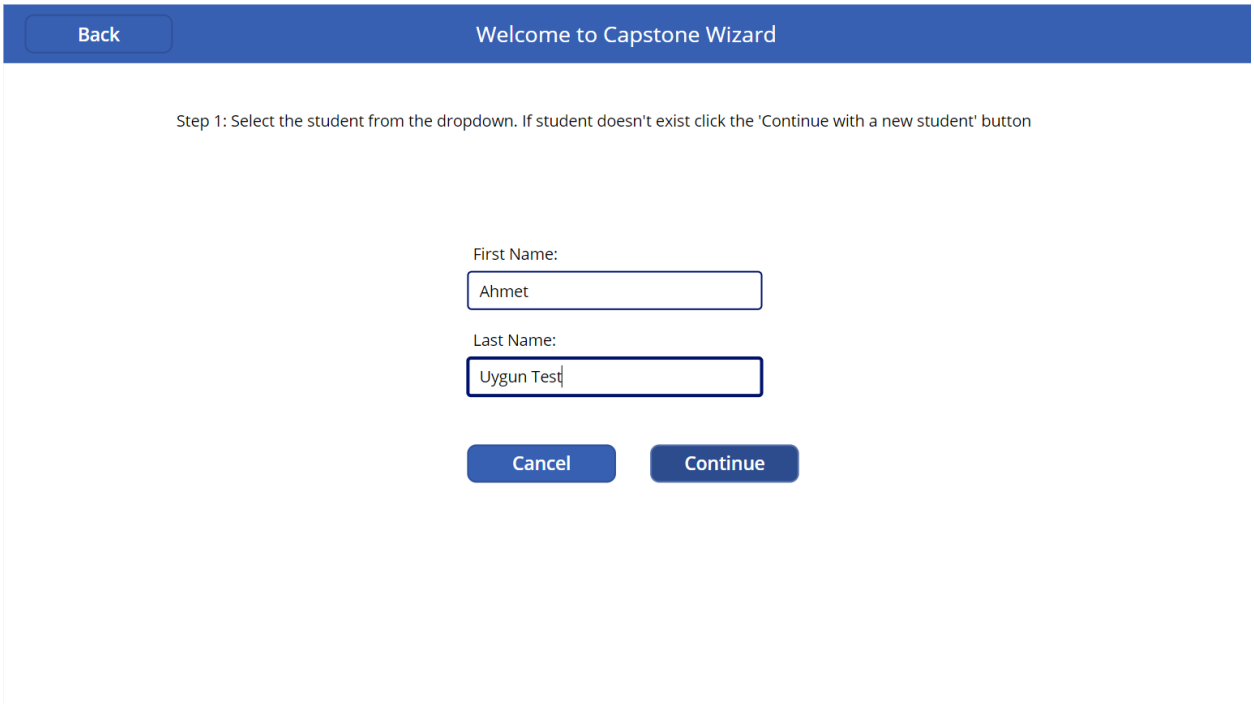
Screen Shot 6.4 – Wizard Tutorial 4



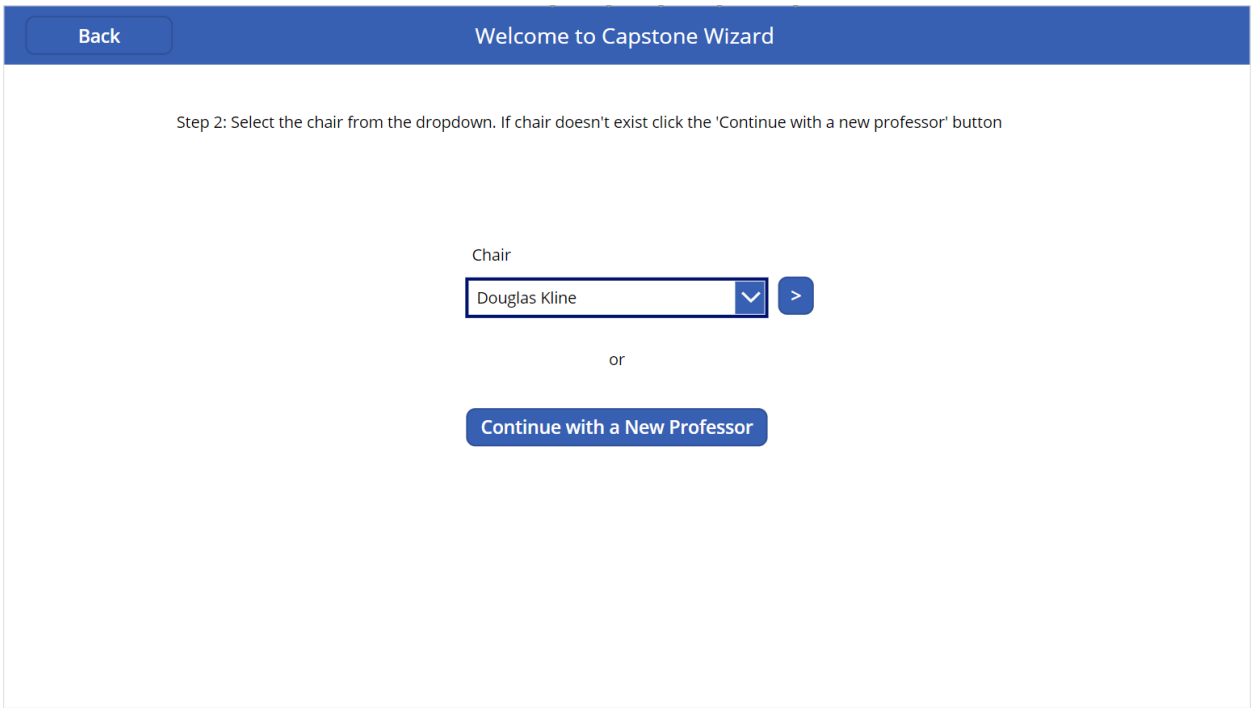
Screen Shot 6.5 – Wizard addStudent



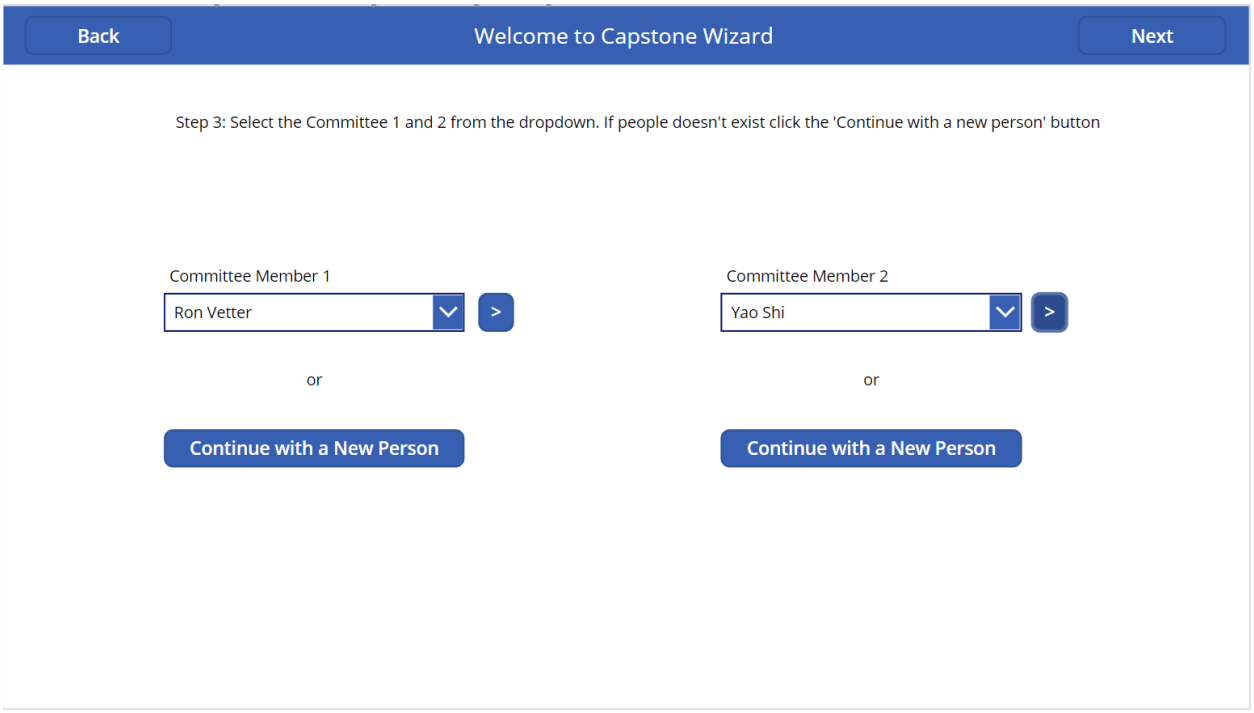
Screen Shot 6.6 – Wizard addStudent 2



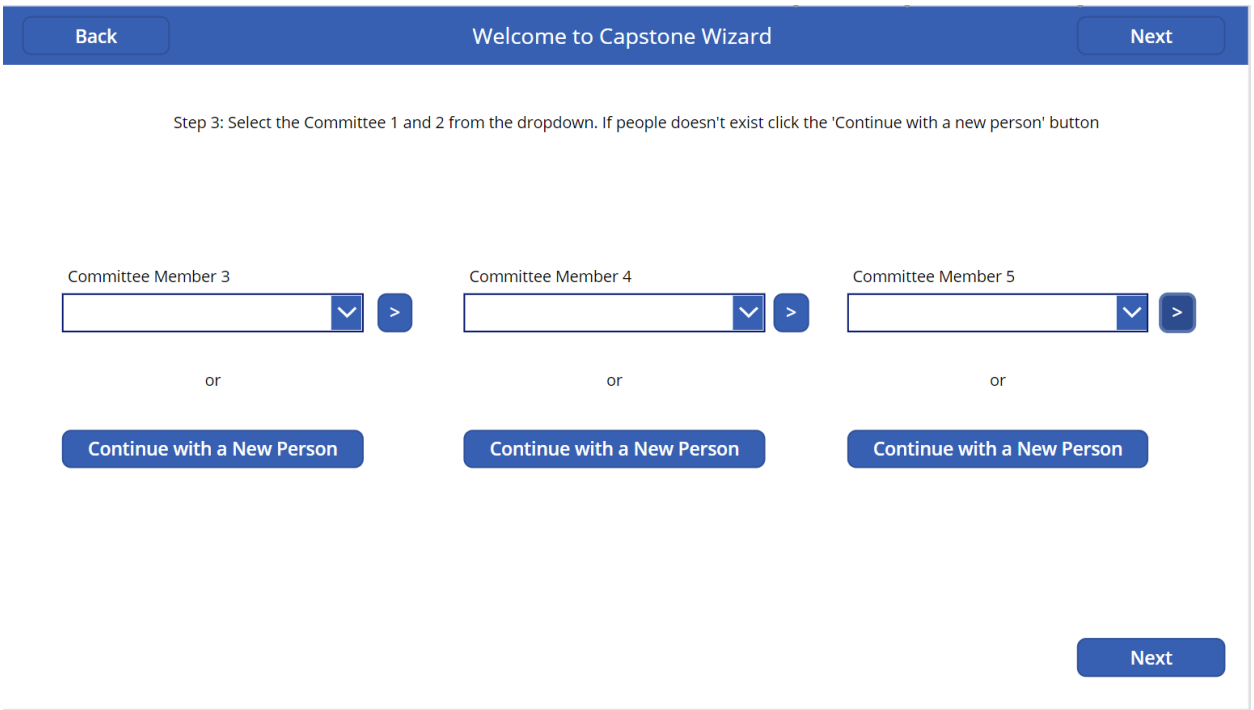
Screen Shot 6.7 – Wizard addChair



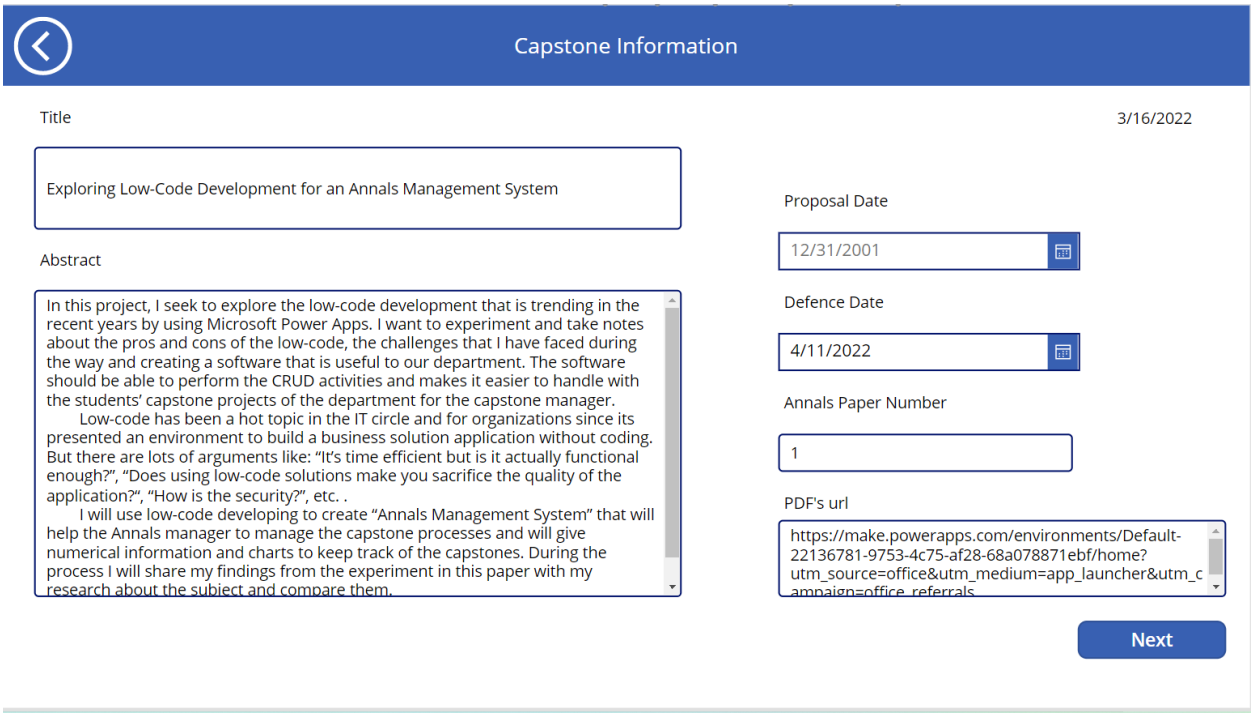
Screen Shot 6.8 – Wizard addCommittee



Screen Shot 6.9 – Wizard addCommittee_1

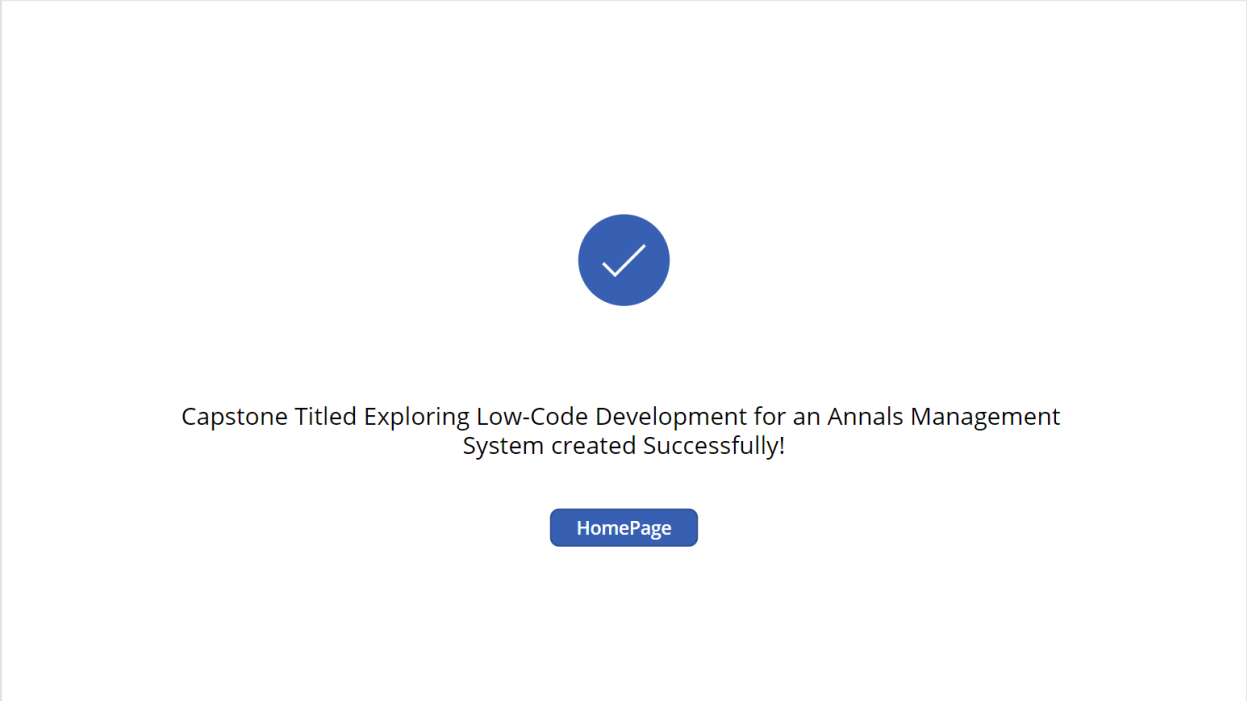


Screen Shot 6.10 – Wizard addCapstone

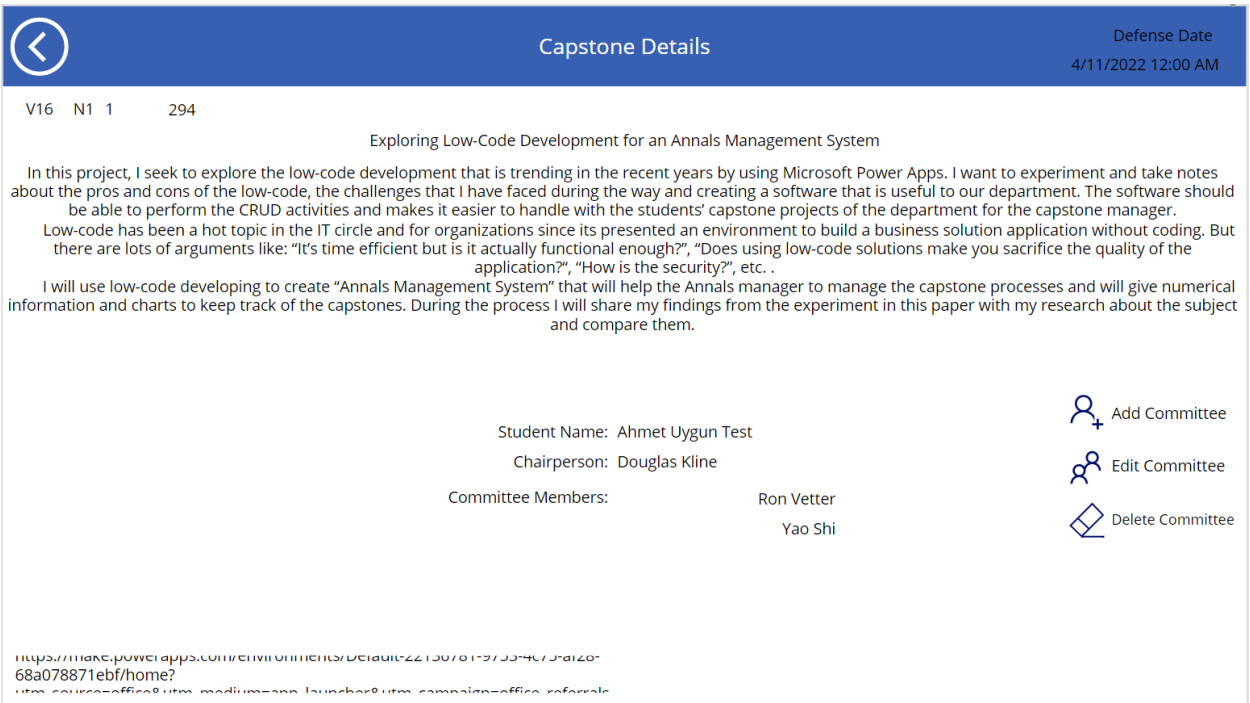


Screen Shot 6.11 – Wizard CapstoneOverview

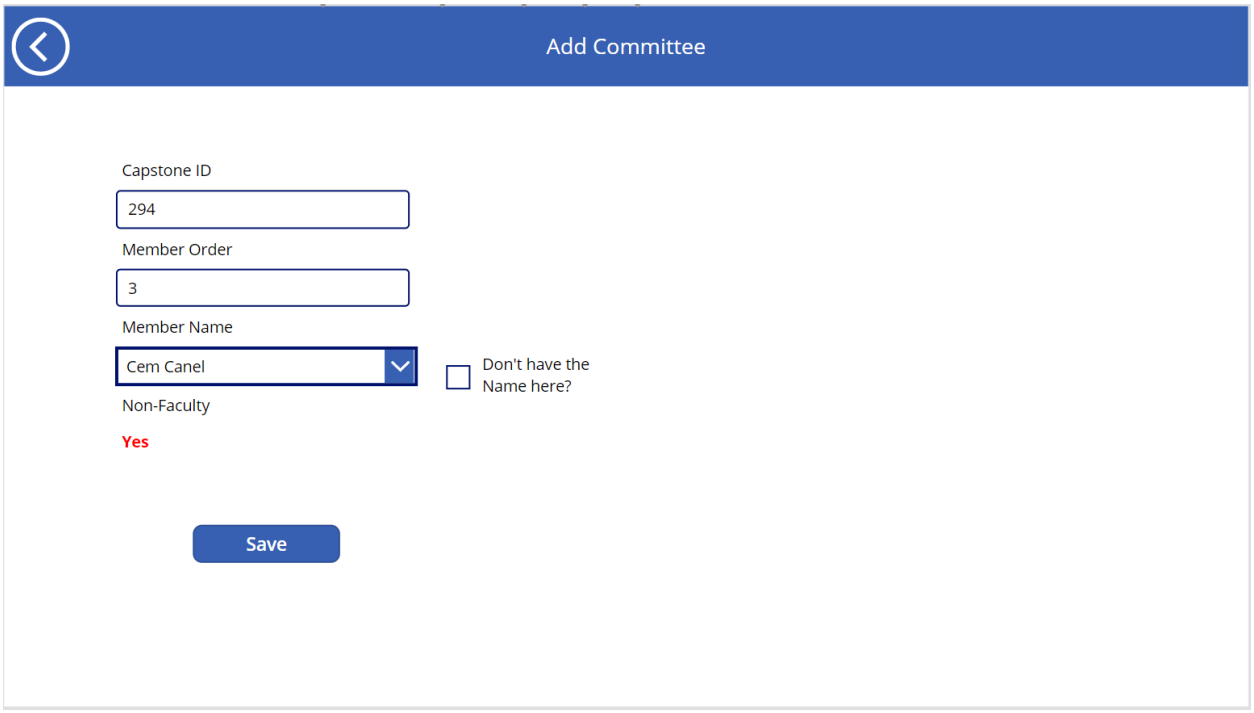
Screen Shot 6.12 - InsertCapstoneSuccess



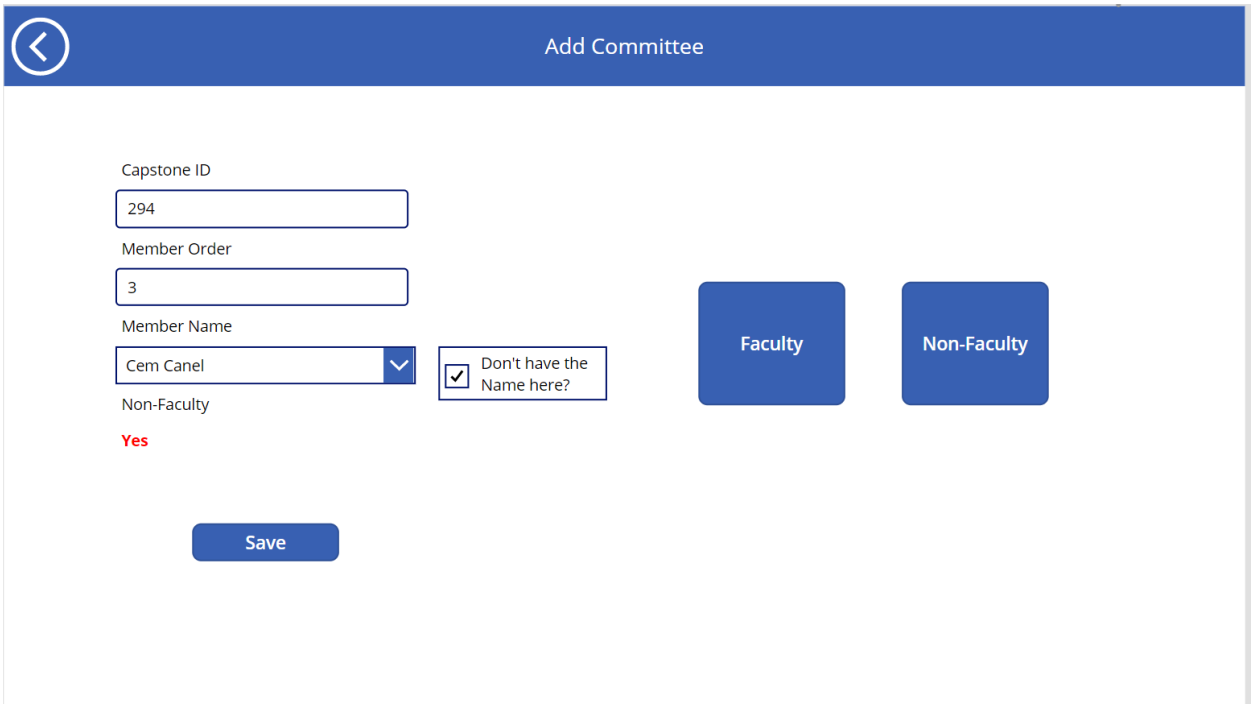
Screen Shot 7.1 – Capstone Details



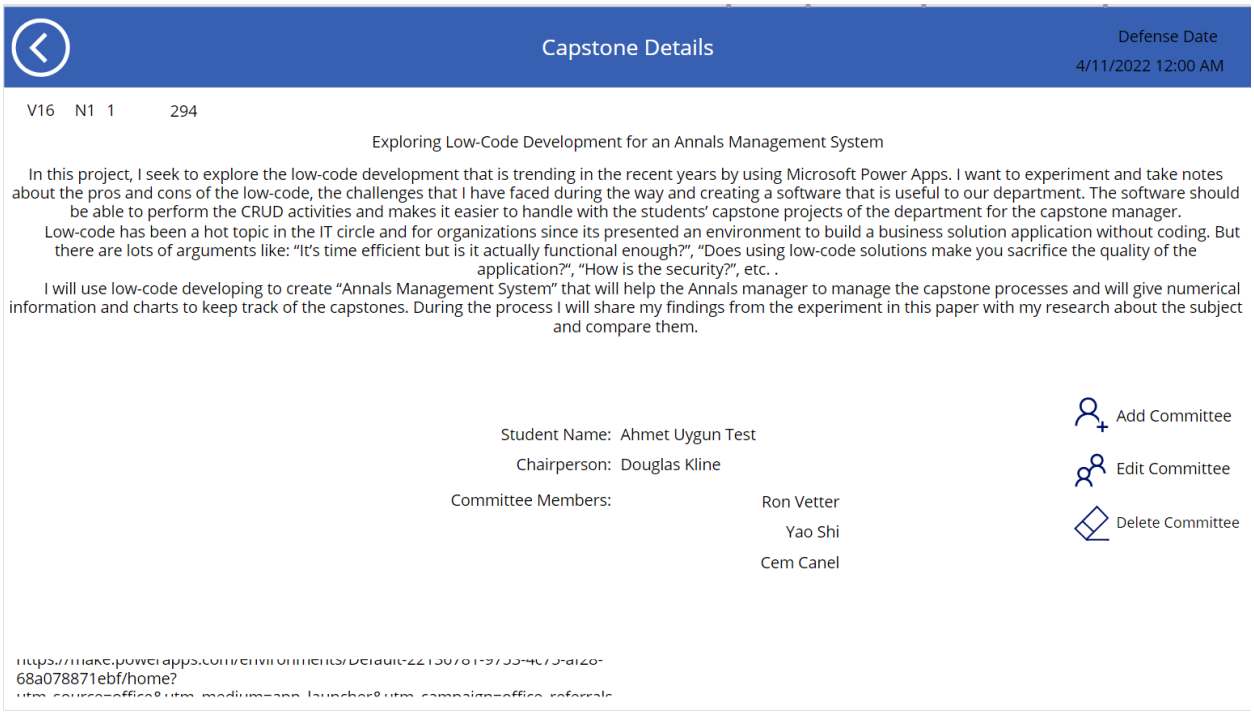
Screen Shot 7.2 - addCommittee



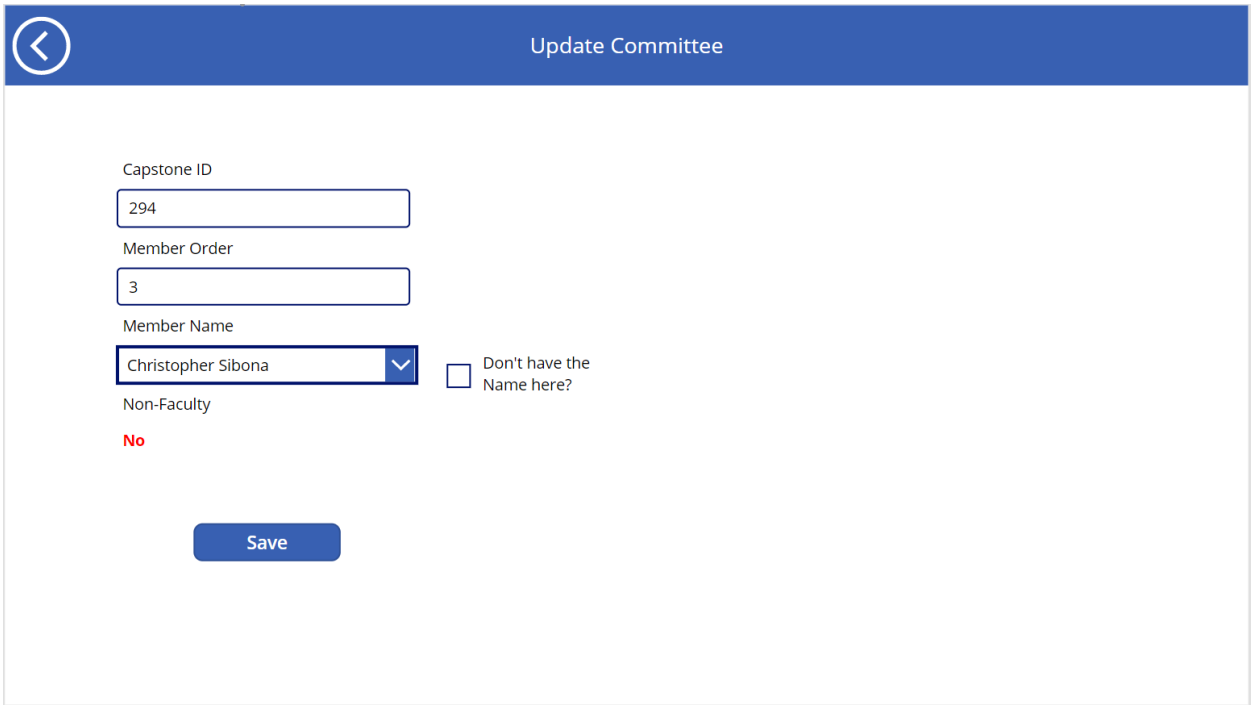
Screen Shot 7.3 – AddCommittee 2



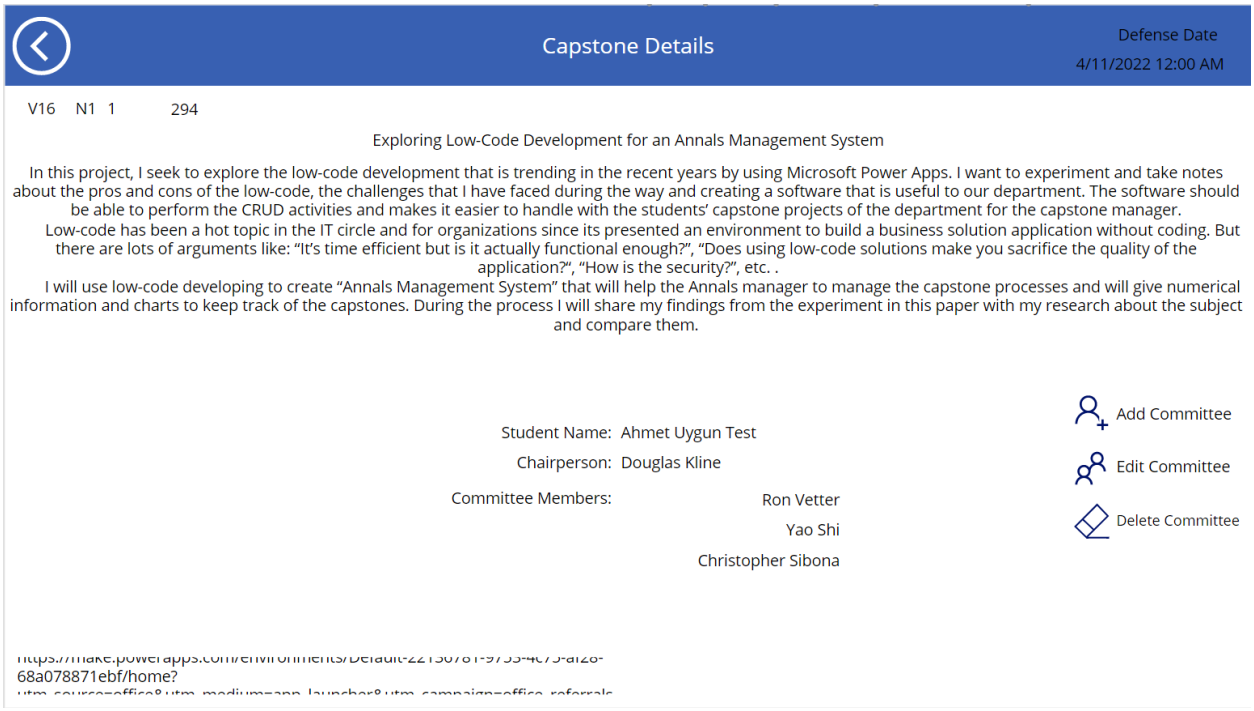
Screen Shot 7.4



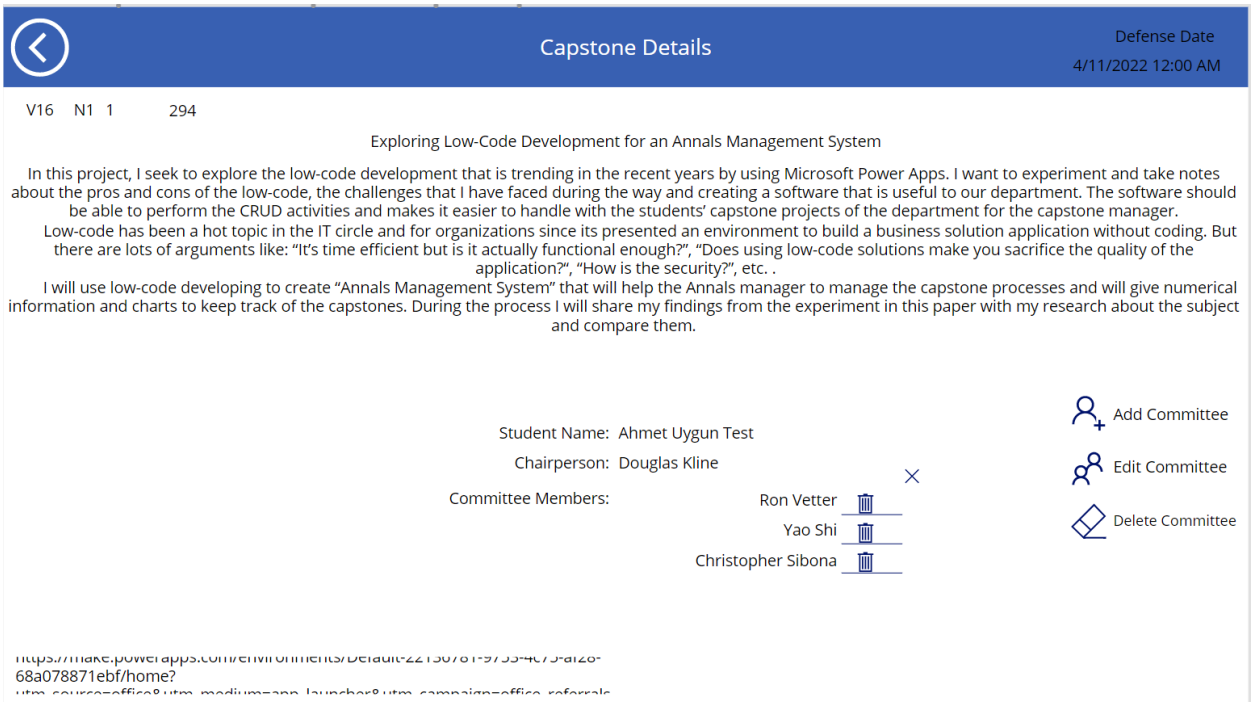
Screen Shot 7.5 – Update Committee



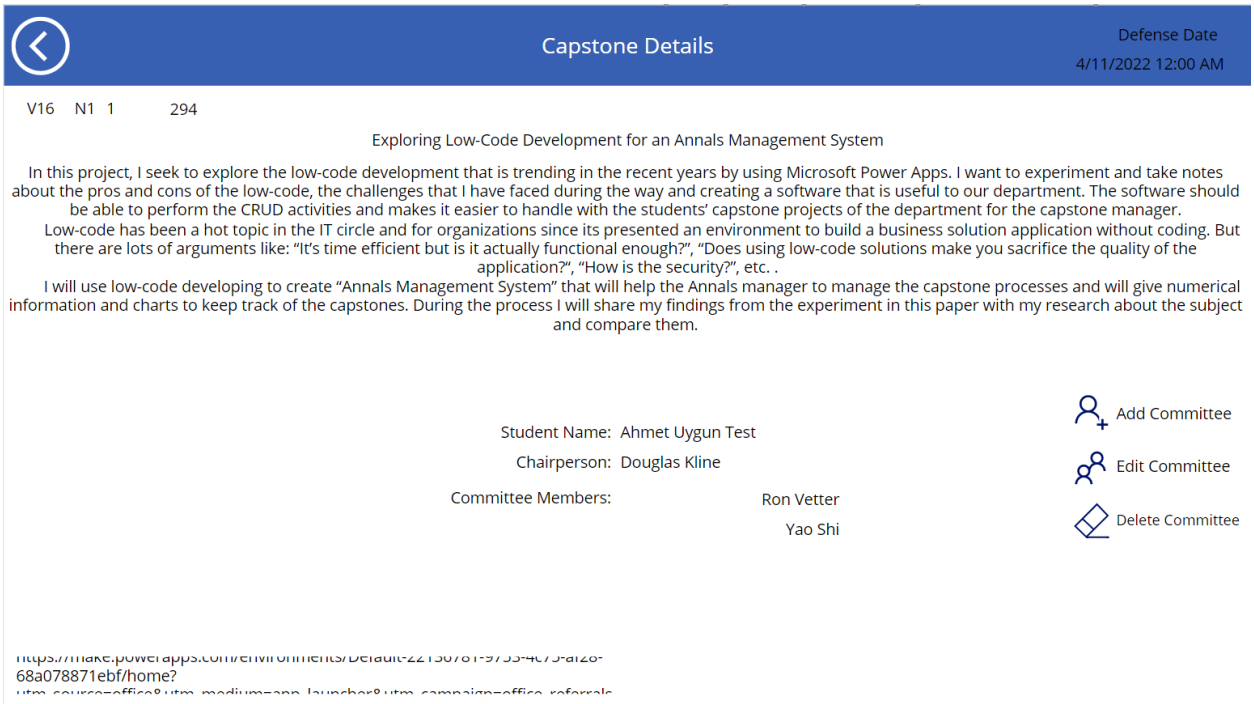
Screen Shot 7.6



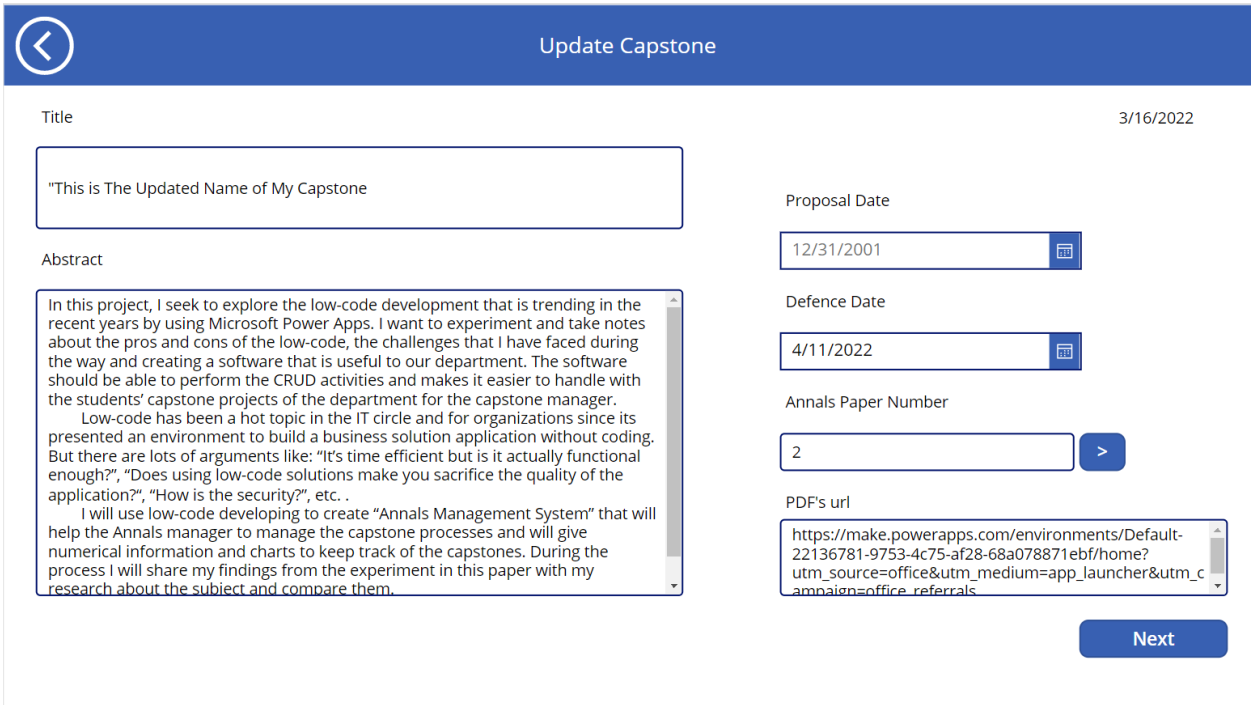
Screen Shot 7.7 – CapstoneDetails (Delete Committee)



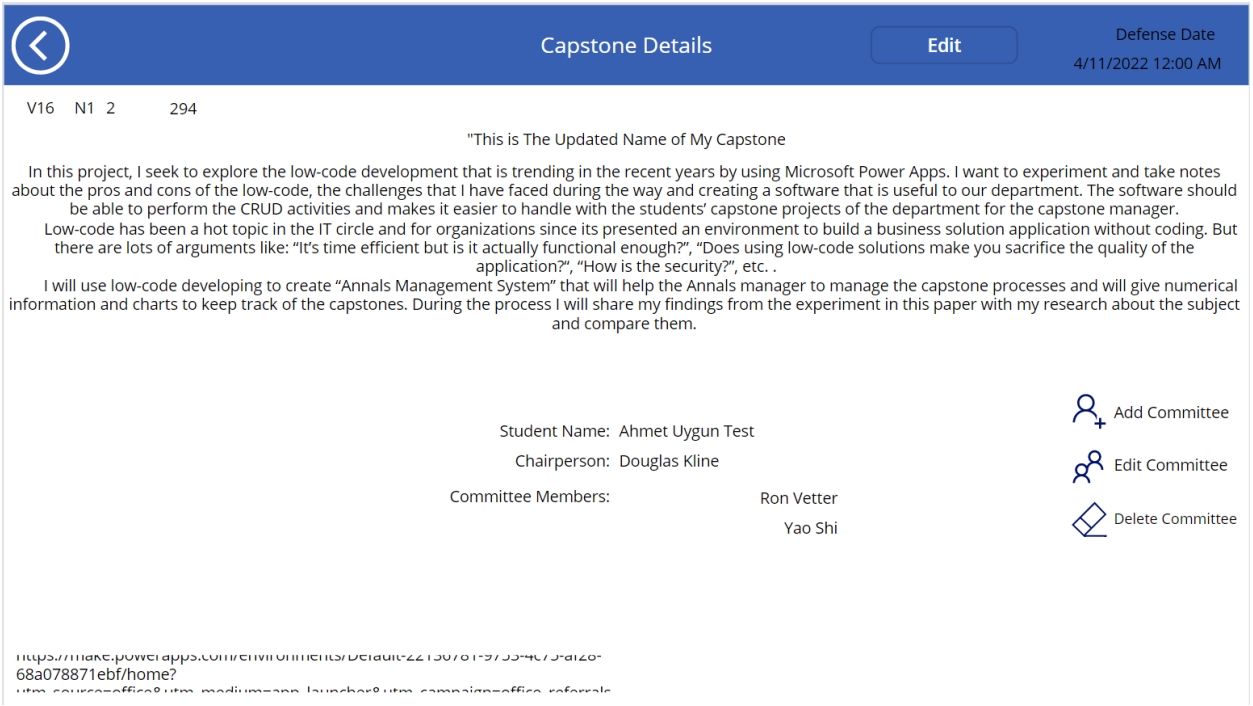
Screen Shot 7.8



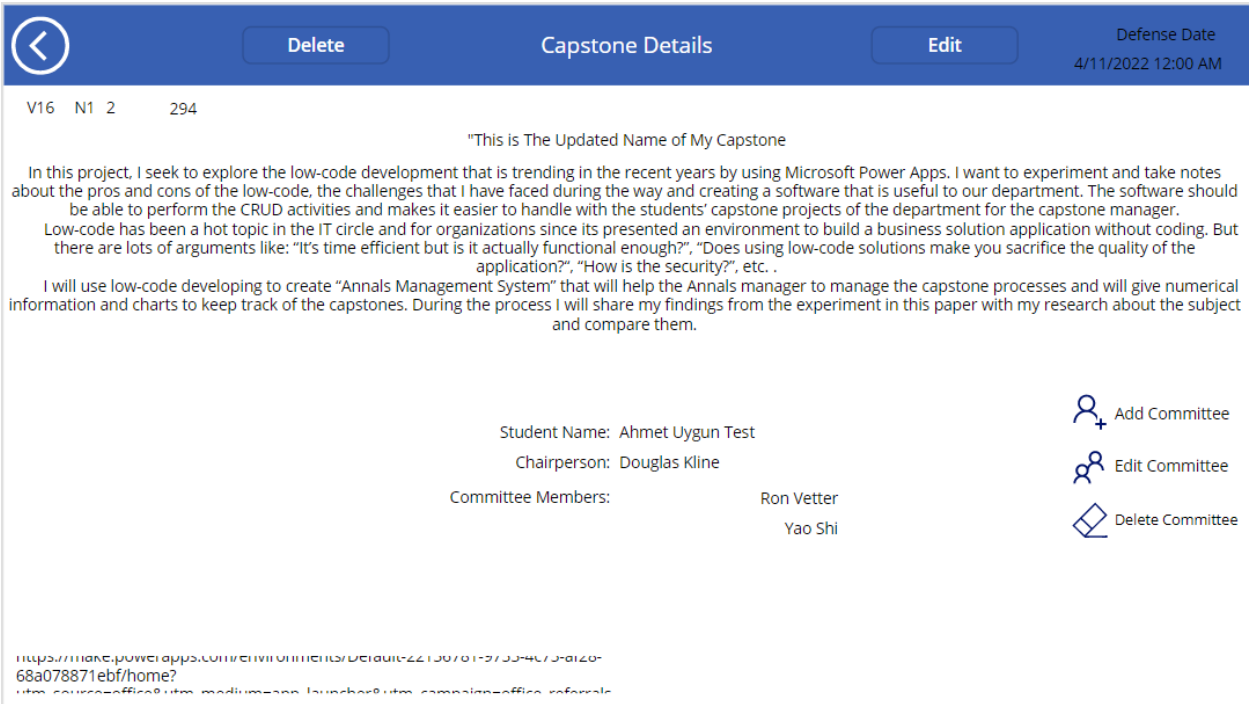
Screen Shot 7.9 – Update Capstone



Screen Shot 7.10



Screen Shot 7.11



The delete button is a dangerous one, so I will not keep that around since clicking it by mistake can cause the deletion of the Capstone immediately. So, I will keep the button hidden, and only the owner or Co-owner of the app can change the property to use it.

Screenshot 8.1 – AddStudent Flow