The D.A.M. Documentation

By the Darden Blue Jackets (Chaison, Chance, Colby, Zach)

Table of Contents

Project Overview

The Digital Academic Manager is intended to be an all in one webApp for universities to use to manage courses, online chats between students, grading, assignments, and more. This document is meant to serve as the design guide and repository of all development concepts.

DAM should at least provide the necessary tools for…**A**) Faculty (Professors/Instructors) to assign course work, grade submissions, and communicate with their students…**B**) Students to search courses, enroll in courses, view and submit course work, and communicate with their teacher, fellow students, and assigned teams.

Project Cost Estimation

Acquisition Phase Distribution

Phase	Effort (Person-Months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	7.4	1.9	3.8	$37,226
Elaboration	29.8	5.8	5.1	$148,903
Construction	94.3	9.7	9.7	$471,527
Transition	14.9	1.9	7.6	$74,452

Software Effort Distribution (Person-Months)

Phase	Inception	Elaboration	Construction	Transition
Management	1.0	3.6	9.4	2.1
Environment	0.7	2.4	4.7	0.7
Requirements	2.8	5.4	7.5	0.6
Design	1.4	10.7	15.1	0.6
Implementation	0.6	3.9	32.1	2.8
Assessment	0.6	3.0	22.6	3.6
Deployment	0.2	0.9	2.8	4.5

Maintenance

Annual Maintenance Effort = 2.4 Person-Months

Annual Maintenance Cost = $11,923

Key Features

Feature 1: Tutor Temple

Students will be able to post code snippets and general questions onto the site which will allow other students to give assistance or feedback. However, the one catch to this is that if student A were to receive help from student B, there would be an email sent to the professor that would indicate that student A has received help from Student B on Assignment 1. This would help prevent Student A and B from violating the honor code and also prevent any confusion on the professor’s end.

Students who voluntarily tutor will gain points added to their account. Once a certain amount of points has been reached, these students will be rewarded with the exclusive Darden Blue Jacket that highlights their passion in helping others succeed.

Feature 2: File Archive

Students will have access to file submissions from previous classes. This feature will only be located in the submission process and in the student account page. Once the student selects the “file archive” option, it will allow the student to sort based on semester or course code to make it quick and easy (Ex. Spring 2023 or CSC 1010). Any submissions in that semester or course will then be displayed to the user.

Archives of files will automatically be removed with notice every year to clear storage, so students will eventually have to backup things they wish to keep before then.

Feature 3: Mobile App

Better integration with the mobile landscape saves users the time and frustration from having to use their mobile or web browser to access DAM. The app includes all reasonable functionality that is to be expected such as viewing classes and assignments, submitting assignments, and viewing grades.

Users will have a one-time verification code sent to their phone number to make the process of using the mobile app more swift.

Feature 4: Integrated Document Editor:

Students and instructors alike will have access to a simplistic document reader and writer to ensure that their documents have been uploaded correctly. Instructors will also be able to make simple annotations to submitted work.

Feature 5: Grade Analysis

DAM includes provision for access to a grade analysis section. Users will be able to input “what if?” scenarios for grades to see how much an assignment would impact their overall grade for the course. Compared to MCS Moodle, DAM’s grade report system is revamped and presents a more accurate view of graded work.

Feature 6: User Customizable

Users will have the option to apply custom themes that make their experience on DAM more satisfactory and engaging (Ex. Light and dark mode).

Feature 7: Universal Functionality

Users will be allowed to select the language they feel most comfortable with in order to make their experience on DAM more natural (Ex. English, Korean, Spanish). - However, the contents inside of files (Such as a pdf) posted by professors will only contain the language that was already in it. There will be no option to change the language since the file is inaccessible.

Feature 8: Free and Open Source

DAM offers free of charge and is open source

Feature 9: Dexter the Customer Support Bot

Dexter is a 24/7 functional bot that will assist users with their questions, comments, or concerns. He will be located in the login page in the event users forget their login information. He will also be located in DAM’s homepage. He has the ability to remind users of due dates, updates, etc. Dexter will not respond to users who ask him irrelevant topics. In the event Dexter cannot satisfy the user, he will tell the user to contact one of the members of the Darden Blue Jackets.

Task Sets

Set of tasks users should be able to accomplish inside the class:

- Access the log in/out function
- View Calendar
- View and access all of their enrolled/instructed courses ● View upcoming due assignments
- Have access to the Course Enrollment & search function ● View upcoming site updates & news

Set of tasks users should be able to accomplish inside the class:

- Locate each day of each week
- Locate the overall grade in the course, recent messages, and upcoming assignments due • Locate the course information (Course name, professor, description, etc) • Sort the month, day, week, or year of any course using the sort option • Find assignments posted for each day of each week (If any)
- Expand and collapse the days of each week for better readability
- Unenroll from the course and logout from the website

Set of Tasks File Archive

- Clearly display classes in ascending order by enrollment date.
- Provide a feature to expand and collapse a class to display files. This will tighten up the user interface.
- Provide easy download for past files.

Set of tasks the user should be able to accomplish when submitting a file

- Identify the Assignment
- Upload a file through drag and drop
- Upload a file through the “Add File” dialog box.
- Upload multiple files through either method
- Choose to commit the submission to the File Archive
- Submit the assignment
- Cancel the submission

Design Models

MENTAL MODEL

Description of every step that comes to mind when asked “What does your website look like?”

1. There’s a login page that will prompt for username, password, and 2FA
2. Once you’re logged in, there will be a list of your enrolled courses, a simple navigation bar to the left, and a calendar to the right
3. When you click on a course, there will be posted assignments for each day or each week, updated messages, the ability to expand and collapse each day or each week, and the option to sort the month, day, week, or year in order to achieve a purpose
4. When you’re submitting an assignment, there will be the classic “drag and drop a file” option, “add file” option, and a “file archive” option. The file archive is checked by default 5. There are unenroll and logout options at the very bottom of every page.

USER MODEL

Site Wide

Language and Appearance (such as Dark & Light modes) options should be available to the user in the page margins (likely the top left corner) of all pages.

Top right corner should give access to the User’s account and log in/out functions

Dashboard

When logged in, the dashboard should serve as the User’s main information overview and navigation hub, being able to view all upcoming assignments, messages, calendar, and all of their enrolled/instructed courses.

Able to find links to;

- Current Courses (either enrolled or instructing)
- Upcoming Assignments by Course
- Course Search
- Messenger
- Submission Archive

File Archive

Student Users can come here to find all of their previously submitted assignments for redownload even after they are no longer enrolled in that course.

Users should be able to search by course code, limited to their user ID. - Instructor Students can search across all archived submissions by course code + Student ID to either review or inspect past assignments for suspected acts of cheating.

Inside a Class

Users should be able to easily view their current grade average for the course. - Course schedule/events/assignments should be divided into weekly drop-tabs that can be opened or closed as the user wishes.

Submission

Student Users should be able to select from submission options, such as a file upload, drag-and-drop, or direct text submission.

Instructor Users should be able to set the following; max/min submission size, submission extension type requirements, and open/closing time.

IMPLEMENTATION MODEL

Dashboard

The dashboard will be the main hub for the website, showing the user all of the classes they are registered in, upcoming assignments in each of those classes, their profile and information, and also giving links to the file archive and grades submenus. Clicking on a class will take the user to the class's page, clicking on an upcoming assignment for a class will take them to the submission page for said assignment, and clicking on their profile will take them to the profile page. The file and grades menus are accessed in the same way by clicking the labeled link for each respectively. This view is the same for instructors in that they can view the courses they are teaching, but there are also options for adding new courses and course administration. The same for administrators.

The dashboard is consistent with the end user's idea of the LMS, they have access to everything they need to complete their coursework without any unnecessary things crowding the screen. The instructors and those who manage the system have interfaces that allow them to modify the system without overcomplicating the process.

File Archive

The file archive is where users can access prior submission for any course they have taken in the past. Courses will be organized chronologically, with the newest classes at the top of the list. Each class will have a collapsible menu detailing all of the work done in the course, and the options to download each individual file or all coursework in an archive file. A similar view will be available for instructors where they will be able to see what was assigned in prior sections of classes.

Sometimes users are proud of their work for a course and want to showcase it or iterate on it, so this accommodation was made based on end user wants. For instructors it can be helpful if they want to base a semester's section on a previous semester's one.

Inside a Class

Inside classes the course of work will be divided into a day-by-day basis, with a section devoted to each day where there will be assignments and other subject matter for the day. There will be boxes for each day of the week, M-F, as well as headers at the top of the page designating what class it is and what week of the semester it is. Prior weeks can be accessed through a submenu at the bottom. Instructors have a master view wherein they can modify each day of each week to include attachments and assignments.

Users expressed an interest in a day-to-day based LMS so the decision was made to organize classes for each day of the week.

Submission

For assignments that take submission the submission page will open. There the student can select to browse for a file on their computer or to drag-and-drop their submission. The student can also select whether or not they want to commit their submission to their file archive (though, this box will be checked by default). There will also be a text box where the student can make any comments regarding the assignment. The instructor can also specify in the creation of an assignment what types of files will be accepted in the submission menu, and the accepted files will show to the student as "File must be: [file format]." Once a file has been uploaded the options to “Submit” or “Cancel” will then be made available to submit the assignment to cancel the submission.

The user wants to be able to submit their work in a quick and easy manner, so incorporating drag and drop into the page can drastically reduce the amount of time needed when prowling through directories on a computer.

Use Cases

USE CASE	User Profile	
Goal in Context	Create a profile to help people get to know each other	
Scope and Level	Students Teachers	
Preconditions	Student is new	
Success End Condition	User profile is completed and updated	
Failed End Condition	Students will not have a user profile	
Primary Actor	Students	
Secondary Actor	Teachers	
Trigger	Selects user profile button	
Description	**Step**	**Action**
	1.	Name, Email, Phone Number is shown
	2.	User photo is shown
	3.	About me is shown
Extensions	**Step**	**Branching Action**
	1a	Student can enter information in the "About me" section
	1b	Student can upload a picture to their page
Sub-Variations		**Branching Action**
		Students can change theme
		Students can set their page to private or public

Priority	Low
Performance	Seconds
Frequency	Low
Channels to actors	User Profile Button
OPEN ISSUES	N/A

USE CASE	Upcoming Assignment	
Goal in Context	Provide a side bar of upcoming assignments	
Scope and Level	Students Teachers	
Preconditions	Students need reminders for their upcoming assignments	
Success End Condition	Students will always be up to date on their upcoming assignments	
Failed End Condition	Students will forget assignments	
Primary Actor	Students	
Secondary Actor	Teachers	
Trigger	Student Logs in	
Description	**Step**	**Action**
	1.	Upcoming Assignments for all classes is shown on the side
	2.	Assignments are arranged with the earliest due date first

Priority	High
Performance	Seconds
Frequency	Often
Channels to actors	Side Bar

OPEN ISSUES	N/A

USE CASE	Tutor Temple	
Goal in Context	Provide a place for students to tutor other students	
Scope and Level	Students	
Preconditions	Students need a place online for tutors	
Success End Condition	Students have a place online for tutors	
Failed End Condition	Students will have to be tutored on campus	
Primary Actor	Students	
Secondary Actor	Tutors	
Trigger	Student selects tutor temple button	
Description	**Step**	**Action**
	1.	Tutor temple is shown
	2.	Topics are shown with most recent at the top
	3.	Poster, course, description is shown under the topic
Extensions	**Step**	**Branching Action**
	1a	Students can create a new topic
	1b	Students can reply to other topics
Sub-Variations		**Branching Action**
		Professor of the class is sent an email when someone replies to a post from their class

Priority	Moderate
Performance	Seconds
Frequency	Often
Channels to actors	Tutor button
OPEN ISSUES	N/A

USE CASE	Team Chat	
Goal in Context	Provide students with a useful team chat	
Scope and Level	Students Teachers	
Preconditions	Students need a chat room for group work	
Success End Condition	Students have a functional chat room for teams	
Failed End Condition	Students will use discord or text	
Primary Actor	Students	
Secondary Actor	Teachers	
Trigger	Class chat is selected	
Description	**Step**	**Action**
	1.	Team chat is displayed at the bottom of the screen
	2.	Messages are shown based on recent posts
	3.	Team members are displayed on the side
Extensions	**Step**	**Branching Action**
	1a	Student types in a post
	1b	Students message is posted to the group

	1c	Replies will be directly under the post
Sub-Variations		**Branching Action**
		Teachers can be added to a team

Priority	Moderate
Performance	Seconds
Frequency	Often
Channels to actors	Class Chat
OPEN ISSUES	N/A

USE CASE	Site News	
Goal in Context	Provide readable news on page	
Scope and Level	Students Teachers	
Preconditions	Student has logged in	
Success End Condition	News is shown	
Failed End Condition	N/A	
Primary Actor	Students	
Secondary Actor	Teachers	
Trigger	Student logged in	
Description	**Step**	**Action**
	1.	Student logged in
	2.	Site news is shown

Priority	Moderate
Performance	Seconds
Frequency	Moderate
Channels to actors	Nav Bar
OPEN ISSUES	N/A

USE CASE	Service Bot	
Goal in Context	Create a bot to help students	
Scope and Level	Students	
Preconditions	Student has a question	
Success End Condition	Bot helps student	
Failed End Condition	Bot provides useless info	
Primary Actor	Student	
Secondary Actor	N/A	
Trigger	Student has a question	
Description	**Step**	**Action**
	1.	Bot is shown
	2.	Student submits a question
	3.	Bot provides feedback

Priority	Low
Performance	Seconds
Frequency	Often
Channels to actors	Side Bar

OPEN ISSUES	N/A

USE CASE	Login to Account	
Goal in Context	Provide a simple login	
Scope and Level	Students and Teachers	
Preconditions	Students need a simple login page	
Success End Condition	Student successfully logs in	
Failed End Condition	A Failure to login page will be displayed	
Primary Actor	Students	
Secondary Actor	Teacher	
Trigger	Student opens up the web page	
Description	**Step**	**Action**
	1.	Page is shown
	2.	Username is selected
	3.	Username is typed
	4.	Password is selected
	5.	Password is typed
Extensions	**Step**	**Branching Action**
	1a	User is successfully logged in
Sub-Variations		**Branching Action**
	1b	User is prompted with incorrect username/password
	1c	After 5 unsuccessful tries the

		user will have their account locked

Priority	High Priority
Performance	Seconds
Frequency	Very Often
Channels to actors	Opening of Webpage
OPEN ISSUES	N/A

USE CASE	Class Overview	
Goal in Context	Create a user friendly class overview section	
Scope and Level	Students, Teacher	
Preconditions	Student has created an account and is enrolling into an account	
Success End Condition	Course overview is easily understood	
Failed End Condition	The course overview page is not user friendly	
Primary Actor	Student	
Secondary Actor	Teacher	
Trigger	Account created	
Description	**Step**	**Action**
	1.	Selects a course to add
	2.	Course overview is shown

Priority	High
Performance	Seconds

Frequency	Moderate
Channels to actors	Course overview button
OPEN ISSUES	N/A

USE CASE	Enrollment	
Goal in Context	Provide an easy way to enroll into a course	
Scope and Level	Students Teachers	
Preconditions	Students are logged in	
Success End Condition	Student is enrolled	
Failed End Condition	Student doesn't get enrolled	
Primary Actor	Students	
Secondary Actor	Teacher	
Trigger	Student selects course	
Description	**Step**	**Action**
	1.	Course code and name are shown
	2.	Course Prerequisites are shown
	3.	Course Textbook is shown
	4.	Teacher biography is shown
Extensions	**Step**	**Branching Action**
	1a	Student selects request enrollment
Sub-Variations		**Branching Action**
		Professor has the opportunity to deny or accept student

Priority	High
Performance	Seconds
Frequency	Moderate
Channels to actors	Course code button
OPEN ISSUES	N/A

USE CASE	Class Chat	
Goal in Context	Allow students to post to a class chat	
Scope and Level	Students	
Preconditions	Students need a place to chat with other students	
Success End Condition	Students have a place to chat with other students	
Failed End Condition	Students wont have a place to chat	
Primary Actor	Students	
Secondary Actor	N/A	
Trigger	Class chat is selected	
Description	**Step**	**Action**
	1.	Class chat is shown
	2.	Latest Message is shown
	3.	Student writes a post in the message bar
	4.	Students message is posted
Extensions	**Step**	**Branching Action**
	1a	Replies are shown under the specific post

Priority	Moderate
Performance	Seconds
Frequency	Moderate
Channels to actors	Class Chat button
OPEN ISSUES	N/A

USE CASE	Submission Archive	
Goal in Context	Allow students to access files submitted	
Scope and Level	Students	
Preconditions	Students need to easily access everything they have submitted	
Success End Condition	Students are able to access old files	
Failed End Condition	Students will be unable to find old files	
Primary Actor	Students	
Secondary Actor	N/A	
Trigger	Submission archive is selected	
Description	**Step**	**Action**
	1.	Submission archive is shown
	2.	Assignment title is shown
	3.	Course, Authors, Date Submitted, and File Type is shown
Extensions	**Step**	**Branching Action**
	1a	Students have the option to download file
Sub-Variations		**Branching Action**

		Files are organized with most recent file at the top

Priority	Very High
Performance	Seconds
Frequency	Often
Channels to actors	Archive button
OPEN ISSUES	N/A

USE CASE	Login Recovery	
Goal in Context	Recover a locked account	
Scope and Level	Students	
Preconditions	Account becomes locked or password is forgotten	
Success End Condition	Account is recovered	
Failed End Condition	Account is still locked	
Primary Actor	Student	
Secondary Actor	N/A	
Trigger	Student is locked out of account or forgotten their password	
Description	**Step**	**Action**
	1.	Student clicks send email
	2.	A 6-digit code is sent via email
	3.	Student enters the code into the available space
	4.	Student clicks send text

	5.	A 6-digit code is sent via text
	6.	Student enters code into the available space

Priority	High
Performance	Seconds
Frequency	Moderate
Channels to actors	Forget password page
OPEN ISSUES	N/A

USE CASE	Login Account Creator	
Goal in Context	Provide 1 Easy Account Creation	
Scope and Level	Students	
Preconditions	Students need an account created	
Success End Condition	An account is successfully created	
Failed End Condition	Account is not created due to: Poor password/ Taken username	
Primary Actor	Students	
Secondary Actor	N/A	
Trigger	Student selects the Create Account hyperlink	
Description	**Step**	**Action**
	1.	Student enters email
	2.	Student enters phone number
	3.	Student enters a special username

	4.	Student enters a valid password (8 Characters, Special Character, and Number)
	5.	Student reenters a valid password
Extensions	**Step**	**Branching Action**
	1a	Account is successfully created
Sub-Variations		**Branching Action**
	1b	Username in use
	1c	Display “Username already taken”
	1d	Password to weak
	1e	Display “Password to weak please use: 8 or more characters, 1 special character, 1 number”

Priority	High Priority
Performance	Once
Frequency	Rare
Channels to actors	Create account button
OPEN ISSUES	Check other accounts for username.

USE CASE	Navigation
Goal in Context	Make navigation seamless and quick
Scope and Level	Students
Preconditions	Students need minimal clicks to sections
Success End Condition	Students will have a user friendly navigation bar

Failed End Condition	Navigation will be redundant and take time	
Primary Actor	Students	
Secondary Actor	Teachers	
Trigger	Student logs in	
Description	**Step**	**Action**
	1.	Dashboard is shown
	2.	A class is selected
	3.	A class is shown
Extensions	**Step**	**Branching Action**
	1a	Grades selected
Sub-Variations		**Branching Action**
		Grades are shown

Priority	High Priority
Performance	Seconds
Frequency	Very often
Channels to actors	Interaction side navigation Bar
OPEN ISSUES	Buttons must be clear

USE CASE	Submitting
Goal in Context	Make submitting easy
Scope and Level	Students
Preconditions	Students need minimal clicks to submit
Success End Condition	Students will be comfortable submitting any type of file

Failed End Condition	Submitting will take to much time	
Primary Actor	Students	
Secondary Actor	N/A	
Trigger	Student completes an assignment	
Description	**Step**	**Action**
	1.	Specific assignment is clicked
	2.	Submission box is show
	3.	Files can be submitted
Extensions	**Step**	**Branching Action**
	1a	Student can submit via drag and drop or accessing the file system

Priority	High Priority
Performance	Seconds
Frequency	Often
Channels to actors	Submission Box
OPEN ISSUES	N/A

Test Cases

Test Case: Security

Objective: Prevent unauthorized users from obtaining access to students' and teachers' confidential information

Description: Users will enter their username and password into the login screen displayed. Wrong usernames or passwords entered 5 consecutive times will result in session timeout for 5 minutes. From there, an additional 5 minutes will be added onto it if the same thing keeps happening. The account trying to be logged in will automatically delete if it reaches the 30 minute mark. An email of this will be sent to the email of that user. Note that there will be a "forgot username or password" option if the user is actually legit.

Process:

- Student successfully finds the webpage
- Student attempts to log in
- Login appears incorrect
- Student attempts to log in 4x and fails
- Disables the ability to enter in username and password for 5 minutes
- After 5 minutes, student repeatedly fails logging in
- Account automatically deletes
- An email is sent to the email of that user

Expected results:

- Student successfully finds the webpage

- Student cannot log in the first 5 consecutive attempts

- Disables the ability to enter in username and password until the 5 minute mark is over • Once it has been disabled 30 minutes, account will be deleted successfully • Email will be sent out to the email of that user

Test case - Calendar

Objective -

Ensure that the calendar is in modern time and allow the user to go back or forward into different months

Description -

Calendars are really necessary for academics because it gives an overall perspective of the workload for each month, reminds users of due dates, etc. Hence, why our calendar must be thoroughly tested.

Progress -

- User successfully logs into DAM

- User locates the calendar on the right

- User is displayed the current month

- User clicks on the arrows to navigate through the months

Expected results -

- User will be on the DAM website

• User will successfully log in

• User will locate the calendar on the right within seconds

• User will click on the arrows to navigate through the months

• Calendar will correctly navigate through the months as the user clicks on the arrows

Test case - Change theme

Objective -

Ensure that changing themes work throughout the website

Description -

User preference is key to establishing user-friendly apps which is why allowing them to set the overall website theme to their preference would provide exactly that.

Progress -

• User logs into DAM

• User locates the person icon at the top right corner of the taskbar

• User selects the person icon

• User sees the “Change theme” option underneath his or her profile pic

• User selects that button

• The button will drop down options for various types of themes

• User will select whatever theme he or she wants

• The theme throughout DAM will adjust to whatever was selected by the user

Expected results -

• User is on the DAM website

• User successfully logs in

• User successfully locates the person icon at the top right corner of the taskbar and selects it • User locates and then selects the “Change theme” option within seconds

• The button successfully drops down options for various types of themes

• The theme selected by the user will be displayed throughout the entire website • User will be happy

Test case - Class chat link

Objective -

Make sure the class chat link takes the user to the correct location

Description -

Links can sometimes get mixed up and users could get sent to the wrong direction if that’s the case. Therefore, thorough testing to make sure the link has the correct location must be accomplished.

Progress -

• User logs into DAM

• User locates the “My enrolled courses” box on the left

• User selects a course

• The course material is displayed to the user

• User locates the “class chat” link next to the course title

- User selects the link
- Link takes the user to the correct location

Expected results -

- User is on the DAM website
- User successfully logs in
- User locates the “My enrolled courses” box on the top left and selects the appropriate course • User locates and then selects the “class chat” link within seconds
- Link takes the user to the class chat

Test Case - Logging in

Objective -

Successfully login

Description -

Users will be required to enter their login credentials before accessing the website. In the event a user has forgotten their login info, there will be a “Forgot your DAM login information?” option below that will give the user the option to receive a 6 digit code through email or text. If the user is new to DAM, there will be a “Create account” option that they will select. It will then require the new user to enter their email, phone number, username, and password (Twice for verification).

Process 1 (Existing user) -

• User is on the DAM website

• User enters login info

• User logs in without any trouble (With the exception of human error such as misspelling)

Process 2 (New user) -

• User is on the DAM website

• User selects the “create account” option

• User enters their email, phone number, username, and password (Twice for verification) • User successfully creates an account and is now successfully logged into DAM

Process 3 (Forgot DAM login info) -

• User is on the DAM website

• User attempts to log in with their assumed login info

• User is not able to log in

• User selects the “forgot your DAM login information?” option

• User chooses to receive a 6 digit code for account recovery through email or text • 6 digit code will successfully send to the selected option

• User enters in the code

• User successfully recovers their account and successfully logs in

Expected Results -

• User successfully finds the DAM website

- User successfully logs in by selecting the appropriate option that applies to them (Returning user, new user, or forgotten login info)

Test case - Navigation bar

Objective -

Ensure that the hyperlinks inside the navigation bar function properly

Description -

Navigation bar is crucial to our design because of the links that are provided inside of it. Some links that it contains are the home page, tutor temple, and submission archive. Having these links to the side of our website would provide stability and conveniency to DAM's homepage because of its quick accessibility.

Process 1 (Home page) -

- User successfully logs in
- User locates the navigation bar on the left
- User selects the "home page" link
- The link will take the user back to the home page

Process 2 (Tutor temple) -

- User successfully logs in
- User locates the navigation bar on the left
- User selects the "tutor temple" link
- The link will take the user to Tutor Temple where tutoring services are held

Process 3 (Submission archive) -

- User successfully logs in
- User locates the navigation bar on the left
- User selects the “submission archive” link
- The link will take the user to his or her submission archive that consists of all of their previous files submitted from previous courses

Expected results -

- User is on the DAM website
- User successfully logs in
- User is able to find the navigation bar on the left within seconds
- User is able to select any of the links in the navigation bar
- All of the links in the navigation bar will successfully take the user to the designated location

Test case - Posts in class chats

Objective -

Ensure that posts being made in class chats are visible to other users

Description -

Communication is key in a classroom, especially for team-oriented classes such as MCS. Implementing class chats as a public and team chat was very necessary for DAM.

Progress -

- User logs into DAM

- User locates the “My enrolled courses” box on the left
- User clicks on a course and is displayed course material
- User locates the “class chat” link and clicks on it
- User is brought to the class chat
- User locates the option to post
- User will type in something and post it once he or she is done
- The post will be visible to all users associated with the poster’s class

Expected results -

- User will be on the DAM website
- User will successfully log in
- User will locate the “My enrolled courses” box on the left within

seconds • User will click on his or her course

- User successfully locates and selects the “class chat” link within

seconds • Link successfully takes the user to the class chat

- User locates the option to post in a class chat within seconds
- User successfully types in something and successfully posts it when

finished • Post is visible to all users associated with the poster’s class

Test case - Privacy settings for user account

Objective -

Ensure that privacy settings match up with the user’s preference

Description -

People value their privacy which is why giving the option to set it as that will prevent their information from being open to the public. Some information can be their phone number or email.

Progress -

• User logs in

• User locates the person icon in the top right corner of the taskbar

• User selects the person icon

• User is displayed their info and the “Set to private/public”

button on the right • User selects that button

• User is given a choice to set to either choice

• User selects a choice

• Profile gets set to that choice

Expected results -

• User successfully is on the DAM website

• User successfully logs in

• User successfully locates the person icon in the top right corner of the taskbar and selects it within seconds

• User successfully locates the “Set to private/public” button on the

right within seconds • User successfully selects their preference of

choice

• Profile successfully gets set to the choice of the user

Test case - Request course enrollment

Objective -

Ensure that only users who have the class on their schedule are allowed to enroll

Description -

Having users in classes they are not apart of can cause confusion to faculty members.

Progress -

• User successfully logs in

• User locates the search bar on the left side of the home page

• User clicks on the search box and types in the course either by

code or name • User is displayed classes based off the search

• User is presented a “request enrollment” button for each course

that pops up • User selects the “request enrollment” button

• The request gets sent to the designated faculty member

Expected results -

• User is on the DAM website

• User successfully logs in

• User successfully locates the “search courses” box on the left side of the home page within seconds

• User successfully clicks on the search box and types the course

code or name • A list of classes that matches the search is displayed to the user

• User selects the “request enrollment” button for any appropriate classes • The request successfully gets sent to the designated faculty member • If the faculty member accepts the request, the student is successfully enrolled in the course

◦If the request was denied, the student will not be enrolled

Test case - Search courses

Objective -

Ensure that the “search courses” box near the top left actually functions properly

Description -

Having a quick and convenient way for users to find their classes is essential to the website. Users will go to the “search courses” box near the top left and type in either the course code (CSC 1010) or the course name (Intro to Python). It’s vital for users to be as specific as possible, especially if users choose to search by course name. This helps prevent any confusion between classes. For example, typing in “Intro to Python” would only result in that class. However, only typing “Python” would result in the “Programming in Python” class as well. It is also important to note that human error may cause unexpected results (Like misspelling words in course names).

Process -

• User is on the DAM website

• User successfully logs in

• User will find the “search courses” box near the top left

• User will click inside the search box and type in the course either by code or name • A list of classes will be displayed on the screen that were related to the user entry • User will select his or her appropriate class

Expected results -

- User will successfully find the DAM website
- User will successfully log in
- User will successfully find the “search courses” box near the top left within seconds • User will successfully click inside the search box and type in the course either by code or name • Appropriate classes that match the user entry will be displayed onto the screen • User will successfully select his or her appropriate class

Test case - Dexter the service bot

Objective -

Ensure that Dexter the service bot responds appropriately to user interaction

Description -

Having a service bot 24/7 available to users is beneficial after business hours. Dexter can respond with feedback to questions, comments, or concerns related to MCS topics. Dexter will not respond to irrelevant topics. With all that being said, this is why it is vital to thoroughly test Dexter the service bot to ensure its functionality in our website.

Process -

- User successfully logs into DAM
- User locates “Dexter the Service Bot” near the top right corner
- User will ask Dexter any MCS-related question in the “Write question” box underneath it • Dexter will respond to the question in the top right corner, if appropriate
  - ◦If the question is not appropriate, Dexter will let the user know that the chat is being monitored and will refer the user to one of the Darden Blue Jacket members for a manual response to his or her question

Expected results -

- User is on the DAM website
- User successfully logs in
- User successfully locates “Dexter the Service Bot” box near the top right corner • User is able to type a question to Dexter in the box underneath it called “Write question” • User is able to click on “Ask” button
- Dexter is able to respond to the question if it is appropriate

Test case - Submission archive

Objective -

Ensure that users will have access to submitted assignments from previous classes.

Description -

Having access to previously submitted work from classes can help save the time, especially if the user is an MCS major.

Progress -

• User logs in

• User locates the “submission archive” link in the navigation bar

on the left • User selects the “submission archive” link

• Link sends the user to the submission archive

• User enters a course code or name in the search bar

• Any files associated with the user’s entry will be displayed (If any)

• User locates and selects the “download file” option

• File gets downloaded

Expected results -

• User is on the DAM website

• User successfully logs in

• User locates and selects the “submission archive” link in the navigation bar on the left within seconds

• Link successfully sends the user to the submission archive

• User successfully locates the search bar and enters the course

code or name • Files associated with the user’s entry successfully

displays onto the screen (If any) • User locates and successfully

selects the “download file” option within seconds • File successfully

downloads

Test case - Submission

Objective -

Ensure that files are being submitted properly and not exceeding the size limit given by the faculty member, and also the text submission functions correctly

Description -

Submission process should be as simplistic as possible. The MCS Moodle’s current way of submitting is pretty over complicated in my opinion.

Progress -

• User successfully logs in

• User locates the appropriate course for the assignment that needs to be submitted • User selects that course

• User locates the assignment that needs to be submitted

• User selects that assignment

• User is presented the assignment description and a submission box below it • The submission box displays “attach file” and “text submission” options

• User selects the appropriate option

• User submits the assignment

  ◦User can also clear the attached file or text if he or she made a mistake

Expected results -

- User is on the DAM website
- User successfully logs in
- User locates the appropriate course for the assignment that needs to be submitted within seconds • User successfully selects the course
- User successfully locates the assignment that needs to be submitted within seconds • User successfully selects the assignment
- User successfully locates the submission box below assignment description within seconds • User successfully selects the appropriate option to submit
- User successfully submits the assignment
    - Or, user successfully clears the attached file or text from the assignment

Test Case: Grade Calculation for Students

Objective: Allow users to insert a score they hope to get into an assignment and the total class average will be calculated.

Description: Once the user has successfully logged in and accessed their grades they will be able to insert a grade into an assignment. If the grade is above 100 or below 0 they will be issued a warning statement. There will also be a statement at the top of the screen once a score has been inserted stating that this is not the actual score. The professor will have to stay up to date on their grading scale for this to be suitable.

Results: The users will be able to input grades they think they will make into the system and the output will be their score.

Test Case: Understandable Navigation

Objective: Obtain information from users regarding how efficient the navigation system behaves. Not only clicks, but also if the layout is understandable.

Test: The user will start on the home screen and be given a path. For instance: Class1>Assignment1>Grades>DirectMessage. The user will be expected to move through the path until they reach the final destination. The users will be timed for the task. All of the paths given to the user should be completed in under 10 seconds.

We can gather useful information from the test because if the user spends a lot of time on the Assignment1 section of the problem then we know the grades section may need to be made more accessible.

Test case - Top taskbar

Objective -

Ensure all buttons located in the top taskbar function properly

Description -

Having a taskbar located at the top is crucial for a home page since 9/10 times they remain constant throughout user navigation. Therefore, it is important that anything located in the top taskbar works properly in order to maintain stability across the entire website.

Process 1 (Languages) -

• User successfully logs in

• User locates the top task bar of the home page

• User selects the “language” button

• The button will drop down languages that will allow the user to

select one of them • User selects a language

• The entire website gets translated into the language set by the user

Process 2 (User account) -

• User successfully logs in

• User locates the top task bar of the home page

• User selects the “person” icon

• User will be taken to their account info

• User will make any necessary changes

Process 3 (Messages) -

• User successfully logs in

• User locates the top task bar of the home page

• User selects the “mail” icon

• User will be taken to his or her inbox of messages

Process 4 (Notifications) -

• User successfully logs in

• User locates the top task bar of the home page

• User selects the “bell” icon

• User will be displayed notifications of recent updates, new posts, upcoming due dates, etc.

Expected results -

• User is on the DAM website

• User successfully logs in

• User is able to locate the top taskbar along with its
buttons within seconds • User is able to select any of the
buttons

• All of the buttons will function properly throughout the entire website

Test case - Upload Image

Objective -

Ensure that uploading images for the profile pic works properly

Description -

It’s important to visualize who each student looks like. This can help distinguish familiarity with one another and can also sort out any potential users who may not be associated with a certain class.

Progress -

• User logs in

• User locates the person icon at the top right corner of the taskbar

• User clicks on the person icon

• User is displayed his or her name, email, bio, phone number, and current profile pic • User selects the “upload image” button

• User will be brought to his or her files

• User will select the image they wish to upload as his or her profile pic

• The profile pic will be set to that image selected by the user

Expected results -

• User successfully logs into the DAM website

• User successfully locates the person icon at the top right corner of the taskbar within seconds • User successfully clicks on the person icon

• User successfully selects the “upload image” button

• User successfully accesses his or her files to choose from

• User successfully selects the image and uploads it as his or her profile pic • Profile pic successfully sets to that selected image by the user

Test case - Tutor Temple

Objective -

Ensure that the tutor temple feature works correctly

Description -

MCS students always seek help because of the diverse and ever-changing field. Sometimes, certain topics can become confusing which is why peer-to-peer tutoring is implemented in this feature in order to help those who need it. It is also, sometimes, more beneficial for students to engage with one another rather than have the traditional student-teacher learning.

Progress -

• User logs in

• User locates the navigation bar on the left along with the

“Tutor temple” link • User selects the “Tutor temple” link

• Link takes the user to Tutor Temple

• The option to “Create a topic” and “reply” or “delete” pre-existing topics will be visible to the user • User selects the “Create a topic” option

◦User enters in a topic/question he or she wishes to ask

◦User clicks “post”

◦Post becomes visible to other users

• User selects the “reply” option to a pre-existing topic

◦User is displayed a chat box with the option to attach a file as well

◦User enters his or her response and posts it onto the pre-existing topic

◦The response becomes visible to other users

• Users (Only visible to the poster and faculty members) selects the “delete” option ◦User is asked one more time if he or she wishes to delete the topic

◦User selects yes

‣ Topic gets deleted

◦User selects no

‣ Topic remains visible in Tutor Temple

Expected results -

• User is on the DAM website

• User successfully logs in

• User locates the navigation bar and successfully clicks on the “Tutor temple” link within seconds • Link successfully takes the user to Tutor Temple

• Options to “Create a topic” or “Reply” and “Delete” pre-existing topics will be visible to the user depending on status

◦Example: Only faculty members and the poster will see the option to delete pre-existing topics

• User will select whatever option he or she wants

• The option selected executes as intended

Security

Assets that need to be protected:

- User login info for both DAM and the email associated with it (Username and password) ○ Pretty much a downward spiral effect once the user login info has been accessed by the attacker/s

Possible attack vectors for the site:

- Pop-ups and ads
- Scam/Phishing email messages and attachments
- Data breach (Ex. Hacker acting as a professor who sends a link for students to click on)
- Brute force password cracker
    - If the username is unknown, a wordlist can be generated of possible usernames ○ Can run multiple passwords against the system to try to log in

Steps we can take to deal with these security issues:

- Constantly perform test cases to determine how secure DAM is
- Identify any scenarios where there can be any form of possible attacks
- Monitor anything that may not seem to have issues with security
    - Just because it’s not a target does not mean it’s secure
- Company-wide password requirements to strengthen passwords
    - Length must be over 8

- ○ Must have 1 special character
- ○ Must have a capital letter
- ○ Must have 1 number
- ○ Password cannot be the same as the username

Security - User Account

The User Account page gives the user the option to populate fields regarding their personal information. By default, user profiles are visible to other users of the site. This way students can view relevant information about fellow students or instructors such as email addresses and what classes they are enrolled in. The visibility of a user’s profile can be set to “Private” to hide this information from others if the user wishes. All of the information entered on this page is stored on the server in a hashed format legible only to the system itself.

Users may opt to change their associated email, phone number, or 2FA app connections on this page. If any changes are made, a notification is pushed to the other unmodified contacts notifying the user that a change has taken place. If the user does not recognize this activity they can then have the account locked.

Security - Login Page

The login page session is secured with encryption to ensure that all transactions occurring on the page are secure and visible only to the server and the user. DAM comes with a configurable account-lockout threshold that can disable accounts that have too many unsuccessful login attempts to mitigate malicious actors from gaining access to accounts. On a successful login, the user may be prompted for some form of

Two Factor Authentication (2FA) depending on where they log in geographically and their time from last login. 2FA apps such as Duo Security, Google Authenticator, or Authy may be used to fulfill the 2FA requirement. 2FA is set up in the user account page, and is implemented to further lock down the system from any malicious actors trying to gain access.

The Create Account dialog opens a new page where new users can create their DAM account. Administrators can configure requirements for the username and password complexity and length requirements that a user must satisfy before creating an account; by default the minimum password length is 12 characters. Additionally, the email used to sign up for an account can be limited to only those in the institution’s domain so that only verified, admitted students and faculty can gain access to the system. Any and all sensitive information relative to the login process is stored in the server in a hashed format so that in the event of a breach of the system the information is useless to any bad actors.

The Login Recovery option opens the DAM Login Recovery page. The page prompts the user for the email used to create their account, and then gives them the option to confirm the login recovery through an email or text verification through their registered email or phone number respectively by sending a 6 digit code that is to be entered. This verification code is only valid for 5 minutes to prevent brute forcing of the page.

Security - File Archive

The File Archive can store any of a student’s prior works for future access. For an additional layer of protection the user is once again prompted for a 2FA push on their app of choice to prevent someone from gaining access to a user’s saved files in case the user walks away from the device they are logged into without logging out.

The archive stores all user submitted files in an encrypted format to maximize security, using client-side encryption to ensure that no data can be intercepted in the transaction between client and server. Administrators can configure the size of individual archives for each user so as to not take up excess space on the server. Likewise, the size of individual files can also be controlled to prevent malicious or oversized files from being uploaded, the likes of which could hamper system performance. File types can also be controlled so that potential exploits utilizing specific file format can be mitigated.

Backup schema for files in the archive can be configured by administrators to reduce the risk of data loss in the event of hardware failure. Access logs are also maintained for the file archive so that any suspicious activity regarding a file can be monitored to ensure data integrity.

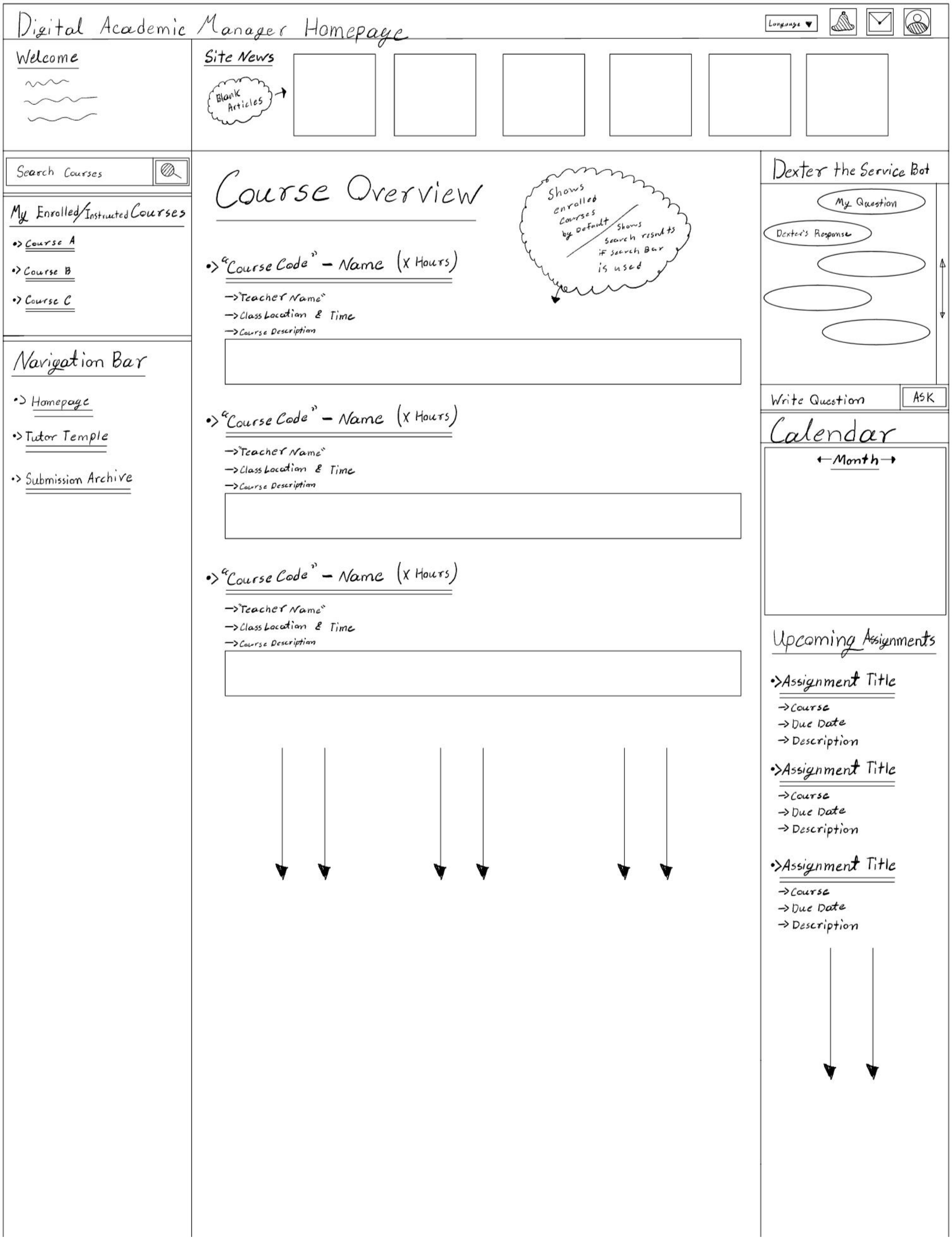
Security - Class Chat

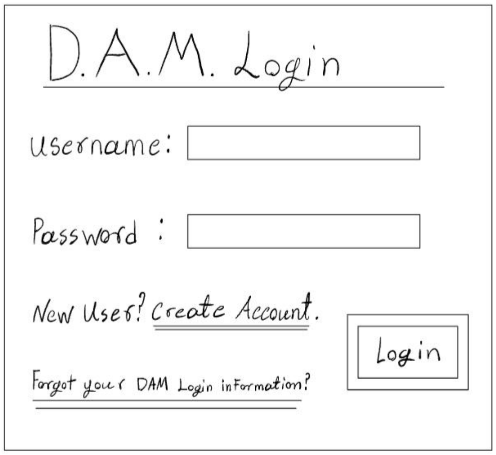
The chat room feature allows students to communicate about classwork in a secured, monitored manner. Any messages sent to the chat room are encrypted to prevent eavesdropping and maintain confidentiality. Only students enrolled in a course may access that course’s chat room; this helps to prevent any cheating from occurring as well as to help mitigate “leaks” of chatrooms to outside parties. When a user chats their name as it appears on their profile is used as their username, and all messages that are sent are also timestamped for timeliness assurance. When sending a chat message users have the option to attach a file to their message. The file types, sizes, and number of files can be controlled through administrative templates to prevent any sort of exploit from happening on opening of certain files. Links can also be controlled in the chatroom, with administrators setting rules on what links are allowed or outright banning them altogether. This way links to malicious sites or downloads will not show up for other users to click on.

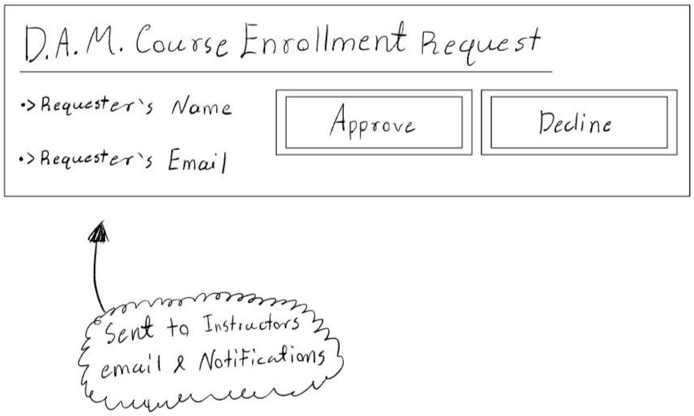
Logs of all chat activities are stored locally so that instructors and administrators can view them if academic violations are suspected to have happened. All logs are encrypted and thus only visible to authorized staff and administrators that want to access them.

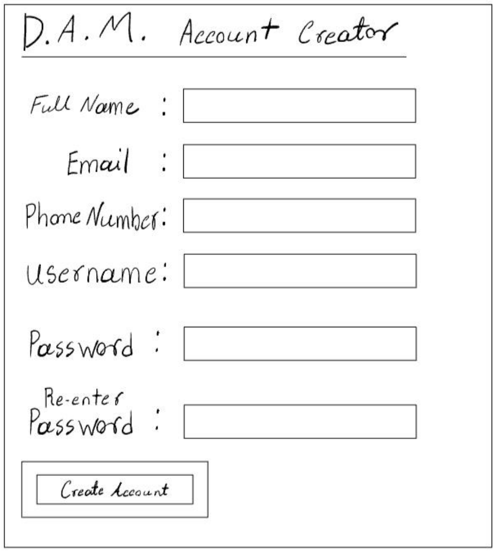
Screen Layouts

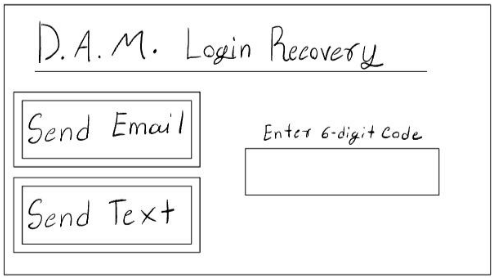
The following pages are illustrative diagrams of the screen layout concepts.

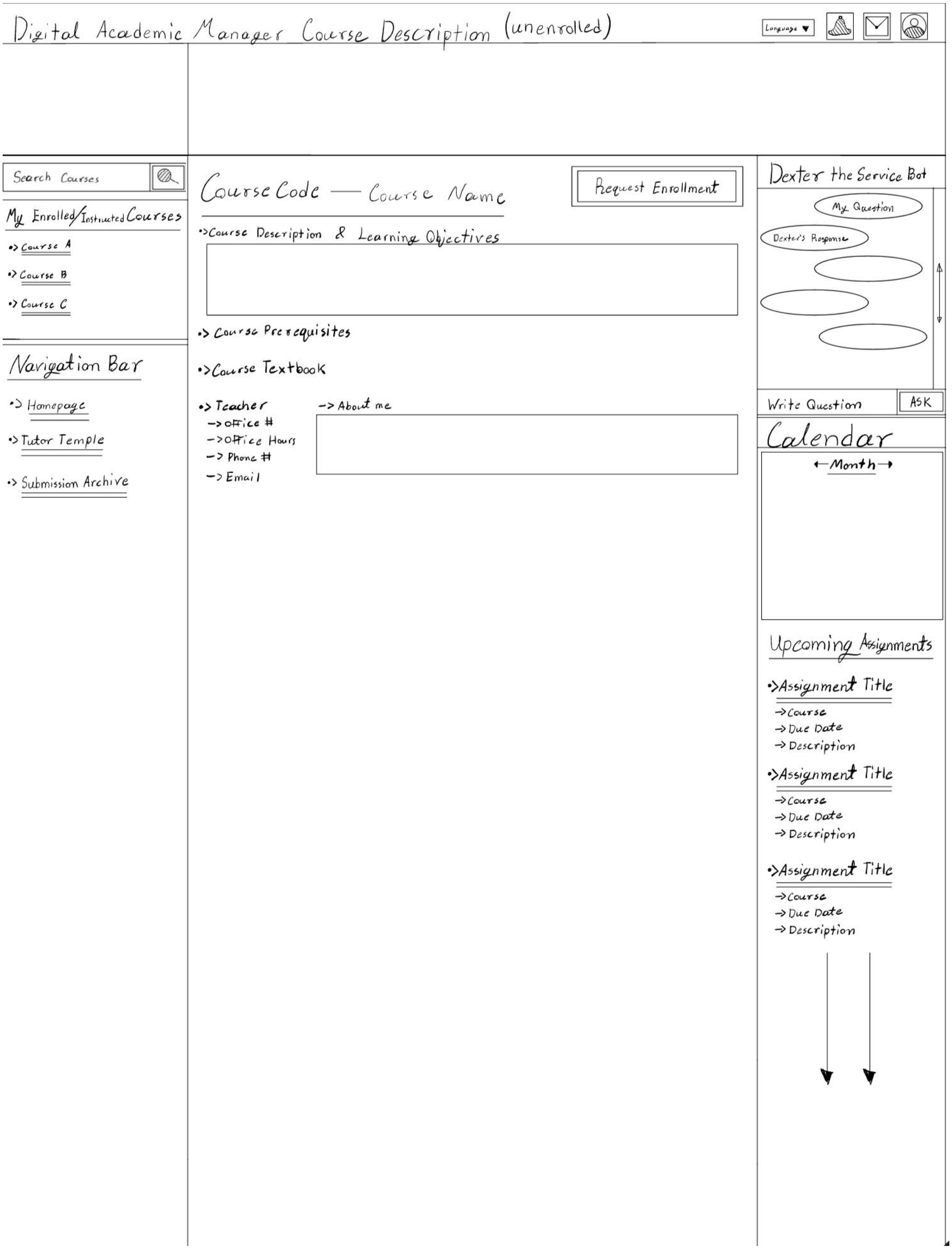


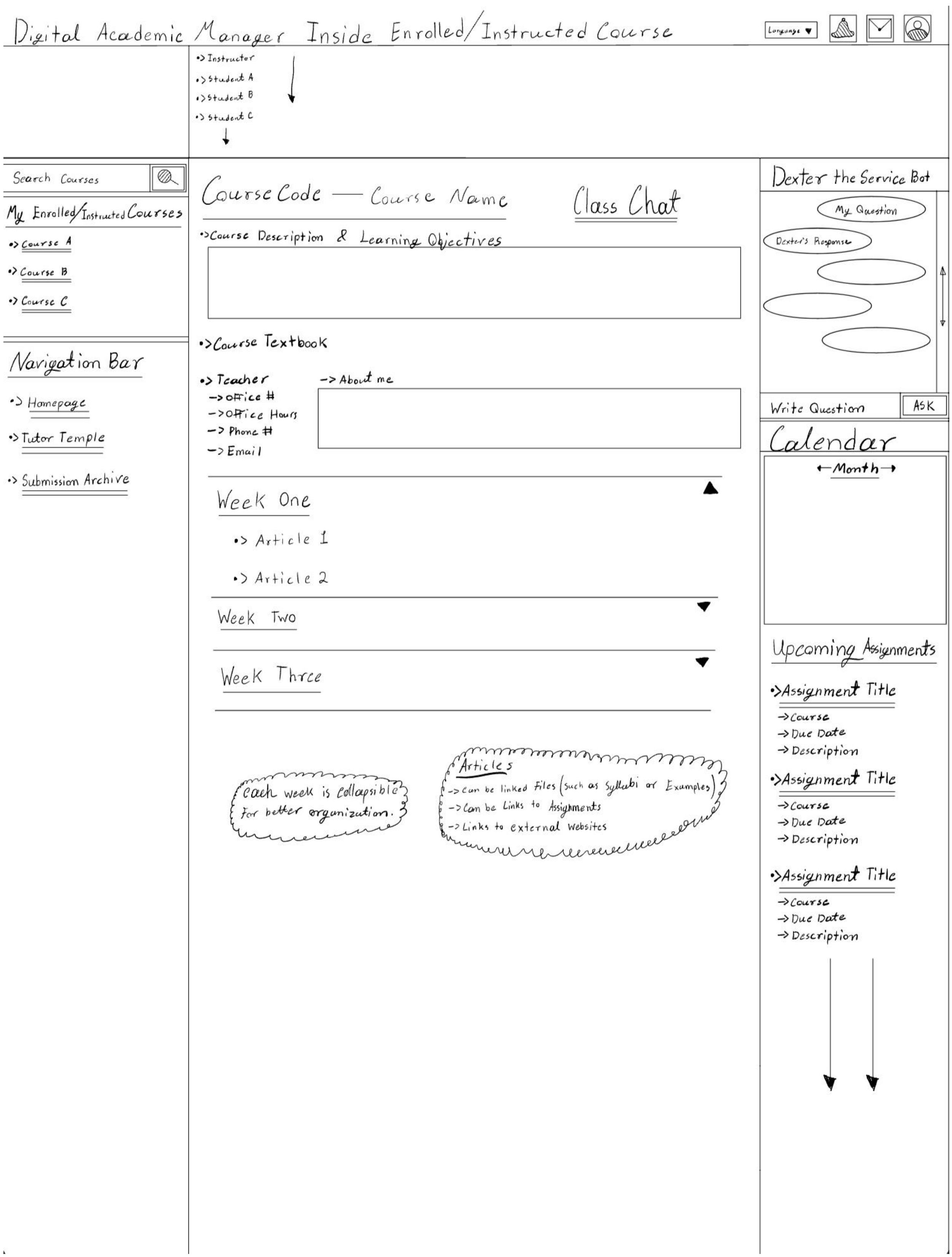

D.A.M. Login
Username:
Password :
New User? Create Account.
Login
Forgot your DAM Login information?


D.A.M. Course Enrollment Request
•> Requester's Name
•> Requester's Email
Approve
Decline
Sent to Instructors email & Notifications


D.A.M. Account Creator
Full Name :
Email :
Phone Number:
Username:
Password :
Re-enter Password :
Create Account

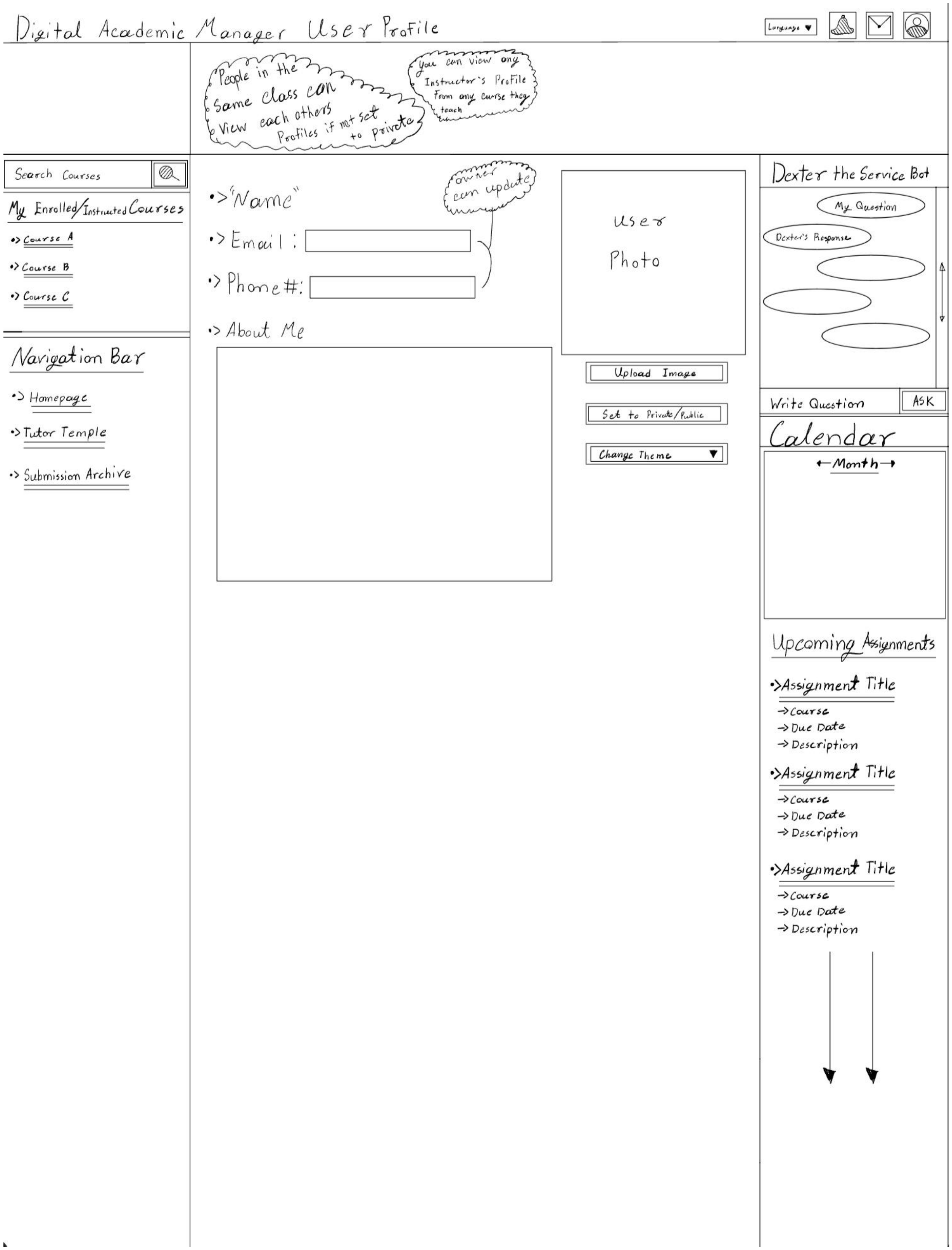

D.A.M. Login Recovery
Send Email
Send Text
Enter 6-digit Code

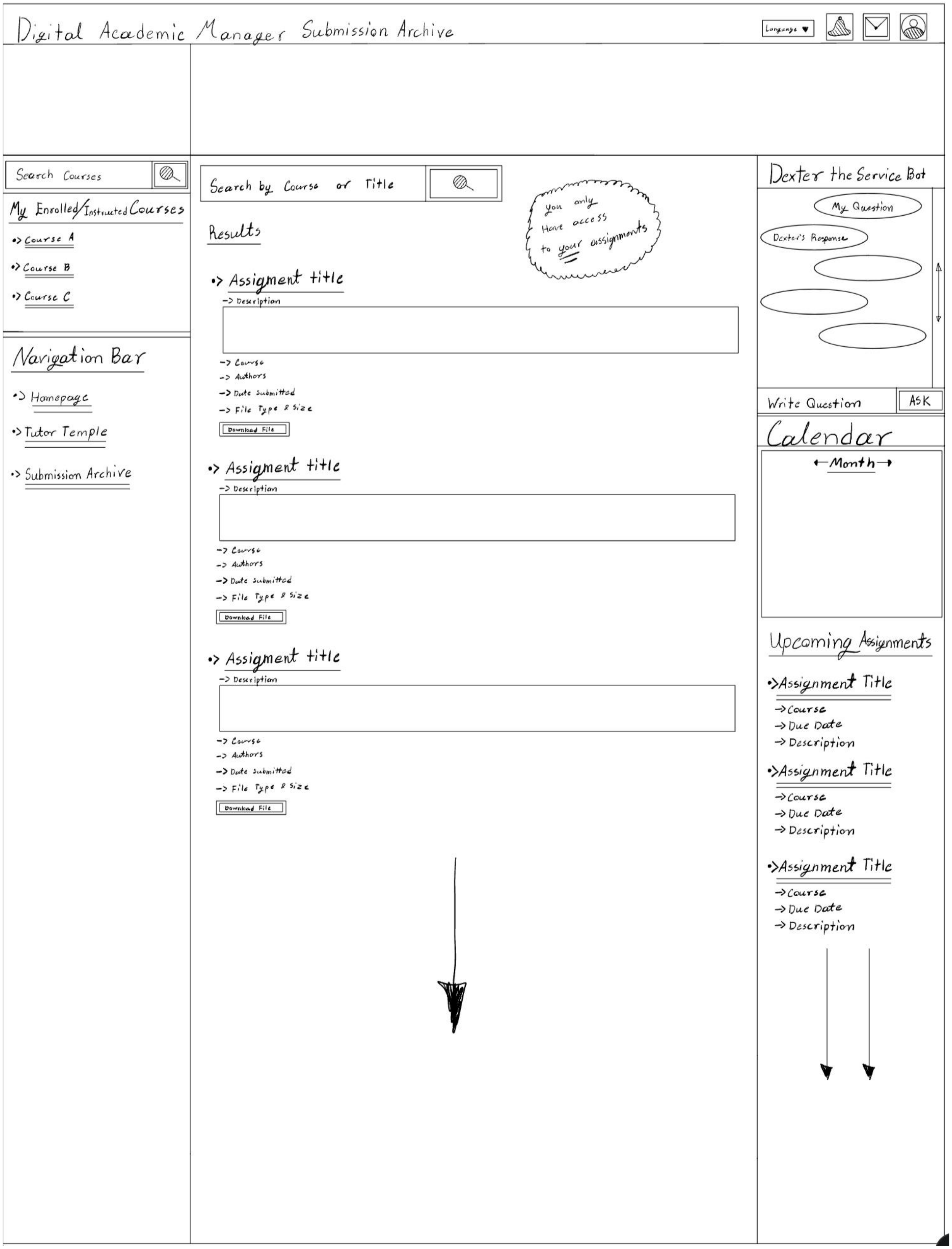

Digital Academic Manager Course Description (unenrolled)
Language ▼
Search Courses
My Enrolled/Instructed Courses
•> Course A
•> Course B
•> Course C
Navigation Bar
•> Homepage
•> Tutor Temple
•> Submission Archive
Course Code — Course Name
Request Enrollment
•>Course Description & Learning Objectives
•> Course Prerequisites
•>Course Textbook
•> Teacher
-> office #
->Office Hours
-> Phone #
-> Email
-> About me
Dexter the Service Bot
My Question
Dexter's Response
Write Question
ASK
Calendar
←Month→
Upcoming Assignments
•>Assignment Title
->Course
->Due Date
->Description
•>Assignment Title
->Course
->Due Date
->Description
•>Assignment Title
->Course
->Due Date
->Description

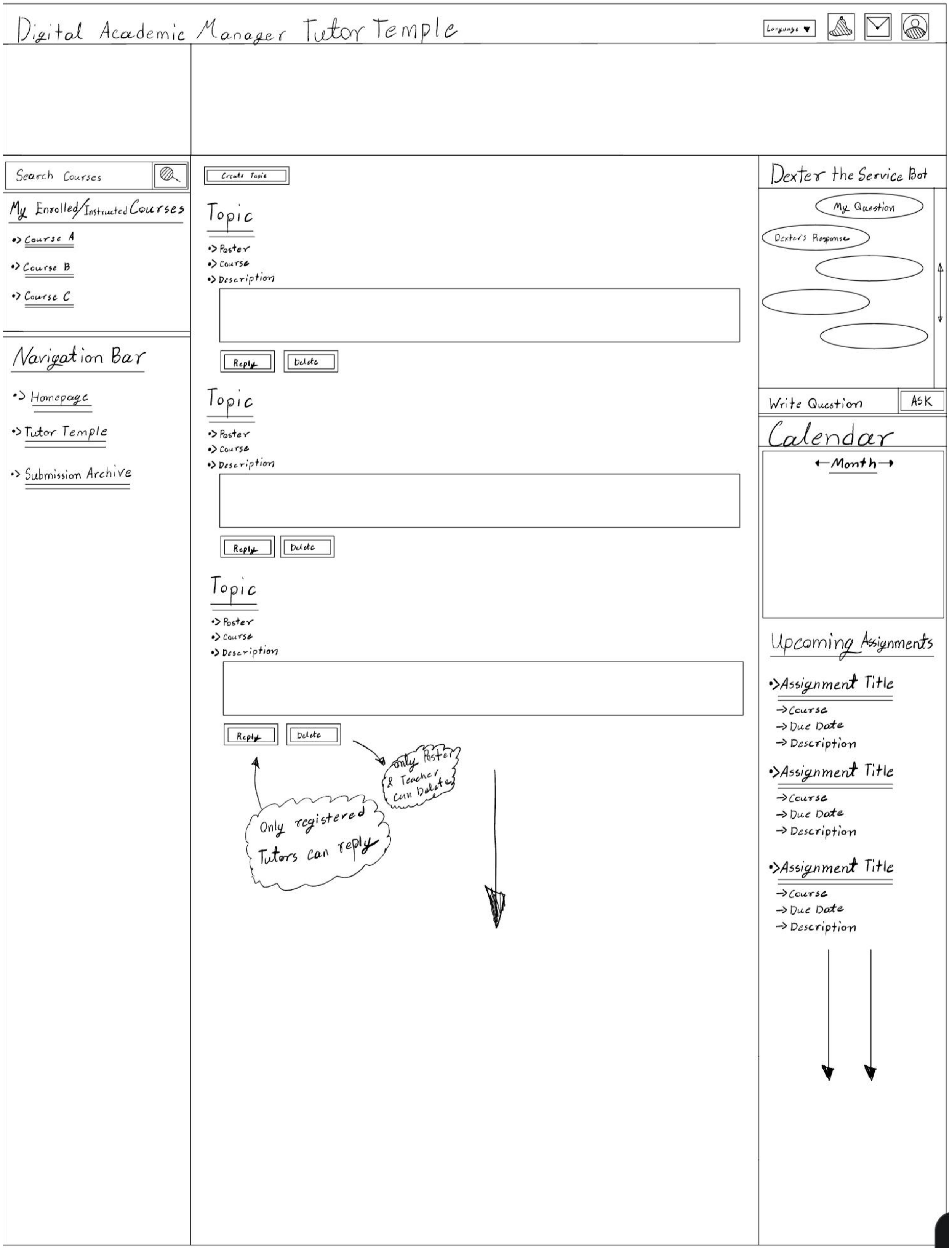


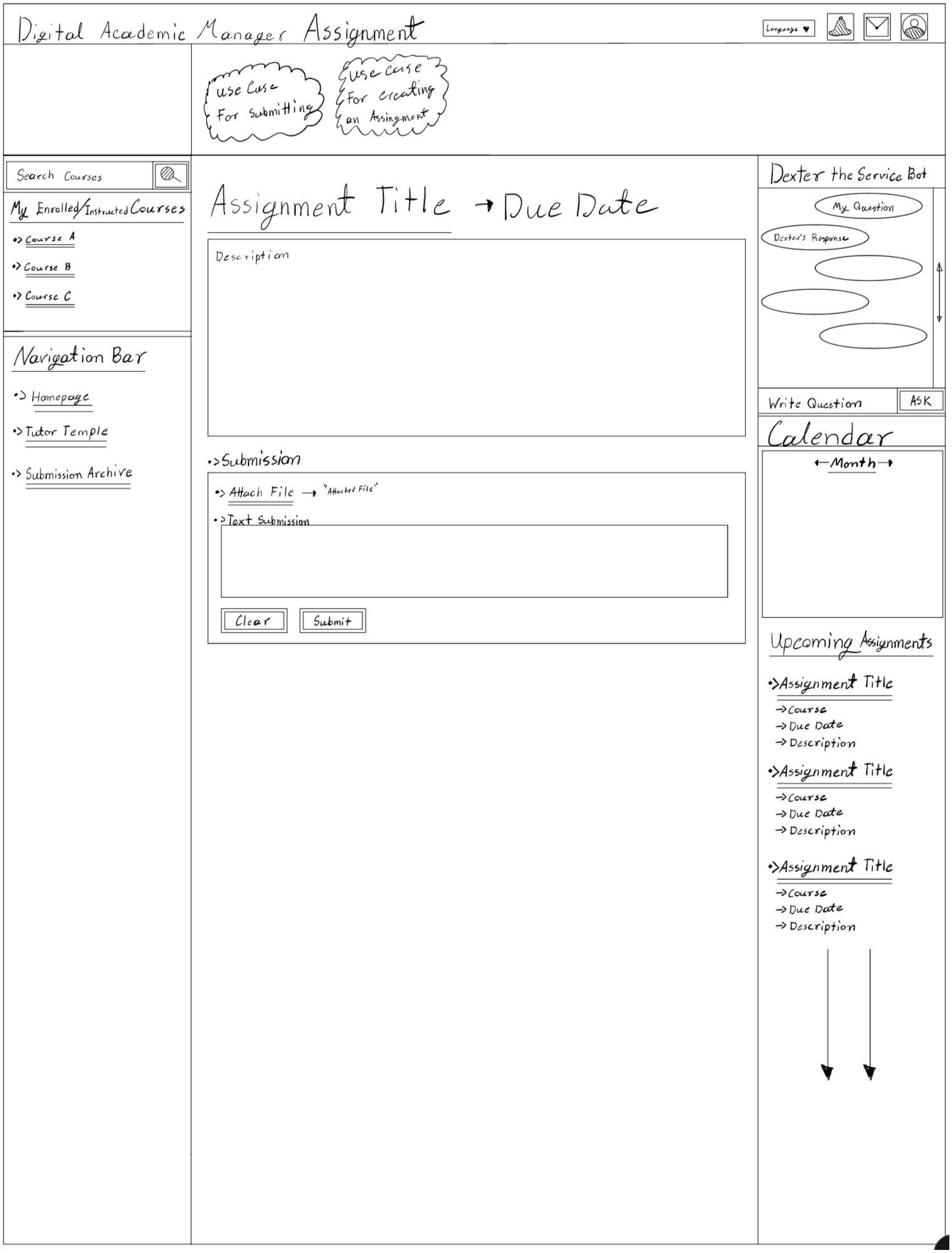
Each week is collapsible for better organization.

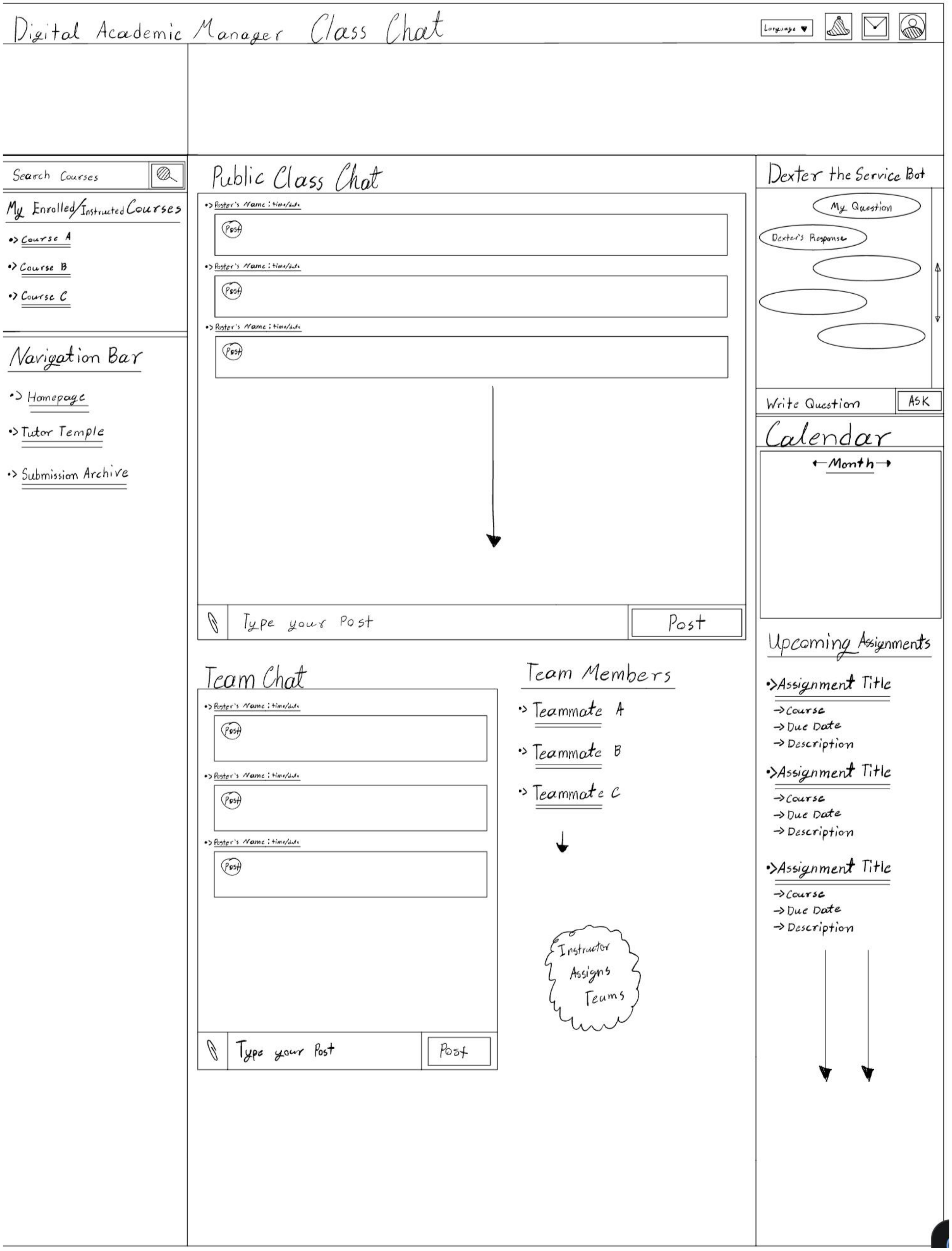
Articles
-> Can be linked files (such as Syllabi or Examples)
-> Can be Links to Assignments
-> Links to external websites


Digital Academic Manager User Profile
Language ▼
"People in the same class can view each others Profiles if not set to private
you can view any Instructor`s Profile From any course they teach
Search Courses
My Enrolled/Instructed Courses
•> Course A
•> Course B
•> Course C
Navigation Bar
•> Homepage
•> Tutor Temple
•> Submission Archive
•> "Name"
owner can update
•> Email :
•> Phone #:
•> About Me
User Photo
Upload Image
Set to Private/Public
Change Theme ▼
Dexter the Service Bot
My Question
Dexter's Response
Write Question
ASK
Calendar
← Month →
Upcoming Assignments
•>Assignment Title
→ Course
→ Due Date
→ Description
•>Assignment Title
→ Course
→ Due Date
→ Description
•>Assignment Title
→ Course
→ Due Date
→ Description


Digital Academic Manager Submission Archive
Language ▼
Search Courses
My Enrolled/Instructed Courses
•> Course A
•> Course B
•> Course C
Navigation Bar
•> Homepage
•> Tutor Temple
•> Submission Archive
Search by Course or Title
Results
you only Have access to your assignments
•> Assigment title
-> Description
-> Course
-> Authors
-> Date submitted
-> File Type & Size
Download File
•> Assigment title
-> Description
-> Course
-> Authors
-> Date submitted
-> File Type & Size
Download File
•> Assigment title
-> Description
-> Course
-> Authors
-> Date submitted
-> File Type & Size
Download File
Dexter the Service Bot
My Question
Dexter's Response
Write Question
ASK
Calendar
←Month→
Upcoming Assignments
•>Assignment Title
→Course
→Due Date
→Description
•>Assignment Title
→Course
→Due Date
→Description
•>Assignment Title
→Course
→Due Date
→Description


Digital Academic Manager Tutor Temple
Language ▼
Search Courses
My Enrolled/Instructed Courses
•> Course A
•> Course B
•> Course C
Navigation Bar
•> Homepage
•> Tutor Temple
•> Submission Archive
Create Topic
Topic
•> Poster
•> Course
•> Description
Reply
Delete
Topic
•> Poster
•> Course
•> Description
Reply
Delete
Topic
•> Poster
•> Course
•> Description
Reply
Delete
Only Poster & Teacher can Delete
Only registered Tutors can reply
Dexter the Service Bot
My Question
Dexter's Response
Write Question
ASK
Calendar
←Month→
Upcoming Assignments
•>Assignment Title
→Course
→Due Date
→Description
•>Assignment Title
→Course
→Due Date
→Description
•>Assignment Title
→Course
→Due Date
→Description


Digital Academic Manager Assignment
Language ▼
use Case For submitting
use case For creating an Assingment
Search Courses
My Enrolled/Instructed Courses
•> Course A
•> Course B
•> Course C
Navigation Bar
•> Homepage
•> Tutor Temple
•> Submission Archive
Assignment Title → Due Date
Description
•> Submission
•> Attach File → "Attached File"
•> Text Submission
Clear
Submit
Dexter the Service Bot
My Question
Dexter's Response
Write Question
ASK
Calendar
← Month →
Upcoming Assignments
•>Assignment Title
→ Course
→ Due Date
→ Description
•>Assignment Title
→ Course
→ Due Date
→ Description
•>Assignment Title
→ Course
→ Due Date
→ Description


Digital Academic Manager Class Chat
Language ▼
Search Courses
My Enrolled/Instructed Courses
•> Course A
•> Course B
•> Course C
Navigation Bar
•> Homepage
•> Tutor Temple
•> Submission Archive
Public Class Chat
•> Poster's Name : time/date
Post
•> Poster's Name : time/date
Post
•> Poster's Name : time/date
Post
Type your Post
Post
Team Chat
•> Poster's Name : time/date
Post
•> Poster's Name : time/date
Post
•> Poster's Name : time/date
Post
Type your Post
Post
Team Members
•> Teammate A
•> Teammate B
•> Teammate C
Instructor Assigns Teams
Dexter the Service Bot
My Question
Dexter's Response
Write Question
ASK
Calendar
←Month→
Upcoming Assignments
•>Assignment Title
→Course
→Due Date
→Description
•>Assignment Title
→Course
→Due Date
→Description
•>Assignment Title
→Course
→Due Date
→Description