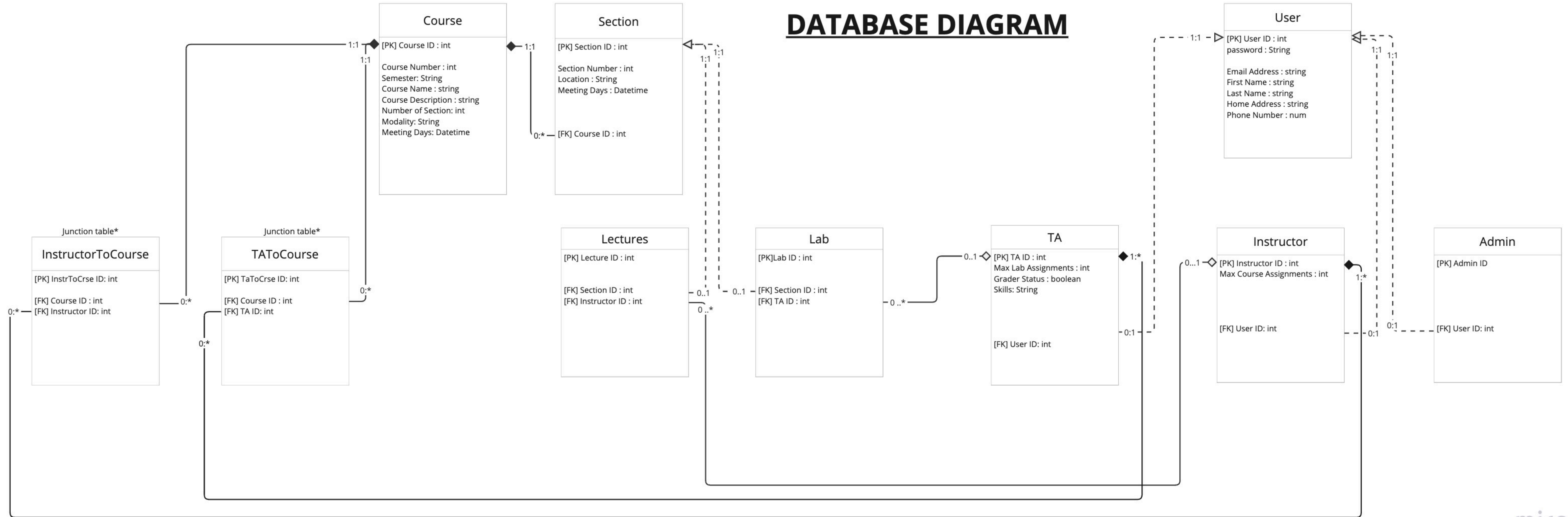


DATABASE DIAGRAM



UNIVERSAL
applies to ALL users...

PROTOTYPES SPRINT 2

LOGIN

Login

id

Password

Forgot Password

Authenticates username and password in the database.

Form Variables: username (String), password (String)

Context Variables: ...

Bad Credentials

Login

id

Password

Back

Bad CREDENTIALS

User submitted wrong username or password.

Form Variables: username (String), password (String)

Context Variables: ...

Bad Credentials : Password Reset

Password Reset Page

Enter ID

New Password

(At least 4 characters)

Back

User selected "forgot password"

Form Variables: username (String), password (String)

Context Variables: ...

Bad Credentials : Password Reset: Bad Input

Password Reset Page

Enter ID

New Password

(At least 4 characters)

ENTER PASSWORD WITH AT LEAST 4 CHARACTERS

User's new password no meeting 4 character requirement

Form Variables: username (String), password (String)

Context Variables: ...

Bad Credentials : Password Reset: Good Input

Login

id

Password

SUCCESSFULLY CREATED NEW PASSWORD

User entered valid ID & new password in reset page

Form Variables: username (String), password (String)

Context Variables: ...

EMAIL PORTAL

Compose New Email

Drafted Email

Index

EMAIL COMPOSITION

Extended Recipients (UMP Email)

Email Body

Cancel

Draft Email

Send Email

Email page to email other users.

Form Variables: recipient (int), body (int)

Context Variables: ...

EMAIL COMPOSITION: Sent Email Successful

Email Successfully Sent

Back

Email page to email other users.

Form Variables: recipient (int), body (int)

Context Variables: ...

EMAIL COMPOSITION: Sent w/o Body

EMAIL SEND FAILED: NO EMAIL BODY

Back

Emails need a body to send.

Form Variables: ...

Context Variables: ...

EMAIL COMPOSITION: Sent w/o Recipients

EMAIL SEND FAILED: NO RECIPIENTS

Back

Emails need a recipient to send.

Form Variables: ...

Context Variables: ...

EMAIL COMPOSITION: Select Draft Email

Email Drafted

Back

Selected draft.

Form Variables: ...

Context Variables: ...

EMAIL DRAFT

Extended Recipients (UMP Email)

Email #1

Email #2

Email #3

Email #4

...

(Select a draft)

Cancel

Select a draft to resume

Form Variables: ...

Context Variables: drafts (Draft - New Model?)

EMAIL: Email View

Sender

Email Body

...

Cancel

Reading an email sent to my account

Form Variables: ...

Context Variables: sender (int), body (int)

USER TAB

My Profile

All Profiles

USER TAB: My Profile

My Profile

First Name

Last Name

Current Address

Home Address

() - ### - ####

Assignments

Course

Section

Enrolled

Home

Edit Contact (info)

View my profile's fields

Form Variables: ...

Context Variables: First/Last name (int), emailAddress (int), homeAddress (int), phoneNum (int), assignments (course, section)

USER TAB: All Profiles

All Profiles

User #1

User #2

User #3

User #4

...

Cancel

View User

Select a profile to view

Form Variables: ...

Context Variables: userName (int), userID (int)

My Profile: Edit Contact: Info

Edit My Profile

First Name

Last Name

Current Address

Home Address

Phone Number

Assignments

Course

Section

Enrolled

Home

Edit Contact (info)

Edit my profile's fields, need valid phone number.

Form Variables: First/Last name (int), emailAddress (int), homeAddress (int), phoneNum (int)

Context Variables: ...

My Profile: Edit Contact: Success Change

SUCCESSFULLY CHANGED INFORMATION

Home

Edit Contact (info)

Successful account change

Form Variables: ...

Context Variables: ...

My Profile: Edit Contact: Enrolled Inquiry

ENTERED A NON-10 DIGIT PHONE NUMERIC PHONE NUMBER

Back

Invalid phone number

Form Variables: ...

Context Variables: ...

ALL Profiles: View Profiles Contact Info

View Profile

User #1

First Name

Last Name

Email Address

Cancel

Viewing selected profile

Form Variables: ...

Context Variables: userID (int), First/Last name (int), email (int)

INSTRUCTOR

PROTOTYPES

SPRINT 2

INSTRUCTOR APPLICATION

Instructor Home

View TAs
Assignments Tab
Assign My TAs -> Section
Email Portal
User Tab

Logout

Instructor home
Form Variables: ...
Context Variables: ...

View TAs

View All TAs

TA #1: ...
Course ID#: ...
Section ID#: ...

TA #2: ...
Course ID#: ...
Section ID#: ...

TA #3: ...
Course ID#: ...
Section ID#: ...

Cancel VIEW MY TAs

Display all active TAs and their course/sec assignment

Form Variables: ...

Context Variables: talD (int), first/last name(str), course/section assignments IDs (int)

View ALL TAs: No active TAs

No Existing TAs to view.

Selected view TAs, but there aren't any TAs

Form Variables: ...

Context Variables: ...

View TAs

View My TAs

TA #1: ...
Course Name#: ...
Section ID#: ...

TA #2: ...
Course Name#: ...
Section ID#: ...

Cancel

Display all my TAs and their course/sec assignment

Form Variables: ...

Context Variables: talD (int), first/last name(str), courseName(str), sectionAssignments IDs (int)

Select Specific TA

Selected TA

TA ID#: ... First/last Name: ...
Skills: ...

Course #1
Section Assignment ID#: ...
Grader?: ...

Course #2
Section Assignment ID#: ...
Grader?: ...

Course #...
Section Assignment ID#: ...
Grader?: ...

Cancel

Select a specific TA

Form Variables: ...

Context Variables: talD (int), first/last (str), skills (str), graderStatus (boolean), course/section assignments.

ASSIGNMENTS:

Assignments

View Courses
View my Sections

None

Select whether want to see my courses/section/TAs

Form Variables: ...

Context Variables: ...

View Courses

View All Courses

Course ID#1 Name: ...
Course ID#2 Name: ...
Course ID#3 Name: ...
Course ID#4 Name: ...
Course ...

Cancel VIEW MY COURSES

Display all Courses

Form Variables: ...

Context Variables: courseName (str), courseID (int)

View My Courses

View My Courses

My Course ID#1 Name: ...
My Course ID#2 Name: ...
My Course ...

Cancel (Select a Course)

Display all of my Courses

Form Variables: ...

Context Variables: courseName (str), courseID (int)

Selected Course

Selected Course

Course Name: ...
Course Description: ...

Instructor: ...

Meeting Times: ...

TA: ...
Skills: ...

Cancel

Display specific course information

Form Variables: ...

Context Variables: courseName (str), courseID (int), courseDescr (str), instructor (Instructor), MeetingTimes (dateTime), ta (TA), skills (str)

View My Section

View My Sections

My Section #1, UserID: #1
My Section #2, ...
My Section #3, ...
My Section #4, UserID: #2
My Section ...

(Select a Section)

Cancel

Display all of my Sections and assignments if applicable

Form Variables: ...

Context Variables: secID (int), secType (String), userID (int)

View My Section: No active my sections

No Sections Assigned to your courses

clicked on view my section but no sections.

Form Variables: ...

Context Variables: ...

View My Section: Select Section

Selected Section

Section #1 ID: ...
User ID#: ..., Name: ...
Course ID#: ...
Instructor ID#: ..., Name: ...

Cancel

Display section fields for selected section, course name/instruct

Form Variables: ...

Context Variables: secID(int), user assignment ID(int) name(str), courseID (int), course's instructor assignment ID (int) name(str)

ASSIGN MY TAs:

Section Selection

Section ID #: ...
Section ID #: ...
Section ID #: ...
Section# ID #: ...

Cancel (Select a Lab)

Display my sections to potentially assign a TA to.

Form Variables: ...

Context Variables: secID (int)

ASSIGN MY TAs: Full

SECTION ALREADY ASSIGNED A TA

BACK

Selected a section but already has a TA

Form Variables: ...

Context Variables: ...

ASSIGN MY TAs: Selected section

TA Selection

TA ID#1: ...
Name: ...
Skills: ...

TA ID#2: ...
Name: ...
Skills: ...

TA ID#3: ...
Name: ...
Skills: ...

Cancel

displays all TAs a part of the section's course

Form Variables: ...

Context Variables: talD (int), first/last name (str), skills (str)

ASSIGN MY TAs: Selected section: Full TA

SELECTED A TA ALREADY ASSIGNED TO SECTION

BACK

TA can't be assigned to same section

Form Variables: ...

Context Variables: ...

ASSIGN MY TAs: Selected TA Grader

SELECTED A TA WITH WRONG GRADER STATUS

BACK

Can't assign a TA lab w/ GS, or TA lec w/o GS

Form Variables: ...

Context Variables: ...

ASSIGN MY TAs: Assign Section: Success

Successfully assigned your TA to one of your Section

Back

Picked available TA & Section

Form Variables: ...

Context Variables: ...

PROTOTYPES

SPRINT 2

TA

TA APPLICATION

TA Home

View TAs

Set Skills

Email Portal

User Tab

Logout

Logged in as TA, interface to interact with app

Form Variables: ...

Context Variables: ...

VIEW ALL TA ASSIGNMENTS:

View All TAs

TA #1: ...

Course ID#: ...

Section ID#: ...

TA #2: ...

Course ID#: ...

Section ID#: ...

TA #3: ...

Course ID#: ...

Section ID#: ...

Cancel

(Select a TA)

Viewing all of the TAs to select

Form Variables: ...

Context Variables: talD (int), first/last name(str), course/section assignments IDs (int)

VIEW ALL TA ASSIGNMENTS: Select TA

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

Skills:

Course #1

Section Assignment #...

Lab/Lecture

Grader Status:

Course #2

Section Assignment #...

Lab/Lecture

Grader Status:

Cancel

Selected a TA, display all their courses & section assignments

Form Variables: ...

Context Variables: talD (int), skills (str), first/last name (str), assignments to courses/sections IDs (int), grader status (boolean)

TA APP: Set Skills

Set Skills

TA #1: ...

Name:

Skills:

Cancel

Confirm Edit

Display my skills to edit

Form Variables: ...

Context Variables: talD (Int), name (str), skills (str)

TA APP: Invalid Input

ERROR: CAN'T HAVE MISSING SKILLS

Fry Again

Can't enter "no skills"

Form Variables: ...

Context Variables: ...

TA APP: Set Skills

successfully edited TA Skills

Successssful skill change

Form Variables: ...

Context Variables:...

ADMIN HOME

Admin Home

Account Management Tab

Course Management Tab

Section Management Tab

Data Access Tab

Email Portal

User Tab

Logout

Logged in as Admin, interface to interact with app

Form Variables: ...

Context Variables: ...

Account Management Tab

Account Management

Create Account Form

Delete Account Form

Edit Account Form

None

Organized interface to add/remove/edit users.

Form Variables: ...

Context Variables: ...

Create Account Form

Create Account

Unique ID:

Role:

Grader Status:

First Name:

Last Name:

Email Address:

Home Address:

Phone Number:

Cancel Create Account

Form to receive user input and potentially add user to database.

Form Variables: ID (int), role (String), graderStatus (boolean), firstName (String), lastName (String), email (String), homeAddress (String), phoneNum (int)

Context Variables: ...

Create Account: Missing Field

MISSING ACCOUNT FIELDS

Try Again

Did not fill in all fields for an account

Form Variables: ...

Context Variables: ...

Create Account: Bad ID

BAD ID: TRIED TO CREATE DUPLICATE ACCOUNT

Try Again

Entered duplicate ID

Form Variables: ...

Context Variables: ...

Create Account: Bad Phone

BAD PHONE NUMBER: must be 10 digit int

Try Again

Phone number must be entered as an int.

Form Variables: ...

Context Variables: ...

Create Account: Success

successfully created account

Back

Valid input, no duplicates, completed all fields

Form Variables: ...

Context Variables: ...

Delete Account Form

Delete Account Selection

User ID: ... Name: ... Role: ...

User ID: ... Name: ... Role: ...

User ID: ... Name: ... Role: ...

User ID: ... Name: ... Role: ...

...

Cancel Delete Account

Select a user to delete

Form Variables: ...

Context Variables: ID (int), role (String), firstName (String), lastName (String)

No Users

No Existing Users to Delete.

Try Again

Clicked on "Delete Account Form" but no users exists.

Form Variables: ...

Context Variables: ...

Delete Account: Confirm

Confirm Delete?

User ID: ... Name: ... Role: ...

Cancel Confirm Relation

Clicked on "Delete Account Form" & selected user to delete.

Form Variables: ...

Context Variables: ID (int), role (String), firstName (String), lastName (String)

Delete Account: Confirm: Success

successfully deleted account

Back

Confirmed delete account

Form Variables: ...

Context Variables: ...

Edit Account Form

Edit Account Selection

User ID: ... Name: ... Role: ...

User ID: ... Name: ... Role: ...

User ID: ... Name: ... Role: ...

User ID: ... Name: ... Role: ...

...

Cancel Edit Account

Select a user to edit

Form Variables: ...

Context Variables: ID (int), role (String), firstName (String), lastName (String)

No Users

No Existing Users to Edit.

Try Again

Clicked on "Edit Account Form" but no users exists.

Form Variables: ...

Context Variables: ...

Edit Account: Input

Edit Account

User ID: ...

First Name:

Last Name:

Email Address:

Home Address:

Phone Number:

Discard Changes Save Changes

User exists, take all admin input to change user info.

Form Variables: , firstName (String), lastName (String), email (String), homeAddress (String), phoneNum (int)

Context Variables: ID (int)

Edit Account: Input: Invalid Change

BAD PHONE NUMBER: must be 10 digit int

Back

Phone number must be entered as a 10 digit int.

Form Variables: ...

Context Variables: ...

Edit Account: Input: Missing Fields

MISSING ACCOUNT FIELDS

Back

Removed required fields

Form Variables: ...

Context Variables: ...

Edit Account: Input: Valid Changes

SUCCESSFULLY CHANGED INFORMATION

Save

Valid input, completed all fields

Form Variables: ...

Context Variables: ...

Course Management Tab

Course Management

Create Course Form

Delete Course Form

Edit Course Form

Assign Instructors -> Course Form

Assign TA -> Course Form

Done

Organized interface to add/remove/edit courses, and assign instr

Form Variables: ...
Context Variables: ...

Create Course

Create Course

Unique Course ID:

Course Name:

Course Description:

Number of Labs:

Semester:

Meeting Times:

Modality:

Cancel Create Course

Form to receive user input and potentially add course to database.

Form Variables: ID(int), crsName(String), crsDesc(String), numLabs(int), semstr(String), modality(String), meetingTime(DateTime)

Context Variables: ...

Create Course : Missing Input

MISSING COURSE FIELD

Try Again

Did not fill in all fields for a course
Form Variables: ...
Context Variables: ...

Create Course : Bad ID

BAD ID: TRIED TO CREATE DUPLICATE COURSE

Try Again

Entered duplicate ID
Form Variables: ...
Context Variables: ...

Create Course: Success

successfully created course.

Return

Valid input, no duplicates, completed all fields
Form Variables: ...
Context Variables: ...

Delete Course Form

Delete Course Selection

Course ID: ... Name: ...

Course ID: ... Name: ...

Course ID: ... Name: ...

Course ID: ... Name: ...

...

Cancel Delete Course

Select a course to delete

Form Variables: ...
Context Variables: ID (int), name (String)

No Courses

No Existing Course to Delete.

Try Again

Clicked on "Delete Course Form" but no course exists.
Form Variables: ...
Context Variables: ...

Delete Course: Confirm

Confirm Delete?

Course ID: ... Name: ...

Cancel Confirm Selection

Clicked on "Delete Course Form" & selected course to delete.

Form Variables: ...
Context Variables: ID (int), name (String)

Delete Course: Confirm: Success

successfully deleted course

Back

Confirmed delete course
Form Variables: ...
Context Variables: ...

Edit Course Form

Edit Course Selection

Course ID: ... Name: ...

Course ID: ... Name: ...

Course ID: ... Name: ...

Course ID: ... Name: ...

...

Cancel Edit Course

Select a course to edit

Form Variables: ...
Context Variables: ID (int), name (String)

No Course

No Existing Courses to Edit.

Try Again

Clicked on "Edit Course Form" but no courses exists.
Form Variables: ...
Context Variables: ...

Edit Course: Input

Edit Course

Course ID:

Course Name:

Course Description:

Number of Labs:

Semester:

Meeting Times:

Modality:

Discard Changes Save Changes

Courses exists, take all admin input to change course info.

Form Variables: crsName(String), crsDesc(String), numLabs(int), semstr(String), modality(String), meetingTime(DateTime)

Context Variables: ID (int)

Edit Course: Input: Invalid Change

Invalid Input

Back

Entered invalid inputs: # of labs must be int.
Form Variables: ...
Context Variables: ...

Edit Course: Input: Missing Fields

MISSING COURSE FIELDS

Back

Removed required fields
Form Variables: ...
Context Variables: ...

Edit Course: Input: Valid Changes

SUCCESSFULLY CHANGED INFORMATION

Return

Valid input, completed all fields
Form Variables: ...
Context Variables: ...

Assign Instructor

Assign Instructor Selection

User ID: ... Name: ...

User ID: ... Name: ...

User ID: ... Name: ...

User ID: ... Name: ...

...

Cancel Select Instructor

Select an instructor to assign

Form Variables: ...
Context Variables: instID (int), instName (String)

No Course

No Existing Courses to Assign.

Clicked on "Assign Instructor Form" but no courses exists.

Form Variables: ...
Context Variables: ...

No Instructors

No Existing Instructor to Assign.

Clicked on "Assign Instructor Form" but no instructor exists.

Form Variables: ...
Context Variables: ...

Assign Instructor: Full

Instructor has max assignments

Clicked on instructor but has reached max course assignments

Form Variables: ...
Context Variables: ...

Assign Instructor: Course Select:

Assign Course Selection

Course ID: ... Name: ...

Course ID: ... Name: ...

Course ID: ... Name: ...

Course ID: ... Name: ...

...

Cancel Select Course

Select a course to assign the instructor to

Form Variables: ...
Context Variables: ID (int), name (String)

Assign Instructor: Course Select: Course Full

COURSE SECTIONS ALL FULL

Try Again

Course already has an instructor assigned to it
Form Variables: ...
Context Variables: ...

Assign Instructor: Course Select: Success

successfully assigned instructor -> course

Return

Successfully selected available instructor & course to assign.
Form Variables: ...
Context Variables: ...

PROTOTYPES

SPRINT 2

Section Management Tab

Section Management

Create Section Form

Delete Section Form

Edit Section Form

Assign User -> Section Form

None

Organized interface to add/remove/edit sections, and assign TA

Form Variables: ...
Context Variables: ...

Assign TA

Assign TA Selection

User ID: ... Name: ...

User ID: ... Name: ...

User ID: ... Name: ...

User ID: ... Name: ...

...

Cancel Select Instructor

Select a TA to assign

Form Variables: ...
Context Variables: taID (int), taName (String)

No Course

No Existing Courses to Assign.

Clicked on "Assign TA Form" but no courses exist.

Form Variables: ...
Context Variables: ...

No TA

No Existing TAs to Assign.

Clicked on "Assign TA Form" but no TA exists.

Form Variables: ...
Context Variables: ...

Assign TA: Full

Instructor has max assignments

Clicked on TA but has reached max course assignments

Form Variables: ...
Context Variables: ...

Assign TA: Course Select:

Assign Course Selection

Course ID: ... Name: ...

Course ID: ... Name: ...

Course ID: ... Name: ...

Course ID: ... Name: ...

...

Cancel Select Course

Select a course to assign the TA to

Form Variables: ...
Context Variables: courseID (int), name (String)

Assign TA: Course Select: Course Full

COURSE SECTIONS ALL FULL

Course already has an TA assigned to it

Form Variables: ...
Context Variables: ...

Assign TA: Course Select: Success

successfully assigned TA -> course

Successfully selected available TA & course to assign.

Form Variables: ...
Context Variables: ...

Create Section

Create Section

Unique Host Course ID: ...

Unique Section ID: ...

Section Type: ...

Meeting Days: ...

Hours: ...

Location: ...

Cancel Create Section

Form to receive user input and potentially add section to database.

Form Variables: ID(int), secType(String), meetingDays(DateTime), hours(DateTime), location(String)
Context Variables: ...

Create Section : Missing Input

MISSING SECTION FIELD

Did not fill in all fields for a section

Form Variables: ...
Context Variables: ...

Create Section : Bad ID

BAD ID: TRIED TO CREATE DUPLICATE SECTION

Entered duplicate ID

Form Variables: ...
Context Variables: ...

Create Section : Invalid Host Course ID

BAD HOST COURSE ID: COURSE DOES NOT EXIST

Entered invalid host course ID for a section

Form Variables: ...
Context Variables: ...

Create Section: Success

successfully created lecture.

Valid input, no duplicates, completed all fields, lecture type

Form Variables: ...
Context Variables: ...

Create Section: Success

successfully created lab.

Valid input, no duplicates, completed all fields, lab type

Form Variables: ...
Context Variables: ...

Delete Section Form

Delete Section Selection

Section ID: ... Name: ... Type: ...

Section ID: ... Name: ... Type: ...

Section ID: ... Name: ... Type: ...

Section ID: ... Name: ... Type: ...

...

Cancel Delete Section

Select a section to delete, if sections exist

Form Variables: ...
Context Variables: ID (int), name (String), secType (String)

No Section

No Existing Section to Delete.

Clicked on "Delete Section Form" but no section exists.

Form Variables: ...
Context Variables: ...

Delete Section: Confirm

Confirm Delete?

Section ID: ... Name: ... Type: ...

Clicked on "Delete Section Form" & selected section to delete.

Form Variables: ...
Context Variables: ID (int), name (String), secType (String)

Delete Section: Confirm: Success

successfully deleted section

Confirmed delete section

Form Variables: ...
Context Variables: ...

Edit Section Form

Edit Section Selection

Section ID: ... Name: ... Type: ...

Section ID: ... Name: ... Type: ...

Section ID: ... Name: ... Type: ...

Section ID: ... Name: ... Type: ...

...

Cancel Edit Section

Select a section to delete

Form Variables: ...
Context Variables: ID (int), name (String), secType (String)

No Section

No Existing Section to Edit.

Clicked on "Edit Section Form" but no section exists.

Form Variables: ...
Context Variables: ...

Edit Section: Input

Edit Section

Section ID: ...

Host Course ID: ...

Section Type: ...

Meeting Days: ...

Hours: ...

Location: ...

Discard Changes Save Changes

Selected section, taking admins input to change section info.

Form Variables: secType(String), meetingDays(DateTime), hours(DateTime), location(String)
Context Variables: secID(int), hostCourseID(int)

Edit Section: Input: Missing Fields

MISSING SECTION FIELDS

Removed required fields

Form Variables: ...
Context Variables: ...

Edit Section: Input: Valid Changes

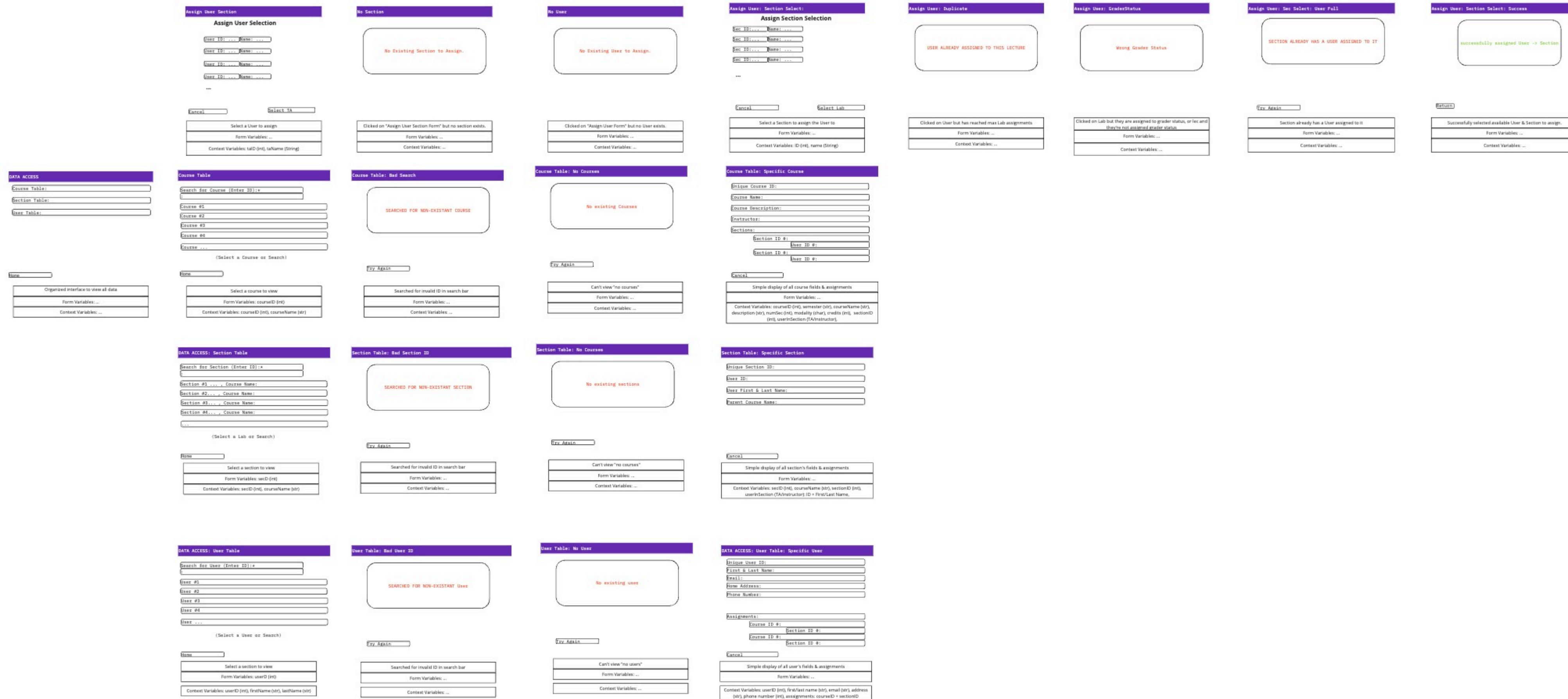
SUCCESSFULLY CHANGED INFORMATION

Valid input, completed all fields

Form Variables: ...
Context Variables: ...

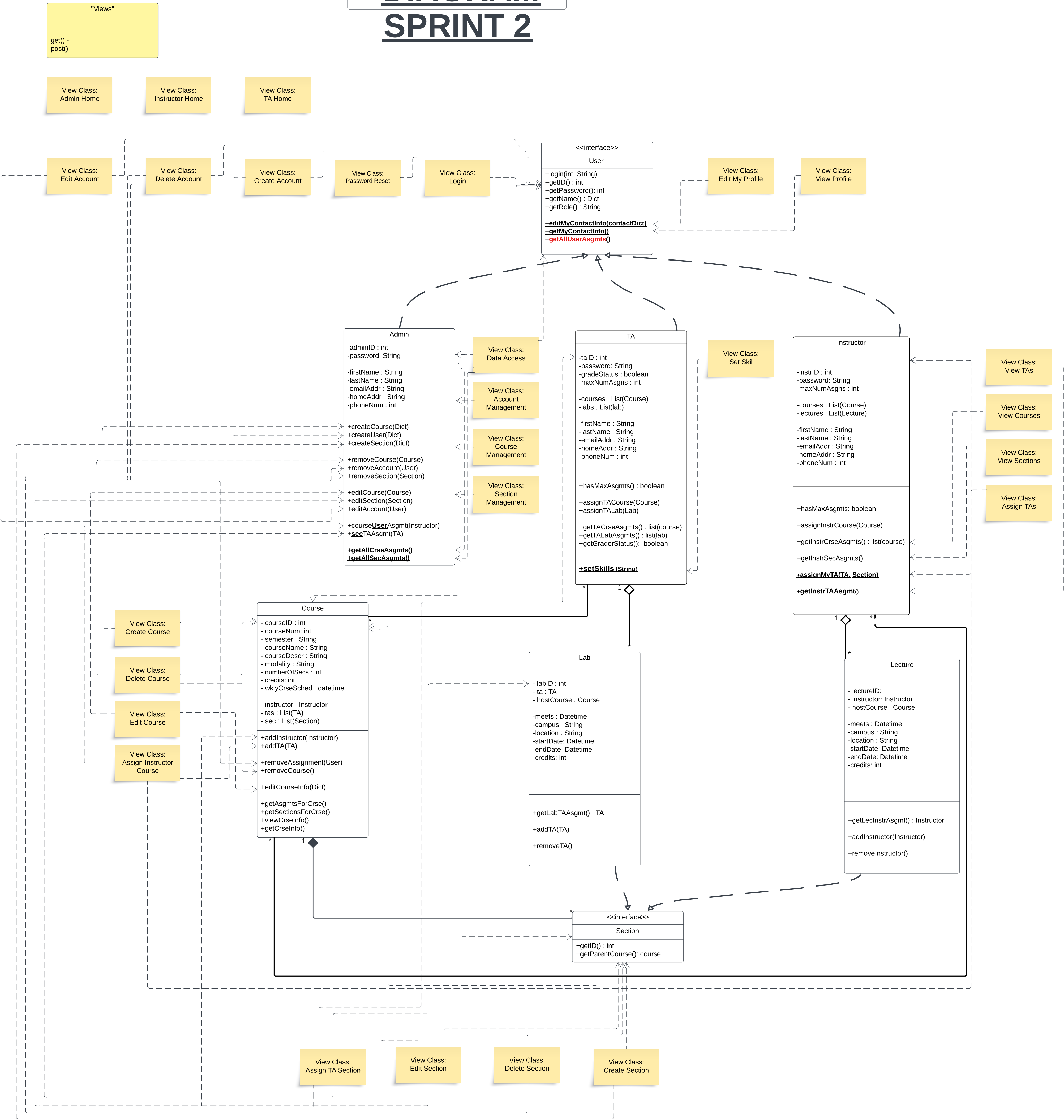
PROTOTYPES

SPRINT 2



SOFTWARE DIAGRAM

SPRINT 2



BEGIN USER

User.editMyContactInfo(contactDict)

Preconditions: Any assignments to the phone number key inside the dictionary must be 10 numbers, they user does not necessarily have to provide “all” fields.

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 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.getAllUserAsgmts()

- THIS MIGHT NEED TO GO IN THE “Admin” but I was thinking it might be useful for TA/instr since they need to view TAs and could just use this method and take what they need

Preconditions: There must be at least one active user.

Postconditions: no change to the host object or the respective sections, it returns all user’s, their roles, and their assignments (As a dictionary of dictionaries, see below). If no users, raises a messages indicating there is no users. Could be used by other “view” user methods.

Example:

```
{  
  TAs: ({“Role”: ...,“Username”: ...,“CourseAsgmts”: ..., “SecAsgmts”: ... }, {...}, {...}),  
  Instructors: (...),  
  Admins: (...)  
}
```

Side-Effects: ---

BEGIN ADMIN

Admin.courseUserAsgmt(activeUser, activeCourse)

- “Admin. courseInstrAsgmt (),” ALREADY EXISTS BUT MAYBE WE CAN USE THIS METHOD FOR “user (instructor/ta) tocourse ”

Preconditions: The user must exist in the system, the user has not exceeded their maximum course assignments, must be an user.

Postconditions: The user will receive the course assignment and the course will receive the user assignment. If max capacity instructor, nonexistent instructor/course provided, user provided not a user, then raise error message.
--

Side-Effects: ---

activeUser: in – out: the user is used as an input for the course’s user field and the user itself will be assigned to a course.
--

activeCourse: in – out: used for assigning user to
--

Admin.secUserAsgmt(activeUser, activeSec)???

- “Admin.labTAAsgmt(activeTA,” ALREADY EXISTS BUT MAYBE WE CAN USE THIS METHOD FOR “user (instructor/ta) to section (Lecture/Lab)”

Preconditions: The User must exist in the system, user can’t already be assigned to section, the user (if TA) has appropriate grader status, and the section doesn’t already have a user.

Postconditions: The user will receive the section assignment and the section will receive the user assignment. If nonexistent user/section, full section, wrong grader status then error message raised.
--

Side-Effects: ---

activeUser: in–out: the user is used as an input for the section’s user field and the user itself will be assigned to the section, will need to figure out if it is a TA/Instructor.
--

activeSec: in–out: the section is used as an input for the user field and the user itself will be assigned to the section.
--

Admin.getAllCrseAsgmt()

Preconditions: There must be at least one active course and user
--

Postconditions: no change to the host object or the respective course, returns all courses and their User assignments (As a list(dict)). If no active courses in the system, it raises an error message.
--

Example:

```
[  
{"CourseID": ..., "taAsgmts" : ... ,"instrAsgmts": ...}  
]
```

Side-Effects: ---

Admin.getAllSecAsgmt()

Preconditions: There must be at least one active section and user

Postconditions: no change to the host object or the respective section, returns sections and their User assignments (As a list(dict)). If no active courses in the system, it raises an error message.
--

Example:

```
[  
{"secID": ..., "asgmt" : ...}  
]
```

Side-Effects: ---

BEGIN TA

TA.setSkills (skills)

Preconditions: Skills must not be empty
Postconditions: set the TAs skills the arguments
Side-Effects: ---
Skills: in – used to set the TAs skill field to this string.

BEGIN INSTRUCTOR

Instructor.assignMyTA(activeTA, activeSec)

Preconditions: TA must exist in the system, TA must be assigned to one of my courses, the section to assign the TA to can't already have a TA.
Postconditions: Modifies the TA's state: it is assigned to a section, and the section is assigned to this TA. Raise error message if TA isn't assigned to this course, if already assigned to this section, or the lab is already full.
Side-Effects: ---
activeTA: in-out – 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 section's assignments and will be mutated to be assigned the sections.
activeSec: in-out: used to take the TA, the TA is also assigned to this specific section.

Instructor.getInstrTAsgmt()

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. List[Dictionaryes]
[{"taID": ..., "secID": ..., "courseID": }]
Side-Effects: ---

BEGIN COURSE

Done?

BEGIN LAB SECTIONS

Done?

BEGIN LECTURE
Done?