SD²: Software Design Document

1. Project Overview

Niner Connect is an application software that gives students the ability to share and view schedules with other fellow students. They can post their schedules, connect, and communicate with other students who have similar classes. Our stakeholders include professors and students who have desired this in the past. Our software will address the problem by making a site where the students can share their schedules in a social media format and they can also communicate with other students who have the same classes. There will be ways to find other students within the site to connect with and view their schedules by providing the students with profiles.

Group Number 4
Project Name: Niner connect
Description: Ability to share scehdules in a social media format for users to connect with friends and fellow students with similar schedules.

PRODUCT BACKLOG (TO-DO)

Team Members Shawn Kiruba, Adam Dauda, Ian York, Stephen Zargo, Nguyen Pham

Story #	Card Front	Card Back	Sprint Number	Priority	Assigned To	Comments	Estimated Points	Estimated Hours	Estimated Velocity	Actual Velocity
ST-0	As a <user who wants this functionality> I need a <what the user wants> so that <why the user wants it>.	Acceptance Criteria <what the user should be able to do with the functionality>							(Estimated Points / Estimated Hours)	(Estimated Points / Actual Hours)
ST - 9	As a Moderator, I want the site to contain majority of the courses taught in UNCC	Implement a course catalog	Sprint 3	7th	Adam		3	2	1.50	#REF!
ST - 10	As a Moderator, I want our website database to be secure, so that the users passwords are safe.	Site encryption, SSO, is enabled.	Sprint 3	4th	Stephen		3	2	1.50	#REF!
ST - 11	As a UNCC student I would like to implement a system where i can post stuff for other people to like and comment	Implement a forum system for student interaction	Sprint 3	6th	Shawn		8	6	1.33	#REF!
ST - 12	As a UNCC student, I would like the layout to be clean, so that I can easily navigate it.	User friendly application	Sprint 3	10th	Ian		1	1	1.00	#REF!
ST - 13	As a UNCC student, I would like to be able to create an account, so I can save my preferences.	Profile page implemented	Sprint 3	9th	Nguyen		3	2	1.50	#REF!
ST - 14	As a UNCC student, i would like to be able to follow other students	Impliment a follow feature	Sprint 4	8th	Shawn		5	4	1.25	#REF!

Group Number 4
Project Name: Niner connect
Description: Ability to share scehdules in a social media format for users to connect with friends and fellow students with similar schedules.

DOING

Team Members Shawn Kiruba, Adam Dauda, Ian York, Stephen Zargo, Nguyen Pham

Story #	Card Front	Card Back	Estimated Points	Estimated Hours	Actual Hours	Estimated Velocity	Actual Velocity	Sprint Number	Priority	Assigned To	Comments
ST-0	As a <user who wants this functionality> I need a <what the user wants> so that <why the user wants it>.	Acceptance Criteria <what the user should be able to do with the functionality>				(Estimated Points / Estimated Hours)	(Estimated Points / Actual Hours)				
ST - 4	As a UNCC student, I need to be able to use my UNCC credentials so that I can use the website.	Logged into web site	1	1		1.00	#DIV/0!	Sprint 2	1st	Stephen	
ST - 5	As a UNCC student, I want to be able to search by courses, so that I can search for a match.	Searching on an entered search term categorized by courses, returns results reflecting the search term(s)	3	2		1.50	#DIV/0!	Sprint 2	3rd	Shawn	
ST - 6	As a UNCC student, I want to input my class schedule into the app, including course names, times, and locations, so I can find other students with similar schedules.	Searching for peers using class schedules and course to find other students to study with	5	4		1.25	#DIV/0!	Sprint 2	5th	Ian	
ST - 7	As a UNCC student, I want to be able to chat with the student I am matched with, so we can meet up and study.	Implement a working chat bar	3	2		1.50	#DIV/0!	Sprint 2	2nd	Adam	
ST - 8	As a Moderator and a UNCC student, I want the website to load quickly, so I don't waste time.	Less than 5 second load time between pages	1	1		1.00	#DIV/0!	Sprint 2	11th	Nguyen	

Group Numbe 4
Project Name Niner connect
Description: Ability to share scehdules in a social media format for users to connect with friends and fellow students with similar schedules.

DONE

Team Member Shawn Kiruba, Adam Dauda, Ian York, Stephen Zargo, Nguyen Pham

						Sprint 1				Sprint 2					
Story #	Card Front	Card Back	Sprint Number	Priority*	Estimated Points	Estimated Hours	Actual Hours	Estimated Velocity	Actual Velocity	Estimated Hours	Actual Hours	Estimated Velocity	Actual Velocity	Assigned To	Comments
ST - 0	As a <user who wants this functionality> I need a <what the user wants> so that <why the user wants it>.	Acceptance Criteria <what the user should be able to do with the functionality>						(Estimated Points / Estimated Hours)	(Estimated Points / Actual Hours)			(Estimated Points / Estimated Hours)	(Estimated Points / Actual Hours)		
ST - 1	Creating user stories to plan and schedule what needs to get done during		Sprint 1	12th	1	1	1	1.00	1.00					Shawn, Stephen	
ST - 2	Creating low fidelity diagram and the context diagram for our created user stories		Sprint 1	13th	1	1	1	1.00	1.00					Ian, Adam, Nguyen	
ST - 3	Creating UML diagram and some activity diagrams for progression on		Sprint 2	14th	1					1	1	1.00	1.00	Ian, Nguyen	

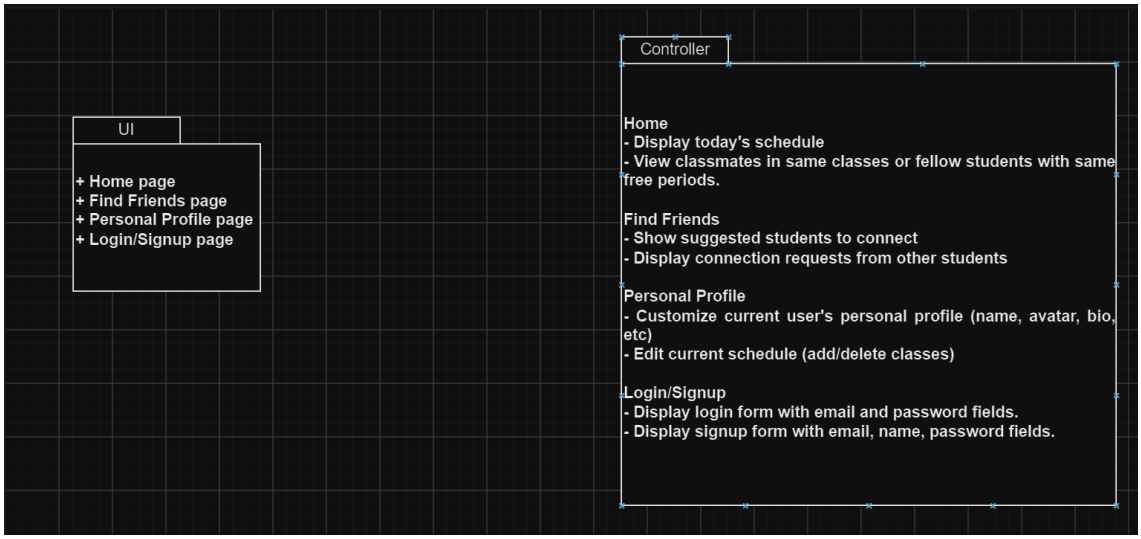
2. Architectural Overview

We went with the client-server architecture because it is straightforward and provides great scalability, security, and reliability that is beneficial to the development process in both short-term

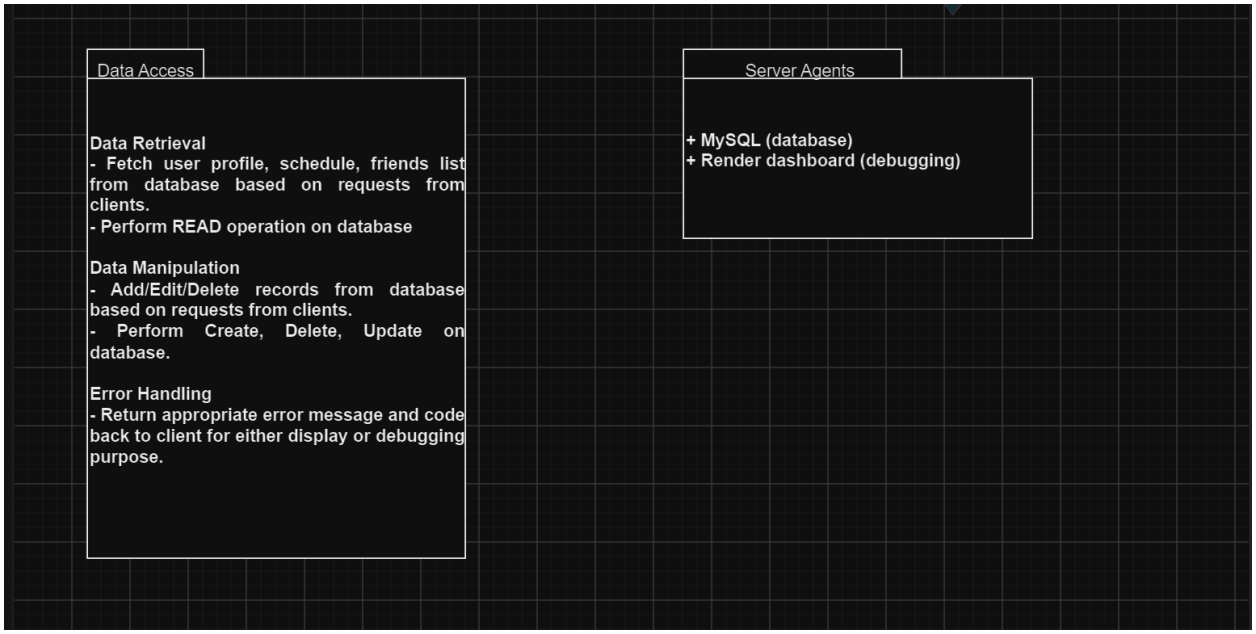
and long-term aspects. Basically having a client and server communicating back and forth helps establish the workflow nicely that is easily scalable and debuggable.

2.1 Subsystem Architecture

- Client:
    - User interface for students to interact with the application, such as viewing schedules, posting their schedules, and communicating with peers.
    - Handling user input and displaying information received from the server in a user-friendly manner.
    - Managing user authentication and session handling (e.g., logging in, logging out).
    - The client communicates with the server to request data, submit user actions (such as posting schedules), and receive updates.

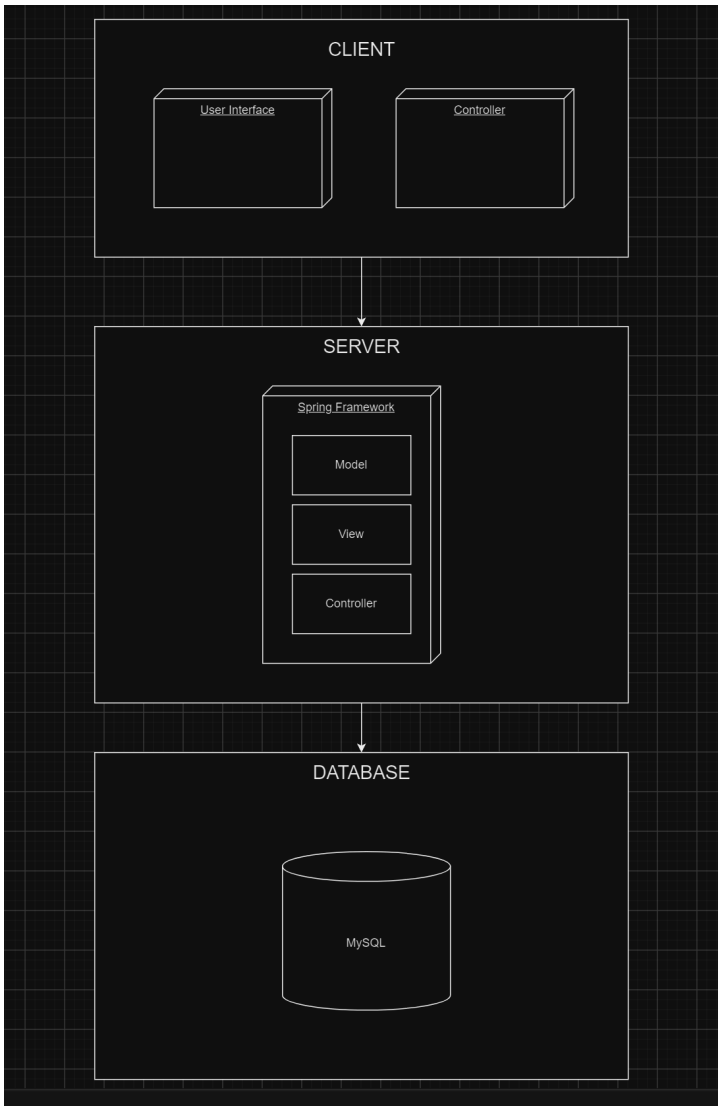


- Server:
    - User authentication: Verifying the identity of users and managing access control to ensure only authorized users can access the system.
    - Schedule management: Storing and organizing schedules posted by students, as well as facilitating schedule sharing between users.
    - Communication management: Handling messages and notifications exchanged between users, such as chat messages or notifications about schedule changes.
    - Data storage: Managing the database where user profiles, schedules, and other relevant data are stored securely.
    - The server processes requests from clients, retrieves or modifies data as needed, and sends responses back to clients.



2.2 Deployment Architecture

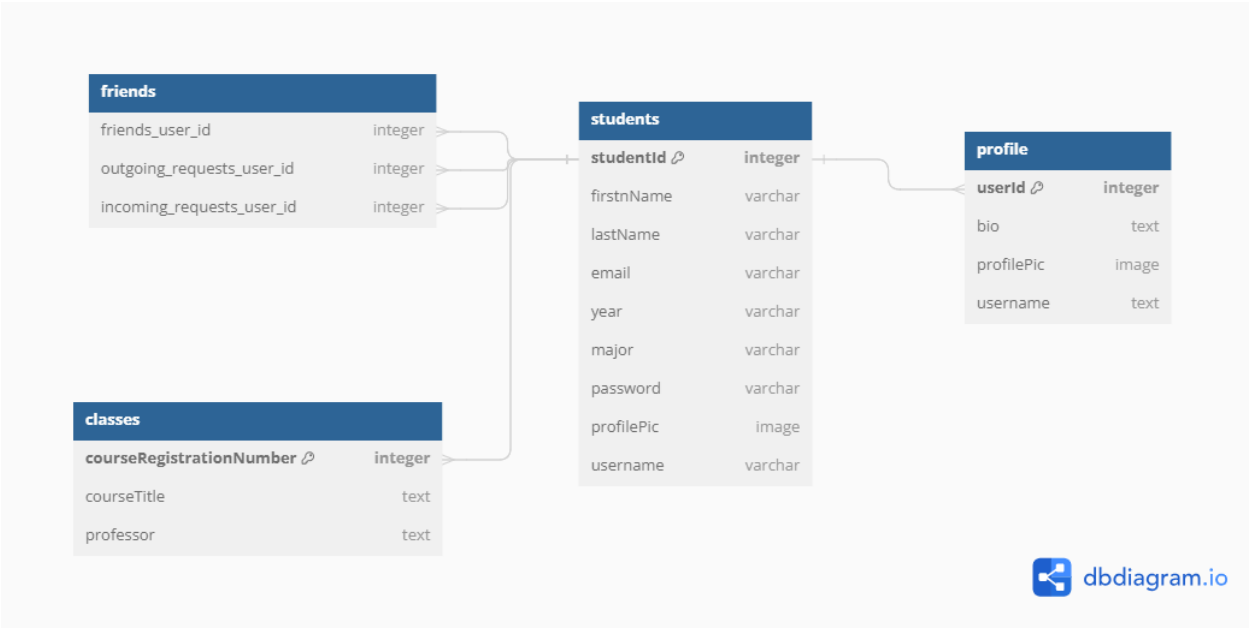
The client will handle data from user requests and communicate with the server to establish appropriate actions (displaying data, sending form data). The server will handle user authentication, persist user sessions, and manipulate data based on user requests, utilizing Spring Framework and MVC (Model, View, Controller). The database will be used to store relevant information and communicate with the server via JDBC Driver.



2.3 Data Model

This section should identify what pieces of information must be stored and your approach to storing data (e.g., flat files, relational database, JSON). If you are using a flat file, you will identify the file format (what are the elements stored in the file, how are they separated, how do you represent the file). If you are using a database for persistent storage, give the database schema (i.e., describe the tables and their columns).

If your system does not need to store any data after a complete execution of the system (e.g. after the user exits), then you do not need to complete this section – just write a single sentence stating "This software does not require persistent data storage."



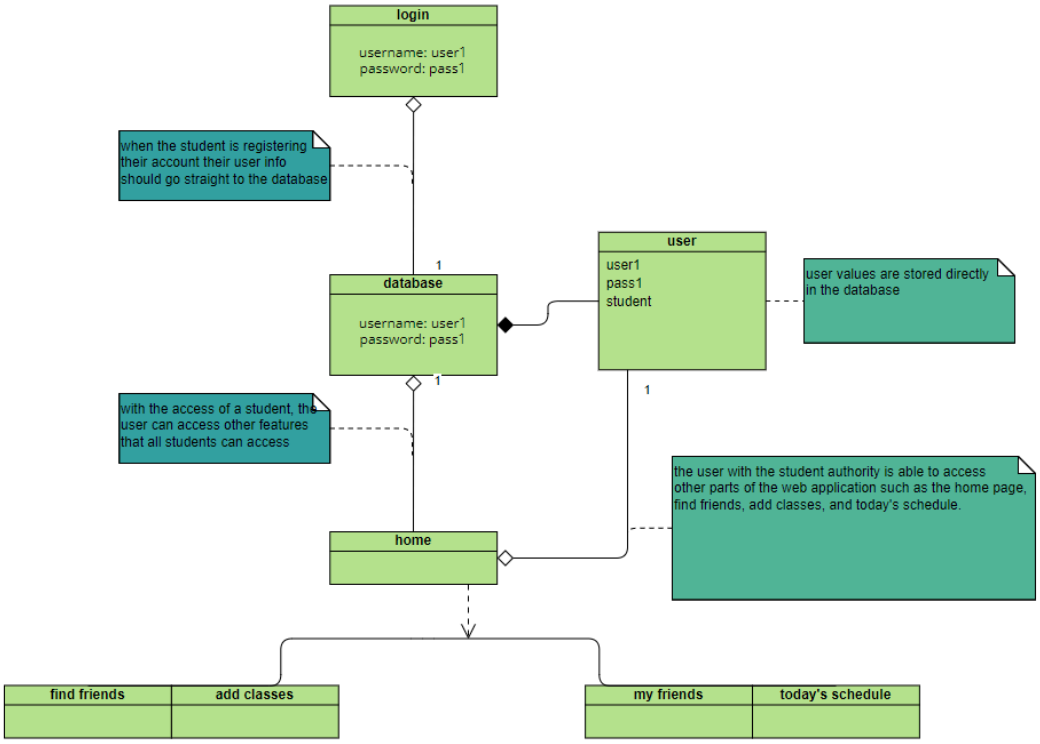
2.4 Global Control Flow

Our system's execution is procedure-driven in many aspects of our web application. There are many parts in our application where the user has to go through the same steps every time. For example, the login page, we have it set up so that the user would have to enter their username and password every time they want to log into their account. Another example would be how if one of our users wants to find other students who have the same schedule they would be able to put in their class name or CRN number to find the class and then a list of students who also are enrolled in that same class. There are no time-controlled actions in our application, although there are many event responses inside of the application that will search and find classes and students that are in similar classes. In terms of concurrency, there are a few different components such as the calendar feature which the user can not only use to view their calendar and their classes for the week but they are also able to view the students that they are friends with that are in those same classes.

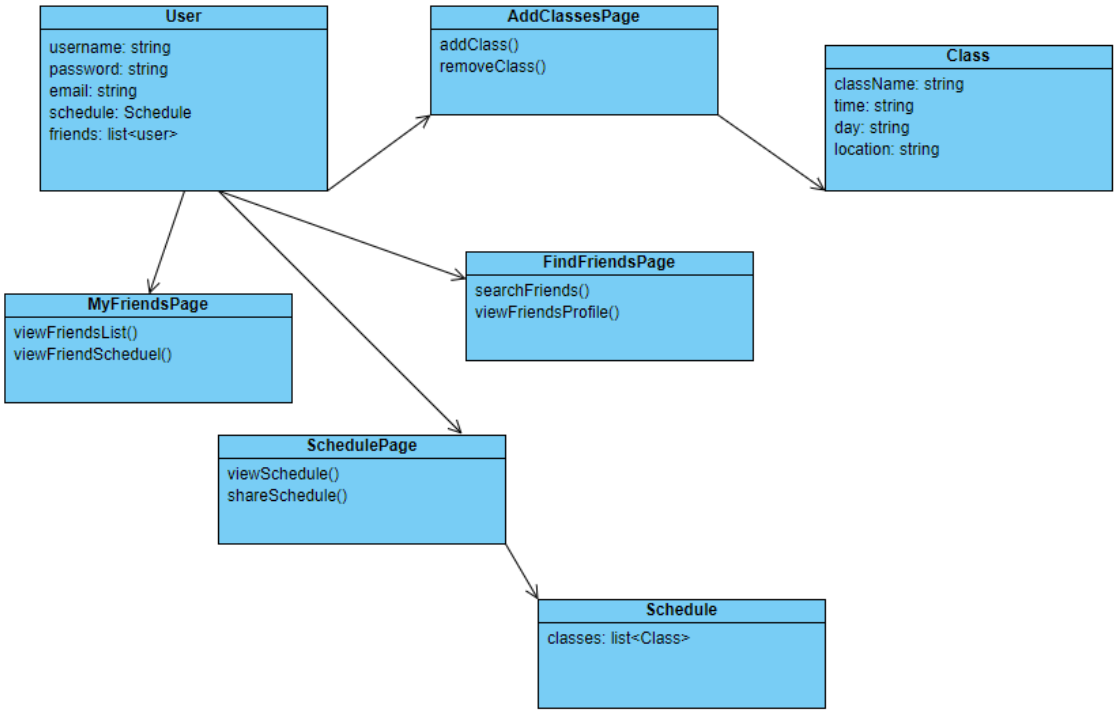
3. Detailed System Design

The design that we went with for the web application is very straightforward and has pages that are immediately accessible to the user as well as effective for the application. We have a navbar that the students can access where they can visit pages such as home, today's schedule, add classes, find friends, and my friends. These were all made for our website where students can share schedules with their friends while also finding more friends that have similar classes in case they need to reach out to other students.

3.1 Static view



This diagram shows a simple feature that we have of our student users logging into their accounts. When they register with our page, we will take that information and add it to our database and give the user the abilities of a student. With this, the student can view and access the many pages that are available to them.



This UML class diagram encompasses most of the features that we have in our web application as of now. As you can see we have a class for Users that stores all the student's information and also gives them access to a student. With this access, they can access pages like my friends, schedule page, find friends, and add classes. Within the My Friends page, they can view their friends list and their schedule. On the schedule page, they can view and share their schedule. On the add classes page, they can add and remove classes from their schedule. On the Find Friends page, they can search and view friends' profiles. Within the schedule page, you can also see a list of classes, and in the add classes page, you can see the detailed information about those classes.

3.2 Dynamic view

You must show the design of your system's behavior using UML sequence diagrams. These sequence diagrams should show the time-ordered sequence of interactions among classes to support an important system function. Your sequence diagrams should be consistent with the class diagrams given in Section 3.1. In other words, you should not have participating objects in an interaction that do not appear in a class diagram; if you find that this is the case, you should go back and revise your class diagram to include the new element. You may supplement your sequence diagrams with state-transition diagrams or activity diagrams (useful for describing algorithms), but these are not required.

Make sure to add your test plans, Previous and current sprints, and brief sprint reviews as well as any additional items you feel are helpful you feel are helpful at any point. Remember, it's not necessary to add documents that are not providing useful information. Only necessary items. If you don't have them ready, you can add them later using version control in Git Hub.

Submission:

Submit your design documents inside your team's GitHub repo. Moreover, **submit a PDF document in Canvas along with the link to the repo; we will not grade any other document type**. Name your electronic

submission as follows: **Team<number>_D2.pdf.** Submit your team assignment via Canvas. Only one team member needs to submit the document.

Tips:

Provide supplemental text to explain your diagrams through captions for them! Make sure that you have described, somewhere in the document, the responsibilities of the elements (e.g., components/modules/classes) in your architecture/class/sequence diagrams. You should describe your design decisions that led you to this design, including a discussion of any alternative designs you considered but discarded. Providing these kinds of descriptions helps in understanding your design (as such, they can have a positive impact on your grade!). *Where to Stop/When to Stop Drawing Interaction Diagrams*? Generating a sequence diagram for the most important user stories is typically a good way to start. If you find that you have a bunch of sequence diagrams that essentially repeat the same interaction, then you are not creating useful models, just redundant ones. In this case, generalize the interaction and provide supplemental text that explains how and where the generalized interaction can be applied to capture multiple user stories/use cases. **Remember, you should model only what is needed and useful as discussed in the class.**

Apply Design Principles! You have learned about the importance of modules with high cohesion, a design with low coupling, and the benefits of relying on abstraction versus a concrete realization (e.g., application logic communicates with a hardware abstraction layer instead of directly with devices). Make sure that you apply these principles and clearly explain how your design achieves them.

Proofread your documents! Everyone on your team should proofread the document before it is submitted. We must be able to understand your design to evaluate it, so you must take care in communicating your design. If you are not skilled in technical writing, plan in advance so that you can ask someone to proofread it for you. The university has a writing resource center that may be helpful to you.

What UML tools should I use to draw the diagrams? Any design tool may be used to draw your UML diagrams such as Draw.io and Lucidchart as we discussed in the class.

Be aware that while drawing programs may provide template tools for creating UML diagrams, those templates may not meet the diagramming guidelines we discussed in class. For instance, some versions of Microsoft Visio provided the wrong arrow for an “extends” relationship in the past for use cases. You should check to make sure your diagram meets the standards discussed in class and make manual edits in the drawing program if necessary to adhere to those standards.