

Table of Contents

Abstract	4
Introduction	5
Project Summary:	5
Problem Statement.....	5
Solution:.....	5
.....	6
Project Objectives/Goals:	6
Project Scope:.....	6
Technologies Used:.....	7
Technical Architecture Diagram:.....	8
User Personas:.....	9
Use Cases:	11
Use Case Diagram:	12
Testing Plan:	15
Overview	15
Methodology.....	15
Scope.....	15
Test Logs and Procedures	16
Testing Review	18
Change Management Plan:	18
Budget:	18
Problems Encountered and Analysis of Problems Solved:.....	19
Conclusion.....	20
References.....	21

List of Illustrations

Figures:

Figure 1 Physical View Diagram8
Figure 2 Physical View Diagram9
Figure 3 Use Case 1.....10
Figure 4 Use Case 1.....10
Figure 5 Use Case 211
Figure 6 Use Case 3.....13
Figure 7 Use Case 3.....14
Figure 8 Budget Estimation Chart.....15

Tables:

Table 1 Timeline.....6-7
Table 2 User Persona Table.....10
Table 3 Use Case Table.....11 - 12
Table 4 Project Budget.....19

Abstract

Nexum is a web application that gives students a place to communicate and share resources online. According to several studies, a lack of resources and an inability to make friends in class are two of the most prevalent reasons students drop out. Melanie Hanson's study for the Education Data Initiative (2021) found that 13% of students drop out of college due to poor social fit and 3% due to mental/emotional issues.

The solution to these problems is to develop a web application that allows students to connect with one another on a website where they can write posts and conduct one-on-one chats with classmates and mentors. This online application would also provide students with a centralized location in which they could find and access all the resources they could need.

Nexum will require the development of a database to store all the postings that the students will make as well as the one-on-one chats. Both the data and the website will be hosted on Azure.

Finally, Nexum is a platform where students can have private chats and get the resources they need to succeed in their academic courses in a safe and secure environment.

Introduction

Project Summary:

Nexum is a web application that can be implemented into a college's programs to target first-year and transfer students with their needs and inquiries. Its main goal will be to assist first-year students and transfer students, who are students who transfer to a university after starting their studies at another institution, by giving community guidance and information to help them have a better first-year experience.

Problem Statement:

Studies that look at social interactions in higher education show that friendships and social connections are crucial factors in whether a student would drop out of college. A study conducted in the UK showed that of 12 students that dropped out, 9 of them marked "difficulties in making compatible friends (social support)" (Fyvie-Gauld, Winn, Wilcox p.711) as a significant reason for dropping out. Another study showed that, among other non-academic reasons, friendship was one of the three major causes of dropouts (González-Rodríguez, Vieira, Vidal, Table 2 p.222). A study done between some universities in Italy showed that "The career guidance students receive from universities is a factor that contributes to poor higher education decisions. Guidance is supported by policies nationally and locally, and universities increasingly invest in the guidance of prospective students. Nevertheless, the narratives of students who drop out suggest that the support these activities provide is quite limited." (Romito, Pillutti, Contini p. 462). A third study conducted across multiple schools in Europe also shows that "Sociodemographic dropout patterns align with differences in access to supportive resources in students' social networks outside school." (Elffers, Louse p. 14). Based on the evidence presented by these studies, it is clear there is a need to connect students with peers and mentors to improve student retention.

Solution:

Some solutions that are currently available are the University of Cincinnati subreddit on reddit.com, various Facebook groups, Twitter, and GroupMe. Another example would be the Learning Communities that UC implemented. Learning Communities are small groups of first-year students that all have the same schedule of classes and have at least one hour during the week to go in a class with only those students and help each other. The only problem that the online solutions have is that they are not limited to university students. Anyone can freely access these groups and add comments or posts that can be misleading. Nexum is a web application that will help provide students with information and assistance for what they will need short and long term, as well as having a platform in which they can chat with an upper classmate or professor about advice on making the best decision for themselves based on the major. Nexum is a web application that provides students with a social network around the university, class, and college. Nexum allows students to connect and interact with others more confidently through an online platform.

During our time at UC, we have experienced being first-year college students and seen firsthand the difficulties they go through. We noticed that being a first-year student has some of the most difficult challenges when finding friends, finding yourself, and finding the best resources that fit your needs. We noticed that your first year is your most crucial year. Many students tend to drop out this year due to hardships related to being a first-year student. Learning communities try to fix this problem without too much success.

Discussion

Project Objectives/Goals:

The goal of the project is to create a platform that allows students to communicate their experiences and assist one another. This will have a primary target of helping newer students have more social interaction and have a place to get help, but it is really for any student who needs it. Unlike other similar solutions, Nexum does not require you to meet in person and by using a web application that can quickly contact another person. Using the experience of students, a database of information from the classes can be built up to let students know what is expected for their class. With it, a student can better prepare for the classes and know what they will need, so they can get the proper tools to be it the right laptop or an excellent book. Some of the goals Nexum can do include:

- A system that can help first-year students know the tool they will need for their major
- A system that can show students all the university-provided tools that can be used as an alternative should they be limited to what is available.
- A networking section that will allow first-year students to connect and network with other students in the same college as well as professors and faculty.

Project Scope:

The scope of Nexum will pertain to the University of Cincinnati. Specifically, the School of IT toward first-year students and transfer students. However, this can be used by any group of students. The web application will allow scalability to other colleges and universities in the future.

- A web application, app, allows students to communicate and provide information about their experiences.
- Database of what students used or need for different classes to be better prepared.
- Profile page so students can find other students in their major.
- Platform for students to communicate with one another.
- Web Scraper

Quick Project Timeline:

Task Name	Duration	Start	End Date
Deciding Project	14 days	9/27/21	10/8/21
Research	5 days	10/8/21	10/13/21
Database design	12 days	10/13/21	10/25/21
UI Design	18 days	10/25/21	11/12/21
Database Interaction	7 days	11/12/21	11/19/21

User functions	12 days	11/19/21	12/1/21
Integrating & finalizing for presentation	14 days	12/1/21	12/15/21
Chat function	1 month	1/1/22	2/29/22
Authentication	3 months	1/1/22	3/25/22
Clean Up webpages	Everyday till done	1/1/22	3/28/22
Properly store comments	1 week	1/17/22	1/24/22
Profile Page	14 days	2/14/22	3/1/22
Create all webpages	3 weeks	2/14/22	3/7/22
Work on possible Web Scraper		3/1/22	3/23/22

Table 1. Timeline

In the first two weeks, time will be spent on defining and figuring out the basic needs of the project, basically having a good idea of the kind of software the work is going to be based on. The research process will constitute the proof that the problem exists and try to find all the solutions that might have been built to try and solve the issue if any. After the research, it is time to build a database that will hold most of the data that will be stored and collected during the creation of the web application, while working on that the UI also is being built. The database interaction is where everything will be checked to make sure that the UI and data as well as the function created can respond to the command of one another. After all of that, the focus will be on integrating everything to make sure that everything is ready for the presentation.

A recent addition to the Nexum project is a web scraper using Scrapy. This tool was used to find laptops being sold on amazon to then compare them to find the best fit for the individual student. Based on major, needs, and activities.

Technologies Used:

The technology used for the Nexum application is common web application technologies such as Visual Studio, Azure SQL Database, JavaScript, HTML & CSS, and Azure Cloud.

- Visual Studio: This was used to build the front end and back end of Nexum.
- Azure SQL database: Database to store data that is gathered from users.
- Python: A easy to learn software for everyone in the team to use as well as realizable so everyone can code if needed.
- JavaScript: This is going to be used for coding on the front end of the application.
- HTML & CSS: This is going to be used for the front-end side of the project, i.e., the website.
- Azure Cloud: The primary platform to host Nexum.

Technical Architecture Diagram:

In Figure 1, the physical view diagram shows Nexum's network topology and the integrations between the host and the system. Azure Cloud Services is the primary point of access that all Nexum's systems go through due to being the primary host of Nexum's website application and database. Also, Nexum is connected to Microsoft login to help create a user identity based on an already existing account. This is important because the system limits who can access the application to people who have a UC email. From there, the services connect to the internet and allow users to access it. Both the Database and Website server host interact with each other to store and display data on each other.

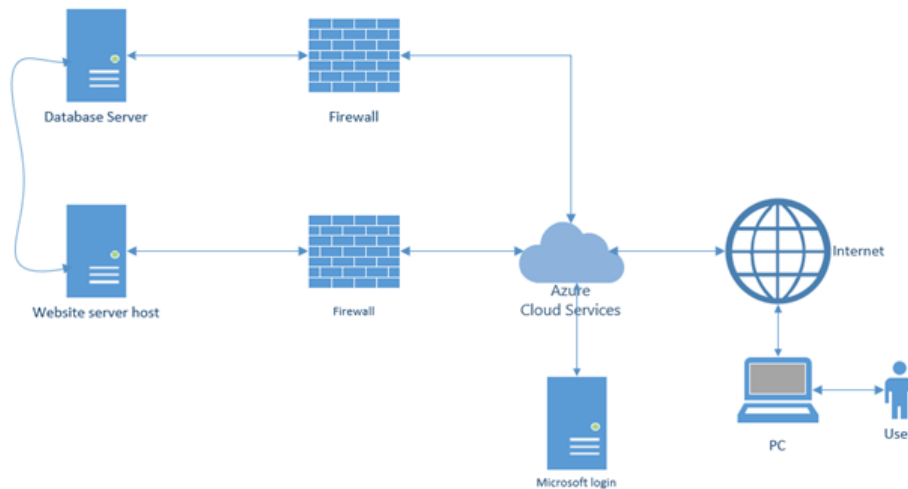


Figure 1. Physical View Diagram

The Scenario view (See figure 2) shows a user's possible actions and how the system interacts with the users. First off, users can send or receive a message, post a comment, or modify their profile. This is all done through the website server inside Azure. When changing anything, it updates the database server, recording and storing data on Azure cloud services. All user inputs must go through a firewall to verify their account is allowed to access the server using the Microsoft login services, which updates the firewall to allow the user in. When the user interacts with the website, the site pulls and gets information from the database server to manage the input of the users. Finally, the user can exit the program disconnecting them from the connection.

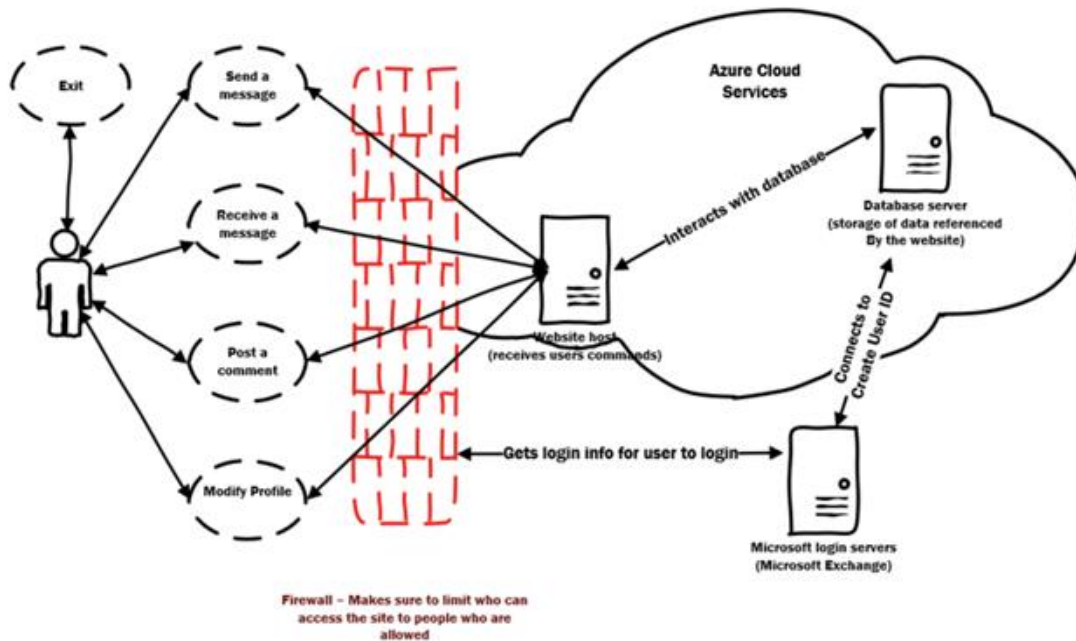



Figure 2. Scenario View Diagram


User Personas:

For user personas, reliable characters were created to represent the key audience segments that could be used for the project. In the case of the project, three personas were created Dakota, Emily, and John. Dakota is a student who would benefit from the web application because he wants to be aware of his choices about the classes he is going to take. Emily, who is a first-generation student, wants to have a place where she can ask questions to students who already have been in her position and can give her advice. And John, who is returning to college, will find the web application extremely helpful and informational. All the personas that are created can benefit from the web application whether they are first-generation, returning students, or right out of high school students.


Table 2 User Persona Tables

User Persona: 1	
	Title: Student
	Name: Dakota
	Age: 18

 <p><i>Figure 3: Picture of a Man</i></p>	<p>Gender: Male</p>
<p>Behavior</p>	
<p>Pain</p>	<p>Does not understand technology</p>
<p>Needs & Goals</p>	<p>To feel confident in his choices as a student</p>

<p>User Persona: 2</p>	
 <p><i>Figure 4: Picture of a woman</i></p>	<p>Title: Student</p> <p>Name: Emily</p> <p>Age: 18</p> <p>Gender: Female</p>
<p>Behavior</p>	
<p>Pain</p>	<p>First-gen college student and feels overwhelmed</p>
<p>Needs & Goals</p>	<p>To feel more comfortable in classes and make connections.</p>

<p>User Persona: 3</p>	
	<p>Title: Student</p>
	<p>Name: John</p>
	<p>Age: 56</p>

 <p><i>Figure 5: Picture of an older man</i></p>	Gender: Male
Behavior	
Pain	Returning to college after several years and feeling lost with online classes and technology.
Needs & Goals	A quick way to connect with classmates to help him navigate around classes and assignments.

Use Cases:

Using the three personas three different use cases will be made to show how Nexum would work in different situations and the needs of people.

Use Case ID	001
Use Case Name	Finding Students related to themselves
End Objective	To find a student related to their attributes, this could be anything from the same class, major and track, interest, etc.
User/Actor	Student
Trigger	The user is looking for someone in the same class as him/her or level.
Frequency of Use	Around 25% of users use it to make connections with their classmates besides getting the technologies they need throughout their college career.
Preconditions	The user is logged in and on the chat section
Basic Flow	

Alternate Flow	
Postconditions	

Use Case ID	002
Use Case Name	Needing guidance about college
End Objective	To find a mentor that will give the student an example of how to succeed in college and life.
User/Actor	Student
Trigger	The user is looking for an upper classmate or a professor whom they can look up to as an example to be a better student.
Frequency of Use	Around 30% of the students use this to find a mentor
Preconditions	The user is logged in and on the chat section
Postconditions	Users get to have a conversation with an upper classmate and get to be their mentee.

Use Case ID	003
Use Case Name	Getting help to navigate online classes from the students and professors around campus
End Objective	Users can seek help from other students on the web application about how the online portion of the class works.
User/Actor	Student
Trigger	Users are perplexed about the online portion of classes and seem to be falling behind and need some help on how to manage.
Frequency of Use	About 10% of the students use this to get help in classes
Preconditions	The user is logged in and on the chat section
Postconditions	The user gets the help needed in the online classes and makes connections along the way.

Table 3 Use Case Tables

Use Case Diagram:

Based on the three use cases designed for the personas, below are the three use case diagrams on how users can use the web application Nexum.

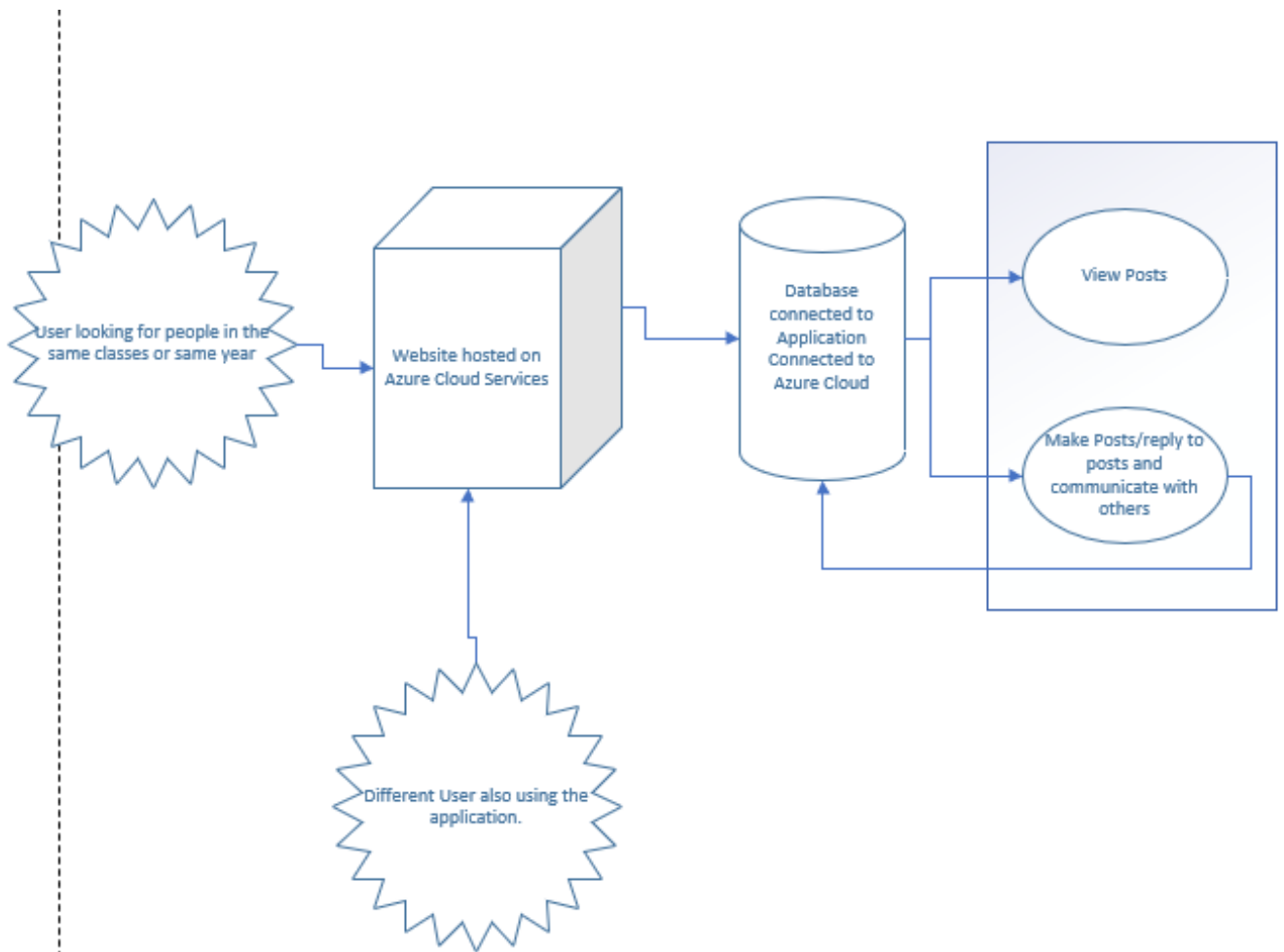


Figure 6: Use Case 1

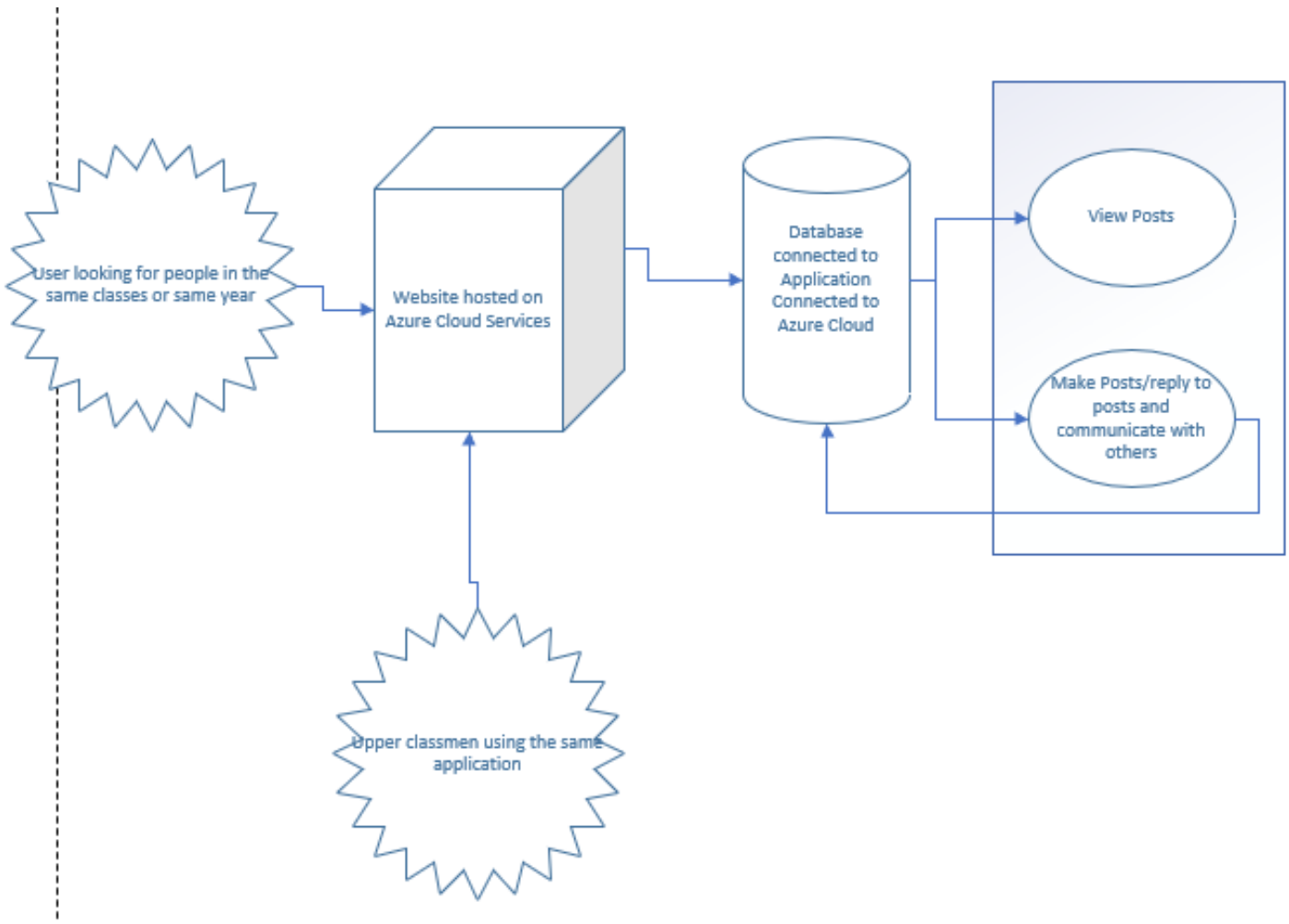


Figure 7: Use Case 2

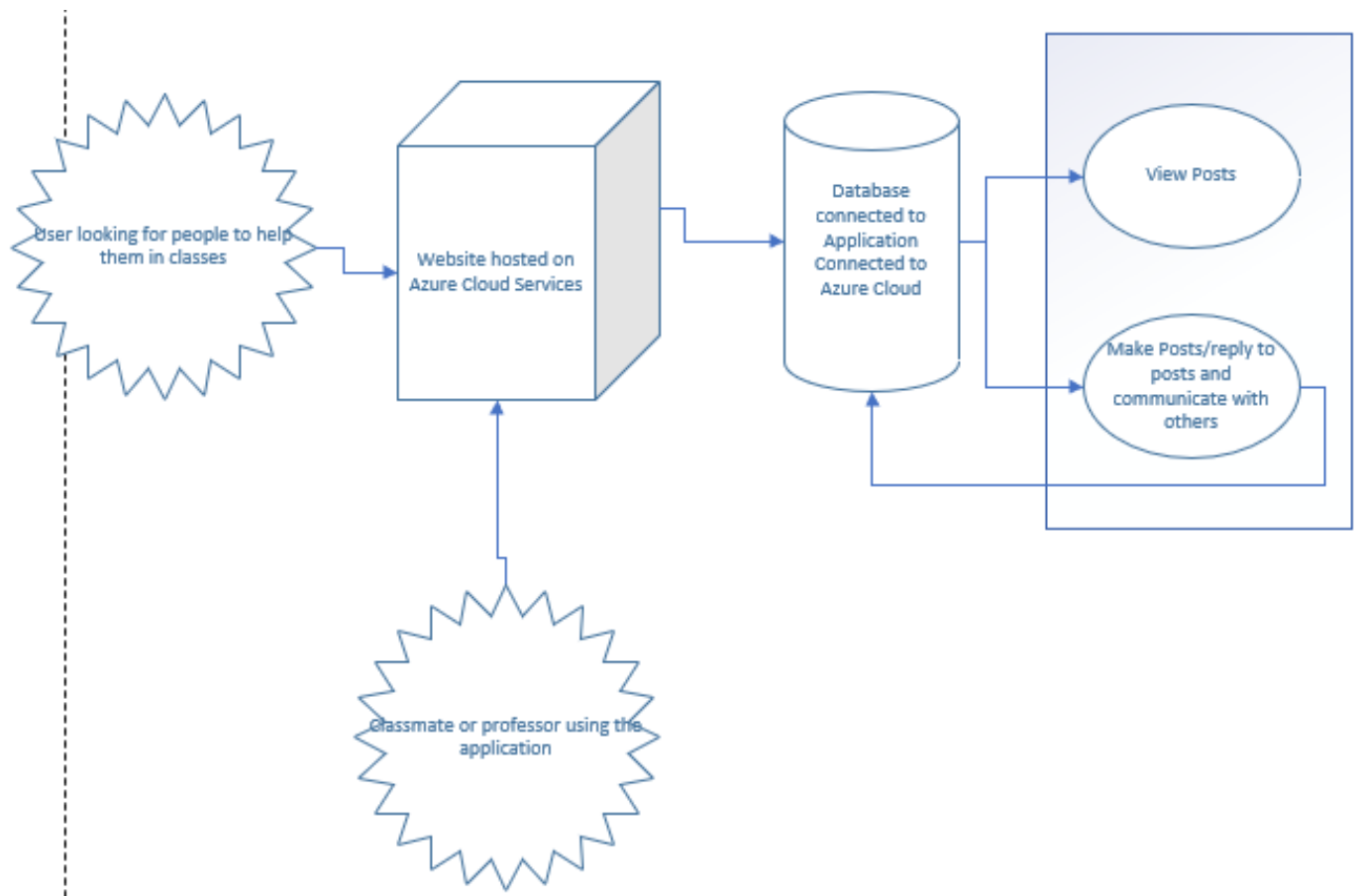


Figure 8: Use Case 3

Testing Plan:

Overview

Throughout the project, different people work on different parts of the project. Step-by-step testing is done to then combine the parts people are working on into a whole project using GitHub and publish to Azure to make sure the code is working together.

Methodology

Throughout the current cycle, one thing at a time was added, to know if anything should cause a conflict or will not work with what is currently running. This is important so that what is working will not break and an incompatible software is not implemented affecting other systems negatively. All this is tested on a mock local system to reduce the cost of running azure. Finally, is sticking to things that are familiar and only branching as needed to find alternatives.

Scope

The scope of the project had to be limited to the time available in the fall and spring semesters. The scoped project includes features to test as follow:

- A. The ability of Students to Message each other
- B. A page to search for classes and view people's comments

- C. A Featured page to highlight classes that are well-liked by students
- D. Limit access to people who are and where UC students are.
- E. Get the project working locally before publishing work on Azure.
- F. Have the project upon Azure by the first week of March.
- G. Have 3-4 Weeks before IT Expo to polish, add features, and finalize changes. This also includes having users test the project and get feedback on what could be improved.

Test Logs and Procedures

- Test #1 – For this test, the objective will be to make sure that Microsoft Azure can host the needed applications and perform functions to make Nexum work.
 - User: Team
 - Role: Get the azure resources functional
 - Host Website – Pass
 - Host SQL Server – In progress
 - Connect to GitHub - Pass
 - Make Sure everyone has access - Pass
 - Date: 1/31/22
- Test #2 – Make sure that GitHub is working well for everyone, and everyone knows how to use GitHub
 - Users: Team
 - Roles: Get everyone on to the same group Merge finished branches and features to master branch
 - Pass/Fail: Pass, everyone was able to access the code on GitHub and was successfully able to pull the code.
 - Date: 2/4/22
- Test #3 – Get a local version of Nexum working before publishing it on Azure to make sure that it's not an Azure issue, but a code issue.
 - Users: Team
 - Roles: Before pushing to Azure make sure it works locally so all members know that it's not an issue with the code
 - Also Having Azure always running drains available credits so testing off Azure is important.
 - Pass/Fail: Pass, the code is functional locally all that is needed is to get the database connections fixed and then the final push to the Azure Database
 - Date: 2/6/22
- Test #4 – Once local code is working start to publish parts of the project into Azure.
 - User: Team
 - Role: This step is once everything is proven to work offline
 - Once up, care has to be done to not permanently change the code till it is known that it will not break anything.
 - Pass/Fail: Partially pass, since this test is still in progress it is a pass as of now
 - Date: 2/6/22
- Test #5 – Once working on Azure get people's feedback on the web application and make edits.

- Users: Friends, first-generation students, students with majors different from IT
- Role: Student accessing the application, navigating the website
- This test is used to check if other users can access the web application via the website and create posts for individual classes
- Pass/Fail: The test passed. The users were successful in creating and uploading posts to different classes.
- Date: 2/4/22
- Test #6 – Test if users can see individual posts that are made in classes.
 - Users: Friends, Students
 - Roles: Students joining the website and seeing multiple classes and individual posts those other users posted.
 - Pass/Fail: The test passes. The users were successfully able to see individual posts that were made by other users.
 - Date: 1/20/22
- Test #7 – Test is users can make a post and have them show up on individual classes
 - Users: Friends, Students
 - Roles: Students that are joining the website will make a new post on any class that they want, and they should be able to see the post that they make show up in the class.
 - Pass/Fail: This test passes. The users were able to create posts as intended.
 - Date: 1/20/22
- Test #8 – Test if users were able to join the Chat and communicate with other users at the same time.
 - Users: Friends, Students
 - Roles: Students should be able to join the open chat room and actively chat with other users.
 - Pass/Fail: This test passes. Users were able to join the chat room and communicate with other users
 - Date: 2/25/22
- Test #9 – Save chat history within the database.
 - Users: Friends, Team
 - Roles: When users enter the chat, they should be able to see the previous chat history.
 - Pass/Fail: This test failed. Users were not able to see previous chat history within the chat. This will be implemented in a future update.
 - Date: 2/25/22
- Test #10 – Web Scraping
 - Users: Team members
 - Roles: Users access the scraper to get external resources

- Pass/Fail: This test passed. The user was able to successfully scrape the web for laptops that could be used by a student. However, this was not fully implemented and will need more time to be fully functional.
- Date: 3/17/22

Testing Review

Over the last two semesters, multiple tests have been done. An example is instead of one-to-one chat changed to a group chat that all users talked in. Looking back at Nexum the scope was much too large with so many parts needing to be done in such a short time with only five people. Another big point is to look for already made solutions to implement them into Nexum instead of trying to make them from scratch. If we were to do this project again in the future the changes that we would make are to change the coding language to one that is easier to code in and more tailored to typescript or JavaScript and use a simpler backend technology such as node.js or SQLite. Additionally, the scope should be more focused and polished.

Change Management Plan:

- C# over Python and Java.
 - This was due to more familiar with implementing security to C# and some team members recently worked in C#.
- HTML has been replaced with CSHTML, which is inline C# code with razor syntax allowing the pages to be interactive.
- For a change to go into effect, 3-4 members need to agree. Also, it must be for a few reasons.
 - There is a better solution.
 - Compatibility issues with already existing code
 - Lack of familiarity
- Changed the project language from python and java to C#, due to multiple team members recently working with C#. With the changes to C#, HTML changed to CSHTML, which is inline C# code with razor syntax, so the web pages are interactive and functional.
- Unable to do a one-to-one chat with users, but instead got a group chats that multiple users can join which was made using Blazer.
- Remade the database design schema to better store the project's data.

Budget:

Personal machines cost roughly \$10,000 to begin with. The internet cost \$2,000 over the course of 8 months, while Azure cost roughly \$200 over the course of the project. Azure at scale would

cost \$3,461.50 per month and \$41,537.98 per year for a college.

Microsoft Azure Estimate					
Your Estimate					
Service type	Custom name	Region	Description	Estimated monthly cost	Estimated upfront cost
App Service		East US	Basic Tier; 1 B1 (1 Core(s), 1.75 GB RAM, 10 GB Storage) x 730 Hours; Windows OS	\$54.75	\$0.00
Azure SQL Database		East US	Single Database, vCore, RA-GRS Backup Storage, General Purpose, Provisioned, Standard-series (Gen 5), Local Redundancy, 1 - 8 vCore instance(s) x 730 Hours, 100 GB Storage, 50 GB Backup Storage	\$1,494.25	\$0.00
Azure Firewall		East US	Standard tier, 1 Logical firewall units x 730 Hours, 0 GB Data processed	\$912.50	\$0.00
Support			Support	\$1,000.00	\$0.00
			Licensing Program	Microsoft Customer Agreement (MCA)	
			Billing Account		
			Billing Profile		
			Total Monthly Cost	\$3,461.50	\$0.00
			Total Yearly Cost	\$41,537.98	
Disclaimer					
All prices shown are in United States – Dollar (\$) USD. This is a summary estimate, not a quote. For up to date pricing information please visit this link . This estimate was created at 3/18/2022 6:44:38 PM UTC.					

Table 4: Budget Estimation Chart

Problems Encountered and Analysis of Problems Solved:

While working on the project a few issues occurred. The first issue faced was on the database side; the design and function were difficult for the database. After visually creating the diagram, it became easier to create and read the flow of data. The next issue was with authentication and what it would reference but by using Azure it became possible to reference an AD type system that can check for UC credentials. However, even though it filters users when they access the site, it has created an issue that glitches parts of the website and prevents people from posting messages. One of the biggest issues is ASP.NET and uploading files on Azure. The code works well before uploading to Azure, but once upload, it seems to break. This upload is done by overwriting the old code, so creating a new resource group and using this as the base code will hopefully resolve the issue. Nearing the end of the project, one of the largest issues to resolve is connecting the front end to the backend of the system and making sure all the parts work well together.

In the second half of the capstone more issues occurred the most problems related to Azure hosting. Every student account being used for the project got disabled leading to the loss of the already hosted web pages, but with proper documentation, recovering the lost data was easy in within 24 hours Nexum was working again. In the end, the web application was hosted locally due to the risk of things running into issues so late in the project as well as not having enough Azure credits to host all of it on Azure. Another issue that arouses was one-to-one chat functionality and the diffusely to implement it. With time running out and many other pressing goals to work on a compromise had to be made to have any form of chat. The chat goal of one-

to-one went to a simpler group chat with no authentication, just an application that allows users to chat with others in a large chat group. The chat function was successful and then hosted on Azure allowing people on different devices to access the chat from different devices. Issues occurred as well with storing class comments into a database and auto assigning them an ID to properly separate comments to proper websites. This issue took assistance from Professor Michael and Professor Dekok to resolve. At the end of the project attempts to include a web scraper were also tried, but again due to time constraints, only the barebones scraper would gather data and put it into an excel sheet.

Conclusion

Throughout the two semesters of having worked on Nexum, our team has learned and developed several new skills. First, the team learned time management; with fewer hard deadlines, it was important to stay on track to not fall behind. The team was able to grasp better the time it would take to do a project and more meetings a week to finish things that needed to be done. Secondly, the team got to know each other better and what each other's strengths and weaknesses to work more effectively. Lastly, the team got more familiar with Azure and what it can do for the project. This is still the most important skill learned since the whole project is hosted by Azure which allowed the creation of this project. Currently, a good framework was built and multiple parts working locally. In the second semester, many issues had to be resolved quickly leading to more work being done quicker and more effectively. When working as a team if you're stuck or need help it is important to rely on the team to get work done efficiently.

References

- Elffers, Louise. "One Foot out the School Door? Interpreting the Risk for Dropout upon the Transition to Post-Secondary Vocational Education." *British Journal of Sociology of Education*, vol. 33, no. 1, Taylor & Francis, Ltd., 2012, pp. 41–61, <https://www.jstor.org/stable/41343722>
- Hanson, Melanie. "College Dropout Rates" EducationData.org, November 22, 2021, <https://educationdata.org/college-dropoutrates>
- Romito, Marco et al. "Why Do Students Leave University? Qualitative Research At An Italian Higher Education Institution". *European Journal Of Education*, vol 55, no. 3, 2020, pp. 456-470. Wiley, doi:10.1111/ejed.12408.
- Wilcox, Paula et al. "'It Was Nothing To Do With The University, It Was Just The People': The Role Of Social Support In The First-Year Experience Of Higher Education." *Studies In Higher Education*, vol 30, no. 6, 2005, pp. 707-722. Informa UK Limited, doi:10.1080/03075070500340036.