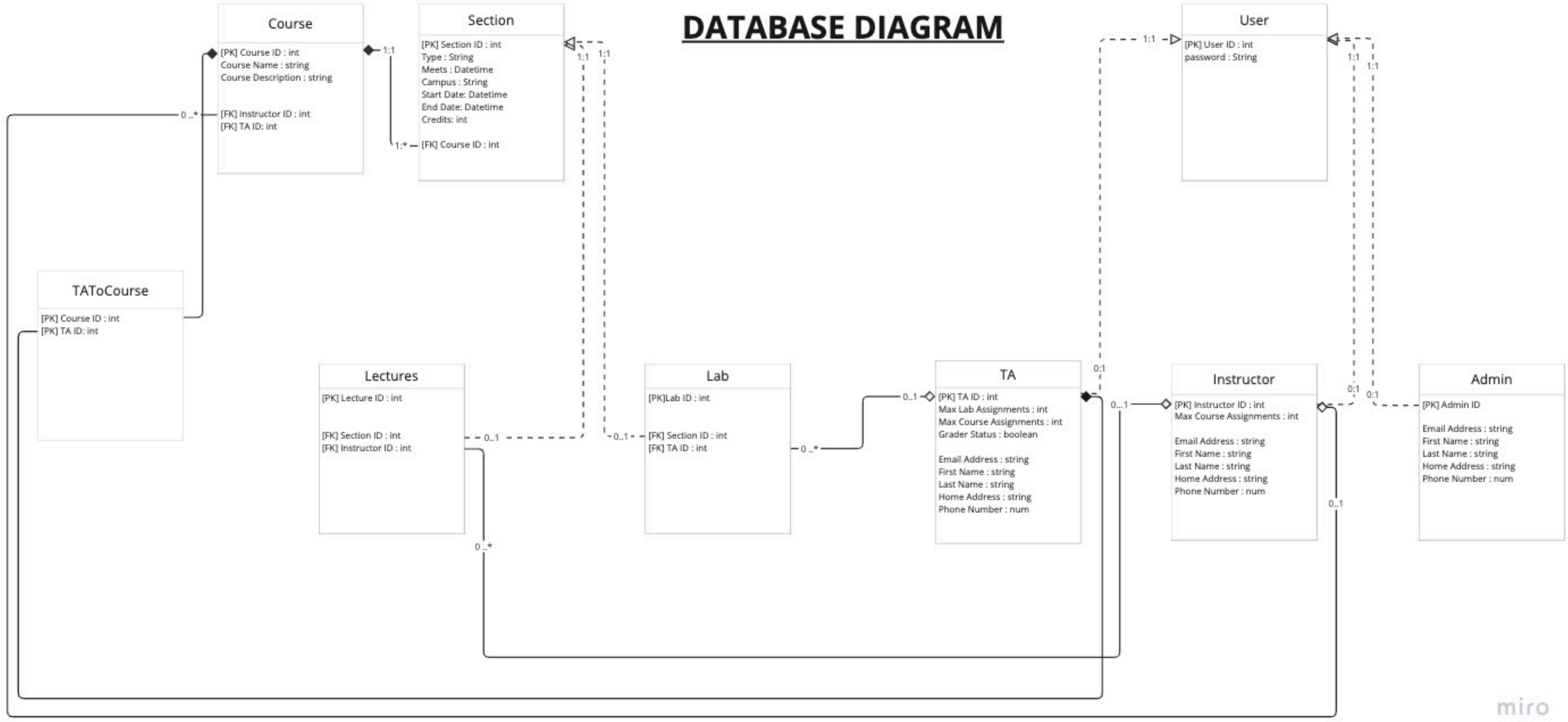
DATABASE DIAGRAM



Course

[PK] Course ID : int
Course Name : string
Course Description : string

[FK] Instructor ID : int
[FK] TA ID: int

Section

[PK] Section ID : int
Type : String
Meets : Datetime
Campus : String
Start Date: Datetime
End Date: Datetime
Credits: int

[FK] Course ID : int

User

[PK] User ID : int
password : String

TAToCourse

[PK] Course ID : int
[PK] TA ID: int

Lectures

[PK] Lecture ID : int

[FK] Section ID : int
[FK] Instructor ID : int

Lab

[PK]Lab ID : int

[FK] Section ID : int
[FK] TA ID : int

TA

[PK] TA ID : int
Max Lab Assignments : int
Max Course Assignments : int
Grader Status : boolean

Email Address : string
First Name : string
Last Name : string
Home Address : string
Phone Number : num

Instructor

[PK] Instructor ID : int
Max Course Assignments : int

Email Address : string
First Name : string
Last Name : string
Home Address : string
Phone Number : num

Admin

[PK] Admin ID

Email Address : string
First Name : string
Last Name : string
Home Address : string
Phone Number : num

SOFTWARE DIAGRAM

<<interface>>
User

+editMyContactInfo(Dictionary)
+getContactInfo() : Dictionary

+login(int, String)
+getID() : int
+getPassword(): int

+viewAllTAAsgmt()
+emailUser(String, String)

Web Page Class: View:Public Contact Info

Web Page Class: Login

Web Page Class: Notify Users

Web Page Class: Assign-TA Course

Web Page Class: Assign-Instructor Course

Web Page Class: User Access

Web Page Class: Delete-Account

Web Page Class: Edit-Account Info/Assignments

Web Page Class: Create-Account

Admin

-adminID : int
-password: String

-firstName : String
-lastName : String
-emailAddr : String
-homeAddr : String
-phoneNum : num

+createCourse(Dictionary)
+createInstructor(Dictionary)
+createTA(Dictionary)

+removeCourse(Course)
+removeAccount(User)

+editAccount(User)

+courseInstrAsgmt(Instructor)
+courseTAAsgmt(TA)
+labTAAsgmt(TA)

+viewAllCrseAsgmt()

Web Page Class: View-TA Assignments

TA

-taID : int
-password: String
-gradeStatus : boolean
-maxNumLabs : int
-maxNumCourse : int

-courses : List(Course)
-labs : List(lab)

-firstName : String
-lastName : String
-emailAddr : String
-homeAddr : String
-phoneNum : num

+hasMaxCrseAsgmts() : boolean
+hasMaxLabAsgmts() : boolean

+assignTACourse(Course)
+assignTALab(Lab)

+getTACrseAsgmts() : list(course)
+getTALabAsgmts() : list(lab)
+getGraderStatus(): boolean

Web Page Class: View-MY Course Assignments

Instructor

-instrID : int
-password: String
-maxNumCourses : int

-courses : List(Course)
-lectures : List(Lecture)

-firstName : String
-lastName : String
-emailAddr : String
-homeAddr : String
-phoneNum : num

+hasMaxCrseAsgmts: boolean

+assignInstrCourse(Course)
+assignMyTA(TA)

+notifyTAs()

+viewAllCrseAsgmt()
+viewInstrTAAsgmt()

+getInstrCrseAsgmts() : list(course)

Web Page Class: View-Course Assignments

Web Page Class: Data Access-Course

Web Page Class: Create-Course

Course

- courseID : int
- courseName : String
- courseDescr : String
- semester : String
- modality : String
- wklyCrseSched : datetime

- instructor : Instructor
- tas : List(TA)
- sec : List(Section)

+addInstructor(Instructor)
+addTA(TA)

+removeAssignment(User)
+removeCourse()

+viewAsgmtsForCrse()
+viewSectionsForCrse()
+viewCrseInfo()

<<interface>>
Section

+getID() : int
+getParentCourse(): course

Web Page Class: Assign-TA a Lab

Web Page Class: Assign-"My TA" a Lab

``

Lab

- labID : int
- ta : TA
- hostCourse : Course

-meets : Datetime
-campus : String
-startDate: Datetime
-endDate: Datetime
-credits: int

+getLabTAAsgmt() : TA

+addTA(TA) : boolean

+removeTA() : boolean

Lecture

- lectureID:
- instructor: Instructor
- hostCourse : Course

-meets : Datetime
-campus : String
-startDate: Datetime
-endDate: Datetime
-credits: int

+getLecInstrAsgmt() : Instructor

+addInstructor(Instructor) : boolean

+removeInstructor() : boolean

METHOD DESCRIPTIONS

BEGIN USER

User.editMyContactInfo(contactDict)

Preconditions: The provided dictionary must contain all of the necessary fields (first/last name, email address, home address, phone number) and any assignments to the phone number key inside the dictionary must be 10 numbers.
Postconditions: If valid changes, the user's contact information is set to the input fields, all assignments to this user will be updated, the GUI will display all the changes. If missing or invalid inputs, it will display an error message.
Side-Effects: ---
contactDict: input – dictionary contains keys named after all of the user's fields who's values are used to update the user's fields.

User.getContactInfo()

Preconditions: all fields for a user have been correctly validated in the constructor.
Postconditions: no change to the host object, returns a dictionary: keys named after the user's fields and values associated with each field.
Side-Effects: ---

User.login(id, password)

Preconditions: given an id that exists in the system and the associated password for that ID
Postconditions: opens application for the role associated with the given ID. Error message displayed if invalid username or password.
Side-Effects: ---
Id: input – an integer to uniquely identify each user Password: input – a string to unlock the application for each unique ID.

User.getID()

Preconditions: ID field for a user have been correctly validated in the constructor, every user contains a unique ID from every other user.
Postconditions: No change to host object, returns an integer representing the User's unique identifier.
Side-Effects: ---

User.getPassword()

Preconditions: password fields for a user have been correctly validated in the constructor, the password is more than 4 characters long (minimum requirement).
Postconditions: No change to host object, returns a string representing the password for the User's login.
Side-Effects: ---

User.viewAllTAAsgmt()

Preconditions: There must be at least one active TA.
Postconditions: no change to the host object or the respective sections, it displays all TAs and their assignments. If no TA’s, displays a messages indicating there is no TAs.
Side-Effects: ---

User.emailUser(recipEmail, body)

Preconditions: the recipient email must be provided and must be an existing user, the email’s body must not be null.
Postconditions: the body is sent to the recipient’s email address, and the “sending” user will also be sent the email in their “sent” tab.
Side-Effects: ---
recipEmail: input – String represents the user receiving the email, will be used to validate whether the email is sent to an existing user. body: input – String representing the body of an email.

BEGIN ADMIN

Admin.createCourse(courseInfo)

Preconditions: The course to be created cannot contain an ID of an already existing course/section. All fields required for a course must be provided in the parameter.
Postconditions: Course with empty assignments and 1 section (a lecture) is created, it is displayed in the GUI and users can now be assigned to it. If provided duplicate ID or missing inputs, displays error message.
Side-Effects: ---
courseInfo: input – dictionary contains keys named after all of the course’s fields who’s values are used to enter the course’s fields.

Admin.createInstructor(instrInfo)

Preconditions: The instructor to be created cannot contain an ID of an already existing User. All fields required for a instructor must be provided in the parameter.
Postconditions: Instructor with empty assignments is created, the instructor can now be assigned to courses. If provided duplicate ID or missing inputs, displays error message.
Side-Effects: ---
instrInfo: input – dictionary contains keys named after all of the instructor’s fields who’s values are used to enter the instructor’s fields.

Admin.createTA(TAInfo)

Preconditions: The TA to be created cannot contain an ID of an already existing User. All fields required for a TA must be provided in the parameter.

Postconditions: TA with empty assignments is created, the TA can now be assigned to courses. If provided duplicate ID or missing inputs, displays error message.
Side-Effects: ---
TAInfo: input – dictionary contains keys named after all of the TA fields who's values are used to enter the TA's fields.

Admin.removeCourse(activeCourse)

Preconditions: The course trying to remove must exist.
Postconditions: the course will be removed from the schedule of classes GUI, and no user can be assigned to it. If provided non-existent course, displays error message.
Side-Effects: The respective sections and assignments to this course will also be removed.
activeCourse: in out - This course will be used to in a search to traverse the courses table to remove. The parameter's state will be mutated: all user assignments/sections are removed.

Admin.removeAccount(activeUser)

Preconditions: The account trying to remove must exist.
Postconditions: the account will be removed from the schedule of classes GUI, and no course/section can link to this user. If provided non-existent account, displays error message.
Side-Effects: The respective course/section assignments to this user will also be removed.
activeUser: in-out - This user will be used to in a search to traverse the user table to remove. The parameter's state will be mutated: all course/section assignments are removed.

Admin.editAccount(activeUser)

Preconditions: the account to edit must exist in the system and the user's input meets formatting requirements for contact information
Postconditions: valid changes to account's state are saved, all GUI elements with this account are updated, course's/sections also reflect any changes. If provided non-existent user, displays error message.
Side-Effects: ---
activeUser: output – This account's information, state, will be modified according to the user's input.

Admin.courseInstrAsgmt(activeInstr)

Preconditions: The instructor must exist in the system, the instructor has not exceeded their maximum course assignments, and the course trying to assign doesn't already have an instructor
Postconditions: The instructor will receive the course assignment and the course will receive the instructor assignment. If max capacity instructor/course or nonexistent user provided, displays error message.
Side-Effects: ---
activeInstr: in – out: the instructor is used as an input for the course's instructor field and the instructor itself will be assigned to a course.

Admin.courseTAAsgmt(activeTA, graderStatus)

Preconditions: The TA must exist in the system and the TA has not exceeded their maximum course assignments.
Postconditions: The TA is assigned to the course, the course adds this TA as an assigned User, and the TA is given the declared grader status. If nonexistent user provided or TA is at maximum course assignments, the error message displayed.
Side-Effects: ---
activeTA: in out - the TA is used as an input for the course's List(TA) field, and the TA itself will be assigned to the course. graderStatus: input – applies to the TA object, specifies whether this TA is meant for a lab or just a course.

Admin.labTAAsgmt(activeTA)

Preconditions: The TA must exist in the system, the TA has not exceeded their maximum lab assignments, the TA hasn't been assigned to grader status, and the lab doesn't already have a TA.
Postconditions: The TA will receive the lab assignment and the lab will receive the TA assignment. If nonexistent TA, full Lab, full TA lab assignments, or grader status is true, then error message displayed.
Side-Effects: ---
activeTA: in – out: the TA is used as an input for the lab's TA field and the TA itself will be assigned to the lab.

Admin.viewAllCrseAsgmt()

Preconditions: There must be at least one active course.
Postconditions: no change to the host object or the respective course, displays all courses and their User assignments. If no active courses in the system, it displays an error message.
Side-Effects: ---

BEGIN TA

TA.hasMaxCrseAsgmts()

Preconditions: Maximum course assignment field has been instantiated in constructor, it is non-negative real number, and any assignments to this TA must be reflected accurately in the TA's state
Postconditions: returns whether the user has reached it's maximum course assignments.
Side-Effects: ---

TA.hasMaxLabAsgmts()

Preconditions: Maximum lab assignment field has been instantiated in constructor, it is non-negative real number, and any assignments to this TA must be reflected accurately in the TA's state.
Postconditions: returns whether the user has reached it's maximum lab assignments.
Side-Effects: ---

TA.assignTACourse(activeCourse)

Preconditions: The "to be assigned" course must exist in the system, The TA has not exceeded their maximum course assignments.
Postconditions: Modifies the TA's state: it is assigned to the course, and the course adds this TA to it's list of TAs.
Side-Effects: ---
activeCourse: in out – the course is used as an input for the TA's list(courses) and the course's state will be mutated to add the TA to it's assigned Users.

TA.assignTALab(activeLab)

Preconditions: The "to be assigned" lab must be active, the TA hasn't been assigned to grader status, and the lab doesn't already have a TA .
Postconditions: Modifies the TA's state: it will receive the lab assignment and the lab will receive the TA assignment.
Side-Effects: ---
activeLab: in – out: the lab is used as an input for the TA's list(lab) and the lab's state will be mutated to assign the TA.

TA.viewAsgmtsForUser()

Preconditions: The course/section assignments to this user must match the user's assignments to the course/section.
Postconditions: no change to the host object or the respective course, displays all of the user's assignments to Sections or Courses
Side-Effects: ---

TA.getTACrseAsgmts()

Preconditions: The TA must contain at least one assignment to a course and the TA’s assignments to courses reflect any assignments from courses to this TA.
Postconditions: Returns a list of the Courses assigned to this TA. Displays if no current assignments
Side-Effects: ---

TA.getTALabAsgmts()

Preconditions: The TA must contain at least one assignment to a lab and the TA’s assignments to labs reflect any assignments from labs to this TA.
Postconditions: Returns a list of the labs assigned to this TA. Displays if no current assignments
Side-Effects: ---

TA.getGraderStatus()

Preconditions: Unique TA has been instantiated.
Postconditions: Returns whether or not the TA has been assigned a grader status.
Side-Effects: ---

BEGIN INSTRUCTOR

Instructor.hasMaxCrseAsgmts()

Preconditions: Maximum course assignment field has been instantiated in constructor, it is non-negative real number, and any assignments to this Instructor must be reflected accurately in the Instructor state
Postconditions: returns whether the user has reached it’s maximum course assignments.
Side-Effects: ---

Instructor.assignInstrCourse(activeCourse)

Preconditions: Given a course that exists in the system, the course can’t already have an instructor, and the instructor’s course assignments are less than the maximum capacity.
Postconditions: the instructor is assigned the course and the course is assigned the instructor. If non-existent course or maximum capacity instructor/course, then error message displayed.
Side-Effects: ---

activeCourse: in out – the instructor is assigned to the given course and the given course itself is modified to include the new instructor assignment.

Instructor.assignMyTA(activeTA)

Preconditions: TA must exist in the system, TA must be assigned to one of my courses, TA can't have grader status, the lab to assign the TA to can't already have a TA.
Postconditions: Modifies the TA's state: it is assigned to a lab, and the lab is assigned to this TA. Displays error message if TA isn't assigned to this course, TA has grader status, or the lab is already full.
Side-Effects: ---
activeTA: input output – a TA that is assigned to one of the instructor's courses, it will be used to modify one of the instructor's course's lab's assignments and will be mutated to be assigned the lab.

Instructor.notifyTAs(body)

Preconditions: notifying instructor must contain a unique email address, there must be ≥ 1 TA in the database, the notification body can't be empty.
Postconditions: If the body is not empty, the body's message will be sent to all the TAs and will be sent to the instructor's "sent" tab. If there isn't a body or any TAs, an error message is displayed.
Side-Effects: ---
Body: input – string representing the message the instructor is trying to send.

Instructor.viewInstrTAAsgmt()

Preconditions: The instructor's assigned courses must contain at least one TA.
Postconditions: no change to the host object or the respective course, displays all of the TA's assigned to courses that are under the Instructor.
Side-Effects: ---

Instructor.getInstrCrseAsgmts()

Preconditions: Any assignments, from a course to this instructor, must be reflected by the instructor's assignments to courses.
Postconditions: no change to the host object or the respective course, returns a list of the courses the instructor is assigned to.
Side-Effects: ---

BEGIN COURSE

Course.addInstructor(activeInstructor)

Preconditions: Course doesn't already have an instructor, the instructor hasn't exceeded their max course assignments.
Postconditions: course is assigned to instructor and instructor assigned to course. Error displayed for maximum capacity reached for instructor or course.
Side-Effects: ---
activeInstructor: in out - This instructor will used to take up this courses instructor slot and it will have to be assigned to this course, it's state will be modified in the process.

Course.addTA(activeTA)

Preconditions: TA hasn't exceeded their max course assignments, the TA hasn't already been assigned to this course.
Postconditions: course is assigned to TA and the TA assigned to course. Error displayed for maximum capacity reached for TA.
Side-Effects: ---
activeTA: in out - This TA will be added to this course's list of TAs, and the TA will have to be assigned to this course.

Course.removeAssignment(activeUser)

Preconditions: The user we're trying to remove exists, and the user is assigned to this course
Postconditions: the user's assignments to this course will be removed and the courses assignments to the user will be removed. Error message displayed if non existent user or user that isn't assigned to the course.
Side-Effects: ---
activeUser: in out – This user will be used to search for a matching user in the course so it can be removed, then all the user's assignments to/from this course will be removed.

Course.removeCourse()

Preconditions: course must be active in the schedule of classes system.
Postconditions: this course will be removed from the schedule of classes GUI and no user can be assigned to it.
Side-Effects: The respective sections and assignments to this course will also be removed.

Course.viewAsgmtsForCrse()

Preconditions: All of the course's user assignments must have a matching assignment to this course. Must be at least one user assignment to this course.

Postconditions: no change to the host object or the respective users, displays all of the course’s users who are assigned to this specific course.
Side-Effects: ---

Course.viewSectionsForCrse()

Preconditions: Sections must be correctly instantiated to belong to the course and the course must respectively contain any section assigned to this course.
Postconditions: no change to the host object or the respective sections, displays all of the course’s sections who are a part of this specific course.
Side-Effects: ---

Course.viewCrseInfo()

Preconditions: All course information must be correctly instantiated in the constructor.
Postconditions: no change to the host object or the respective sections, displays all of the course information (name, ID, course description, semester,modality, sections,assignments)
Side-Effects: …

BEGIN SECTION

Section.getID()

Preconditions: Section has been assigned a unique ID at instantiation.
Postconditions: returns an integer representing the sections unique ID
Side-Effects: …

Section.getParentCourse()

Preconditions: Section must always be assigned a parent course, as per composition, and the section should not exist if the course is removed.
Postconditions: returns a course object representing the course that the section belongs to.
Side-Effects: …

BEGIN LAB SECTIONS

Lab.getLabTAAsgmt()

Preconditions: the TA field within the lab section has been correctly set to a TA that is assigned to this lab. Must be a TA assigned to return.
Postconditions: no changes to the lab’s state, just return the TA that is currently assigned. If no TA assigned then error message displayed.
Side-Effects: ---

Lab.addTA(activeTA)

Preconditions: the lab cannot already have a TA assigned, the TA cannot be assigned to grader status.
Postconditions: the TA will be assigned to this section, and the section will be assigned to the TA. Will display error message if assigning a TA with grader status, or if the Lab is already full.
Side-Effects: ---
activeTA: in out: the specific TA we want to assign to the lab and have the lab assigned to the TA

Lab.removeTA()

Preconditions: there must be a TA assigned to the lab section
Postconditions: the TA is removed from the lab section, the lab and TA itself still exist. Error displayed if no TA present.
Side-Effects: ---

BEGIN LECTURE

Lecture.getLecInstrAsgmt()

Preconditions: the instructor field within the lecture section has been correctly instantiated to be a instructor.
Postconditions: no changes to the lecture's state, just return the instructor that is currently assigned to the lecture. Error displayed if there isn't an instructor assigned.
Side-Effects: ---
…

Lecture.addInstructor(activeInstr)

Preconditions: the lecture cannot already have an instructor assigned.
Postconditions: the instructor will be assigned to this section, and the section will be assigned to the TA instructor
Side-Effects: ---
activeInstr: in out: the specific instructor we want to assign to the lecture and have the lecture assigned to the instructor

Lecture.removeInstructor()

Preconditions: there must be a instructor assigned to the lecture section

Postconditions: the instructor is removed from the lecture section, the lecture and instructor itself still exist; they’re just not connected any more (as per aggregation).
Side-Effects: ---

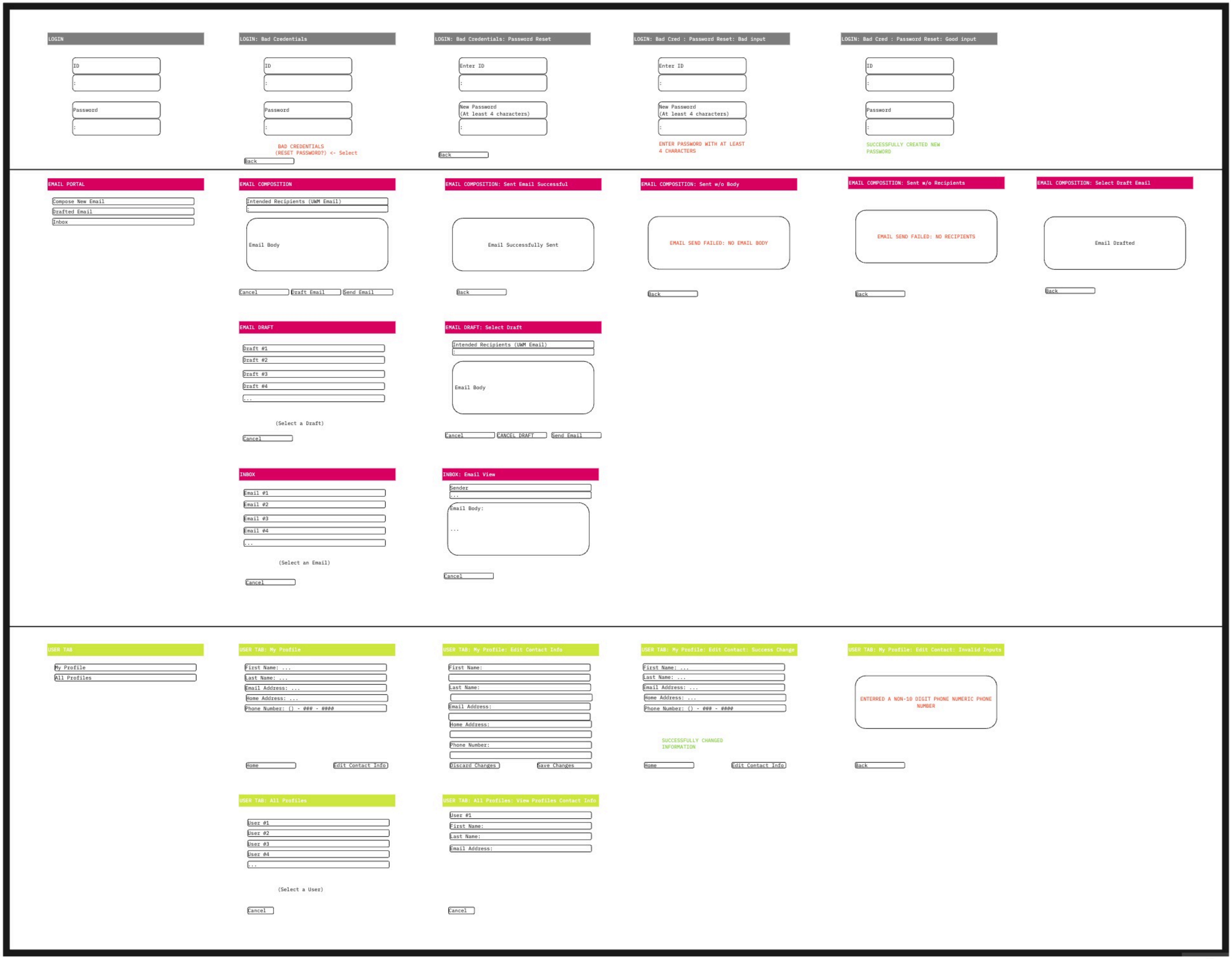
STATE DIAGRAM

"Universal" App
TA App
Instructor App
Admin App
Login
TA Application Contains All States From Universal Application.
Admin Application Contains All States From Universal Application.
Instructor Application Contains All States From Universal Application.
Email Portal
Email Composition
Sent Email
Failed Send: No Body
Failed Send: No Recievers
Drafted Email
Drafted Emails
Inbox View
Email View
Select Email Tab
Home
Select Compose New Email
Select Send email
No Email Body
No intended recipients
Select Draft email
Select Draft
Cancel Draft
Select Drafted email
Select Inbox
Select an email
Cancel
Back
User Tab
My Profile
Edit Contact Info Form
Account Changes Saved
Bad Inputs (non-10 digit phone #)
View Public Profile's Contact info
All Profiles
Select Users
Select My Profile
Select All Profile
Edit Contact Info
Discard Changes
Select Save: Valid Inputs
Select Save: Invalid Inputs
Select Account
Return
Account Management Tab
Create Account Form
Account Created
Error: Missing Fields
Bad Account ID
Creating Unique Account
Creating w/ Missing Any Fields
Creating w/ Duplicate Account
try again
Select Create Account Form
Delete Account Form
Account Found
Account Deleted
Confirm Deletion
Searching Existing Account
Searching Non-Existing Account
Select Delete Account Form
Edit Account Form
Input Form
Select Edit Account Info
Select Edit Account Form
Select Account Management
Course Management Tab
Select Course Management
Create Course Form
Course Created
Added Lab Section
add another lab
Add Lab Section
Creating Unique Course
Empty Fields
Bad Course ID
Searching Duplicate Course
Select Create Course
Display all Instructors
Display all courses
Assigned Instructor to Course
Instructor Assignment Full
Instructor's Assignments Full
Open Instructor Slot
Course w/ Instructor
Select Instructor Below Max Assignments
Select Instructor Max Assignments
Select Assign Instructor
Display all TAs
Number of Labs, Grader Status Form
Bad Input: Required "#" Labs & Grader Status
Assigned TA to Course
Course's TA Assignments full
TA's Course Assignments Full
Open TA Slot
Max Number of TAs
Select Valid # Labs & Grader Status
Non-Numeric # Labs or No grader Status
Select TA Below Max Assignments
Select TA Max Assignments
Select Assign TAs
Select assign TAs to Lab
Display All TAs
Display All Lab Sections
Assigned TA to Lab Section
TA Assignment Full
No Grader Lab Assignment
TA's Lab Assignments Full
Lab w/ TA
Select TA no Grader Satus & below Max Lab Assignments
Select TA Grader Status
Select TA at Max Assignments
return
select data access tab
Data Access Tab
Course Table
Course Info Displayed
Bad Course ID
Search Existing Course
Search Non-Existing Course
Select Course Table
Lab Table
Lab Info Displayed
Bad Lab ID
Search Existing Lab
Search Non-Existing Lab
Select Lab Table
User Table
Public and Private Info Displayed
Bad Account ID
Search Existing Account
Search Non-Existing Account
Select User Table
logout
select view assignments
View Assignments Tab
Select View course assignments
Select View TA Assignment
View All Courses Tab
Displayed Section Assignments
Select a Course
View All TA Assignment Tab
Display Assignment Info for selected TA
Select a TA
home
Selected TA Assignments
View All TA Assignment Tab
Select TA
Select View TA Assignment
Good Credentials: If TA
Good Credentials: If Admin
Good Credentials: If Instructor
Bad Credentials
Bad Login
Back: try again
Password Reset
Reset Password
Back: valid password
Failed Password Requirements
Displayed All TAs Assigned to my Course
Select Assign TAs to a Lab
Display All My Lab Sections
Assigned TA to Lab Section
TA Assignment Full
Bad TA: graders ≠ lab assignments
TA's Lab Assignments Full
Select TA: no Grader Satus & below Max Lab Assignments
Select TA: Grader Status
Select TA: Max Assignments
View Assignments Tab
Select View ALL TA Assignment
Select View MY TA Assignments
Select View course assignments
View ALL TAs
Select specific TA
Current Assignments for TA
View MY TAs Assignments
View All Courses
Select Course
Displayed Assignments for a Course
Select My Courses
Display all my Courses
My Course's Scheduler Info
My Course's Lecture Info
My Course's Lab Info
Select my course's Lecture
Select my course's Lab Section

PROTOTYPE: PT 1

TA
INSTRUCTOR
ADMIN

UNIVERSAL: applies to ALL users...



BL #19

LOGIN
ID
:
Password
:

LOGIN: Bad Credentials
ID
:
Password
:
BAD CREDENTIALS
(RESET PASSWORD?) <- Select
Back

LOGIN: Bad Credentials: Password Reset
Enter ID
:
New Password
(At least 4 characters)
:
Back

LOGIN: Bad Cred : Password Reset: Bad input
Enter ID
:
New Password
(At least 4 characters)
:
ENTER PASSWORD WITH AT LEAST
4 CHARACTERS

LOGIN: Bad Cred : Password Reset: Good input
ID
:
Password
:
SUCCESSFULLY CREATED NEW
PASSWORD

BL #5
BL #13

EMAIL PORTAL
Compose New Email
Drafted Email
Inbox

EMAIL COMPOSITION
Intended Recipients (UWM Email)
:
Email Body
Cancel
Draft Email
Send Email

EMAIL COMPOSITION: Sent Email Successful
Email Successfully Sent
Back

EMAIL COMPOSITION: Sent w/o Body
EMAIL SEND FAILED: NO EMAIL BODY
Back

EMAIL COMPOSITION: Sent w/o Recipients
EMAIL SEND FAILED: NO RECIPIENTS
Back

EMAIL COMPOSITION: Select Draft Email
Email Drafted
Back

EMAIL DRAFT
Draft #1
Draft #2
Draft #3
Draft #4
...
(Select a Draft)
Cancel

EMAIL DRAFT: Select Draft
Intended Recipients (UWM Email)
:
Email Body
Cancel
CANCEL DRAFT
Send Email

INBOX
Email #1
Email #2
Email #3
Email #4
...
(Select an Email)
Cancel

INBOX: Email View
Sender
...
Email Body:
...
Cancel

BL #10
BL #15
BL #16
BL# 18

USER TAB
My Profile
All Profiles

USER TAB: My Profile
First Name: ...
Last Name: ...
Email Address: ...
Home Address: ...
Phone Number: () - ### - ####
Home
Edit Contact Info

USER TAB: My Profile: Edit Contact Info
First Name:
Last Name:
Email Address:
Home Address:
Phone Number:
Discard Changes
Save Changes

USER TAB: My Profile: Edit Contact: Success Change
First Name: ...
Last Name: ...
Email Address: ...
Home Address: ...
Phone Number: () - ### - ####
SUCCESSFULLY CHANGED
INFORMATION
Home
Edit Contact Info

USER TAB: My Profile: Edit Contact: Invalid Inputs
ENTERRED A NON-10 DIGIT PHONE NUMERIC PHONE
NUMBER
Back

USER TAB: All Profiles
User #1
User #2
User #3
User #4
...
(Select a User)
Cancel

USER TAB: All Profiles: View Profiles Contact Info
User #1
First Name:
Last Name:
Email Address:
Cancel

TA

PROTOTYPE: PT II

BL #17

TA APPLICATION

Email Portal

User Tab

View All TA Assignments tab

Logout

VIEW ALL TA ASSIGNMENTS TAB

TA #1: (First Name) (Last Name): (ID)

TA #2: (First Name) (Last Name): (ID)

TA #3: (First Name) (Last Name): (ID)

TA #4: (First Name) (Last Name): (ID)

TA ...

(Select a TA)

Cancel

VIEW ALL TA ASSIGNMENTS Tab: Select TA

Course #1

Section Assignment #...

Lab/Lecture

Course #2

Section Assignment #...

Lab/Lecture

Course #...

Section Assignment #...

Lab/Lecture

Cancel

INSTRUCTOR **PROTOTYPE PT.III**

BL #12
Extra Fluff - my TA assignments
BL #11

Extra Fluff - my Course Info

BL #14

INSTRUCTOR APPLICATION
Email Portal
User Tab
View Assignments Tab
My Courses Tab
Assign My TAs a Lab Section
Logout

BL #12
Extra Fluff - my TA assignments
BL #11

VIEW ASSIGNMENTS TAB:
View ALL TA Assignments
View MY TA Assignments
View ALL Course Assignments
Home

VIEW ASSIGNMENTS TAB: View ALL TA Assignments
TA #1: (First Name) (Last Name): (ID)
TA #2: (First Name) (Last Name): (ID)
TA #3: (First Name) (Last Name): (ID)
TA #4: (First Name) (Last Name): (ID)
TA ...
(Select a TA)
Cancel

VIEW ASSIGNMENTS TAB: View All TA Asgnmts: Select
TA ID#:
Course #1
Section Assignment ID#:...
Grader?
Course #2
Section Assignment ID#:...
Grader?
Course #...
Section Assignment ID#:...
Grader?
Cancel

VIEW ASSIGNMENTS TAB: View MY TA Assignments
TA #1: ...
Course ID#: ...
Section ID#: ...
TA #2: ...
Course ID#: ...
Section ID#: ...
TA #3: ...
Course ID#: ...
Section ID#: ...
Cancel

VIEW ASSIGNMENTS TAB: All Courses
Course #1
Course #2
Course #3
Course #4
Course ...
(Select a Course)
Cancel

VIEW ASSIGNMENTS TAB: All Courses: Selected Course
Course #1
Lecture: Assignment
User: ...
Lab #1: Assignment
User: ...
Lab #2: Assignment
User: ...
Cancel

BL #??? -> fluff

MY COURSES:
My Course #1
My Course #2
My Course #3
My Course #4
My Course ...
(Select a Course)
Cancel

MY COURSES: Selected Course
Course Name: ...
Course Description: ...
Instructor: ...
Meeting Times: ...
View My Course's Lecture
View My Course's Labs
Cancel

MY COURSES: Selected Course: View Lecture
Lecture ID: ...
Instructor First & Last Name: ...
Email: ...
Cancel

MY COURSES: Selected Course: View Labs
Lab #1 ID: ...
TA ID#: ...
Lab #2 ID: ...
TA ID#: ...
Lab #3 ID: ...
TA ID#: ...
Cancel

BL #14

ASSIGN MY TAS:
TA #1: ...
Course ID#: ...
TA #2: ...
Course ID#: ...
TA #3: ...
Course ID#: ...
(Select a TA)
Cancel

ASSIGN MY TAS: Selected TA Grader
SELECTED A TA WITH GRADER STATUS
BACK

ASSIGN MY TAS: Select TA Full
SELECTED A TA WITH MAXIMUM LAB ASSIGNMENTS
BACK

ASSIGN MY TAS: Assign Lab
Lab1 ID #: ...
Lab2 ID #: ...
Lab3 ID #: ...
Lab# ID #: ...
(Select a Lab)
Cancel

ASSIGN MY TAS: Assign Lab: Full Lab
SELECTED A LAB ALREADY ASSIGNED TO A TA
BACK

ASSIGN MY TAS: Assign Lab: Success
successfully assigned your TA to one of your Labs
Back

ADMIN

PROTOTYPE PT IV - a) 1st half of Admin

BL #?? -> fluff

Extra Fluff - view assignments

BL #2
BL #3
BL #4

BL #1
BL #7
BL #8

BL#9

BL#6

ADMIN APPLICATION
Email Portal
User Tab
View Assignments Tab
Account Management Tab
Course Management Tab
Assign TAs to Lab Tab
Data Access Tab
Logout

VIEW ASSIGNMENTS TAB:
View ALL TA Assignments
View ALL Course Assignments
Home

VIEW ASSIGNMENTS TAB: View ALL TA Assignments
TA #1: (First Name) (Last Name): (ID)
TA #2: (First Name) (Last Name): (ID)
TA #3: (First Name) (Last Name): (ID)
TA #4: (First Name) (Last Name): (ID)
TA ...
(Select a TA)
Cancel

VIEW ASSIGNMENTS TAB: View All TA Asgnmts: Select
TA ID#:
Course #1
Section Assignment ID#:...
Grader?
Course #2
Section Assignment ID#:...
Grader?
Course #...
Section Assignment ID#:...
Grader?
Cancel

VIEW ASSIGNMENTS TAB: View All Course Assignments
Course #1
Course #2
Course #3
Course #4
Course ...
(Select a Course)
Cancel

VIEW ASSIGNMENTS TAB: View All Crse Asgnmts: Select
Course #1
Lecture: Assignment
User: ...
Lab #1: Assignment
User: ...
Lab #2: Assignment
User: ...
Cancel

BL #2
BL #3
BL #4

ACCOUNT MANAGEMENT TAB
Create Account Form
Delete Account Form
Edit Account Form
Home

ACCOUNT MANAGEMENT TAB: Create Account Form
Unique ID:*
Role:*
First Name:*
Last Name:*
Email Address:*
Home Address:*
Phone Number:*
Cancel
Create Account

ACCOUNT MANAGEMENT TAB: Create Account: Missing Field
MISSING ACCOUNT FIELDS
Try Again

ACCOUNT MANAGEMENT TAB: Create Account: Bad ID
BAD ID: TRIED TO CREATE DUPLICATE ACCOUNT
Try Again

ACCOUNT MANAGEMENT TAB: Create Account: Success
successfully created account
Back

ACCOUNT MANAGEMENT TAB: Delete Account Form
Search for Account (Enter ID):*
:
Cancel

ACCOUNT MANAGEMENT TAB: Delete Account: Bad ID
SEARCHED FOR NON-EXISTANT ACCOUNT
Try Again

ACCOUNT MANAGEMENT TAB: Delete Account: Found
User ID: ...
First Name: ...
Last Name: ...
Email Address: ...
Home Address: ...
Phone Number: () - ### - ####
Cancel
Confirm Deletion

ACCOUNT MANAGEMENT TAB:Delete Account: Found: Success
successfully deleted account
Back

ACCOUNT MANAGEMENT TAB: Edit Account Form
Search for Account (Enter ID):*
:
Cancel

ACCOUNT MANAGEMENT TAB: Edit Account: Bad ID
SEARCHED FOR NON-EXISTANT ACCOUNT
Try Again

ACCOUNT MANAGEMENT TAB: Edit Account: Found
User ID: ...
First Name: ...
Last Name: ...
Email Address: ...
Home Address: ...
Phone Number: () - ### - ####
Cancel
Edit Account Info

ACCOUNT MANAGEMENT TAB: Edit Account: Found: Input
First Name:
Last Name:
Email Address:
Home Address:
Phone Number:
Discard Changes
Save Changes

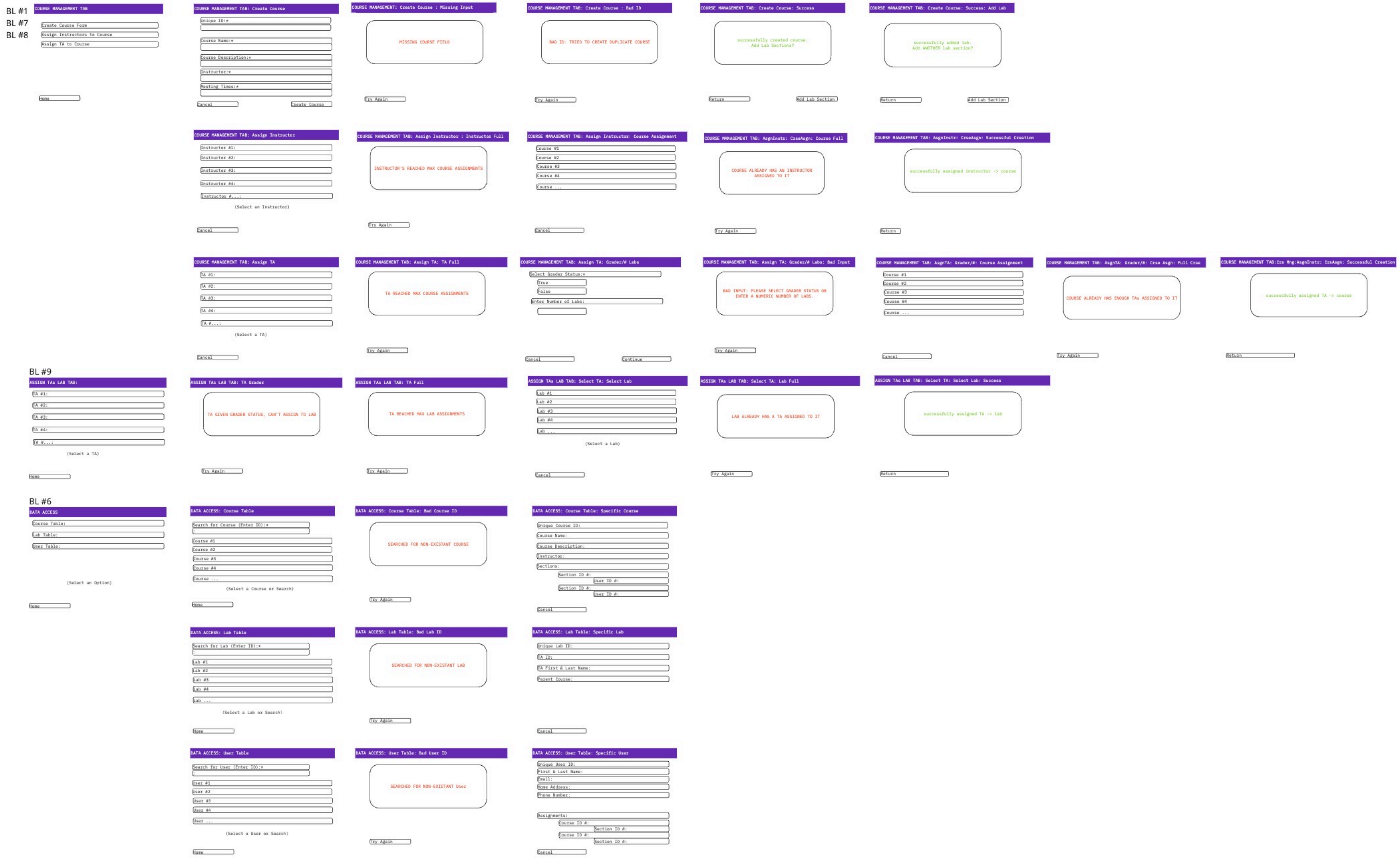
ACCOUNT MANAGEMENT TAB: Edit Account: Input: Invalid Change
ENTERRED A NON-10 DIGIT PHONE NUMERIC PHONE NUMBER
Back

ACCOUNT MANAGEMENT TAB: Edit Account: Input: Valid Changes
SUCCESSFULLY CHANGED INFORMATION
Return

BL #1
BL #7
BL #8

ADMIN

PROTOTYPE PT IV - b) 2nd half of Admin



BL #1
BL #7
BL #8

COURSE MANAGEMENT TAB
Create Course Form
Assign Instructors to Course
Assign TA to Course
Home

COURSE MANAGEMENT TAB: Create Course
Unique ID:*
Course Name:*
Course Description:*
Instructor:*
Meeting Times:*
Cancel
Create Course

COURSE MANAGEMENT: Create Course : Missing Input
MISSING COURSE FIELD
Try Again

COURSE MANAGEMENT TAB: Create Course : Bad ID
BAD ID: TRIED TO CREATE DUPLICATE COURSE
Try Again

COURSE MANAGEMENT TAB: Create Course: Success
successfully created course.
Add Lab Sections?
Return
Add Lab Section

COURSE MANAGEMENT TAB: Create Course: Success: Add Lab
successfully added lab.
Add ANOTHER lab section?
Return
Add Lab Section

COURSE MANAGEMENT TAB: Assign Instructor
Instructor #1:
Instructor #2:
Instructor #3:
Instructor #4:
Instructor #...:
(Select an Instructor)
Cancel

COURSE MANAGEMENT TAB: Assign Instructor : Instructor Full
INSTRUCTOR'S REACHED MAX COURSE ASSIGNMENTS
Try Again

COURSE MANAGEMENT TAB: Assign Instructor: Course Assignment
Course #1
Course #2
Course #3
Course #4
Course ...
Cancel

COURSE MANAGEMENT TAB: AsgnInstr: CrseAsgn: Course Full
COURSE ALREADY HAS AN INSTRUCTOR ASSIGNED TO IT
Try Again

COURSE MANAGEMENT TAB: AsgnInstr: CrseAsgn: Successful Creation
successfully assigned instructor -> course
Return

COURSE MANAGEMENT TAB: Assign TA
TA #1:
TA #2:
TA #3:
TA #4:
TA #...:
(Select a TA)
Cancel

COURSE MANAGEMENT TAB: Assign TA: TA Full
TA REACHED MAX COURSE ASSIGNMENTS
Try Again

COURSE MANAGEMENT TAB: Assign TA: Grader/# Labs
Select Grader Status:*
True
False
Enter Number of Labs:
Cancel
Continue

COURSE MANAGEMENT TAB: Assign TA: Grader/# Labs: Bad Input
BAD INPUT: PLEASE SELECT GRADER STATUS OR ENTER A NUMERIC NUMBER OF LABS.
Try Again

COURSE MANAGEMENT TAB: AsgnTA: Grader/#: Course Assignment
Course #1
Course #2
Course #3
Course #4
Course ...
Cancel

COURSE MANAGEMENT TAB: AsgnTA: Grader/#: Crse Asgn: Full Crse
COURSE ALREADY HAS ENOUGH TAs ASSIGNED TO IT
Try Again

COURSE MANAGEMENT TAB:Crs Mng:AsgnInstr: CrsAsgn: Successful Creation
successfully assigned TA -> course
Return

BL #9

ASSIGN TAs LAB TAB:
TA #1:
TA #2:
TA #3:
TA #4:
TA #...:
(Select a TA)
Home

ASSIGN TAs LAB TAB: TA Grader
TA GIVEN GRADER STATUS, CAN'T ASSIGN TO LAB
Try Again

ASSIGN TAs LAB TAB: TA Full
TA REACHED MAX LAB ASSIGNMENTS
Try Again

ASSIGN TAs LAB TAB: Select TA: Select Lab
Lab #1
Lab #2
Lab #3
Lab #4
Lab ...
(Select a Lab)
Cancel

ASSIGN TAs LAB TAB: Select TA: Lab Full
LAB ALREADY HAS A TA ASSIGNED TO IT
Try Again

ASSIGN TAs LAB TAB: Select TA: Select Lab: Success
successfully assigned TA -> lab
Return

BL #6

DATA ACCESS
Course Table:
Lab Table:
User Table:
(Select an Option)
Home

DATA ACCESS: Course Table
Search for Course (Enter ID):*
:
Course #1
Course #2
Course #3
Course #4
Course ...
(Select a Course or Search)
Home

DATA ACCESS: Course Table: Bad Course ID
SEARCHED FOR NON-EXISTANT COURSE
Try Again

DATA ACCESS: Course Table: Specific Course
Unique Course ID:
Course Name:
Course Description:
Instructor:
Sections:
Section ID #:
User ID #:
Section ID #:
User ID #:
Cancel

DATA ACCESS: Lab Table
Search for Lab (Enter ID):*
:
Lab #1
Lab #2
Lab #3
Lab #4
Lab ...
(Select a Lab or Search)
Home

DATA ACCESS: Lab Table: Bad Lab ID
SEARCHED FOR NON-EXISTANT LAB
Try Again

DATA ACCESS: Lab Table: Specific Lab
Unique Lab ID:
TA ID:
TA First & Last Name:
Parent Course:
Cancel

DATA ACCESS: User Table
Search for User (Enter ID):*
:
User #1
User #2
User #3
User #4
User ...
(Select a User or Search)
Home

DATA ACCESS: User Table: Bad User ID
SEARCHED FOR NON-EXISTANT User
Try Again

DATA ACCESS: User Table: Specific User
Unique User ID:
First & Last Name:
Email:
Home Address:
Phone Number:
Assignments:
Course ID #:
Section ID #:
Course ID #:
Section ID #:
Cancel