Pydra Sprint 2 Design Doc
Alidz Khaligian, Lucas Patron, Jiahui Yang, Anya Flickinger, Wolfgang Strubbe
CS 361

Insert more PBIs

As an Administrator, I want to assign instructors and to courses, so that they can properly be managed and connected.
Estimate: M
GIVEN An Administrator is logged into the system WHEN They select the Assign Course button on the dashboard THEN The system navigates them to the Assign Course page WHEN The administrator selects the Course Name field AND Enters a valid course name WHEN The administrator selects the Assignee Name field AND Enters a valid username WHEN The administrator selects assign THEN The user is assigned to a Course AND The users assigned courses are updated in the database

As a user, I want to edit my contact information, so that I can ensure the department and instructors have the most up-to-date information.
Estimate: M
GIVEN The User wants to up their contact information WHEN The User logs into the web application AND They select the Edit My Info button on the dashboard THEN The system navigates them to the Account Edit page WHEN The Instructor selects Field to change THEN The contact information field generates drop down option

WHEN The user selects the phone field
AND The user enters a new phone number in the Change to field
THEN The previous phone number is removed from the database
AND The new phone number is added to the database
WHEN The user selects the name field
AND The user enters a new name in the Change to field
THEN The previous name is removed from the database
AND The new name is added to the database
WHEN The user selects the email field
AND The user enters a new email in the Change to field
THEN The previous email is removed from the database
AND The new email is added to the database
WHEN The user selects the address field
AND The user enters a new address in the Change to field
THEN the previous address is removed from the database
AND The new address is added to the database

As a supervisor/administrator, I want to be able to create accounts so that I can access course and user information in my account (Create Account)

L: Create a table in the database for users, link the username to the password, write tests

GIVEN the username for the account is not already in use
WHEN I enter the username
AND a password
AND an email address
AND the account will be created and saved to the database

GIVEN the account username I provide is already in use
WHEN I enter the username
THEN a password
AND an email address
AND I will receive a message that the username is already in use, and the account will not be created

As an Administrator, I want to delete an account so that I can keep our security systems well maintained

M: Delete an account from the database

GIVEN The Administrator is logged into the system
AND There is at least one account that needs to be deleted
WHEN The Administrator navigates to the directory page
AND Selects the option to delete the account
THEN The system should delete the account and its associated data
AND The system should display a success message to confirm the deletion or update the account list reducing the number of accounts by one in the database

As a user, I must be able to log in so that I can view my own information and access the tools of my role

L: We must set up a database to store the encrypted passwords safely, as well as design a login screen that interacts with that database

GIVEN I log into my account
WHEN I enter my correct login information
THEN I am brought to my home page
AND I have access to the tools of my role, no more and no less
AND I can view all of my account information

GIVEN I am on the login screen
AND I enter an incorrect username or password
THEN I will receive an "Invalid username or password" error message
WHEN I attempt to use the incorrect password

As an Administrator, I want to be able to create new courses so that they are saved in the database

M: UI for adding new course and entering info, add info to database, tests

GIVEN I have the Administrator right to access this feature
WHEN I enter a course name not already in use
AND valid course time
THEN The course information will be saved in the database
AND a success message will appear on the screen

GIVEN I have the Administrator right to access this feature
WHEN I enter a course name already in use
THEN The course information will not be saved in the database
AND a message that this course already exists will appear

GIVEN I have the Administrator right to access this feature
WHEN I enter a course name not already in use
AND an incorrectly formatted course time
THEN The course information will not be saved in the database
AND an error message saying time is formatted incorrectly will appear on screen

---

As an Administrator, I want to be able to add sections to courses, so that they are linked to a specific course

M: UI for adding new section, add info to database, tests

GIVEN I have the Administrator right to access this feature
WHEN I choose to add a section
AND Enter a course which exists
AND a unique section ID for the course
AND submit
THEN the section will be added to the course

GIVEN I have the Administrator right to access this feature
WHEN I choose to add a section
AND Enter a course which exists
AND a section ID already in use for the course
AND submit
THEN a message will say that this section id is already in use

GIVEN I have the Administrator right to access this feature
WHEN I choose to add a section
AND Enter a course which does not exists
AND a section ID
AND submit
THEN a message will say that the course I am trying to add to does not exist

As a TA, I want to be able to add my skills, so that they can be taken into consideration when being assigned to courses

M: UI for adding skill, add skill model to database, tests

GIVEN I am a TA
AND Enter a new skill
AND submit
THEN the the skill will be added to my skills
AND a success message will display

GIVEN I am not a ta
WHEN I choose to add a skill
THEN an error message will display

GIVEN I am a TA
AND Enter a duplicate skill
AND submit
THEN the skill will not be added
AND an error message will display

As an Administrator, I want to assign TAs to courses and lab sections so that I can manage which courses and labs TAs are assigned (Assign TAs to courses)

Estimate: L - We need to create the unit tests, implementation, and acceptance tests for an assign section function, as well as write the html

GIVEN The Instructor needs to assign a TA to a lab specific lab section WHEN
The Instructor is logged into the web application
AND They are on the assign section page
AND they enter a correct course, section, and TA

WHEN they click the "submit" button

THEN the TA is assigned to the correct section

GIVEN The Instructor needs to assign a TA to a lab specific lab section WHEN
The Instructor is logged into the web application
AND They are on the assign section page
AND they enter a correct course and section, but an incorrect TA

THEN the TA is not assigned and an error is displayed to the screen

GIVEN The Instructor needs to assign a TA to a lab specific lab section WHEN
The Instructor is logged into the web application
AND They are on the assign section page
AND they enter a correct course and TA, but an incorrect section

THEN the TA is not assigned and an error is displayed to the screen

GIVEN The Instructor needs to assign a TA to a lab specific lab section WHEN
The Instructor is logged into the web application
AND They are on the assign section page
AND they enter a correct TA and section, but an incorrect course

THEN the TA is not assigned and an error is displayed to the screen

GIVEN The Instructor needs to assign a TA to a lab specific lab section WHEN
The Instructor is logged into the web application
AND They are on the assign section page
AND they enter a correct course, section, and TA

AND the TA is already assigned to the section

THEN the TA is not assigned and an error is displayed to the screen

GIVEN The Instructor needs to assign a TA to a lab specific lab section WHEN
The Instructor is logged into the web application
AND They are on the assign section page
AND they enter a correct course, section, and TA

AND the section does not belong to the course

THEN the TA is not assigned and an error is displayed to the screen

---

As an Administrator, I should be able to access a page that displays a list of all TAs skills registered on the platform, so that I know what TA's are qualified to teach certain courses.
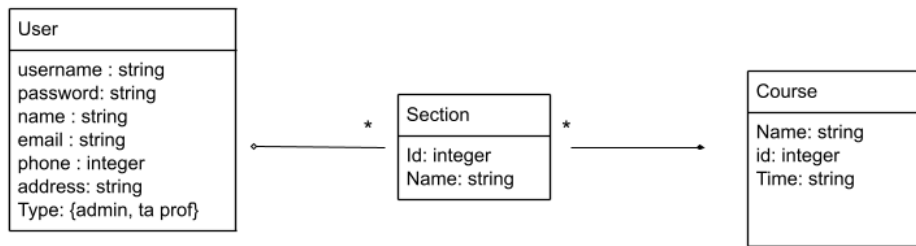
Estimate: M

GIVEN that I am logged in as an Administrator

WHEN I navigate to the "view skills" section

THEN I should be directed to a page that allows a search of registered TAs and their respective skills.

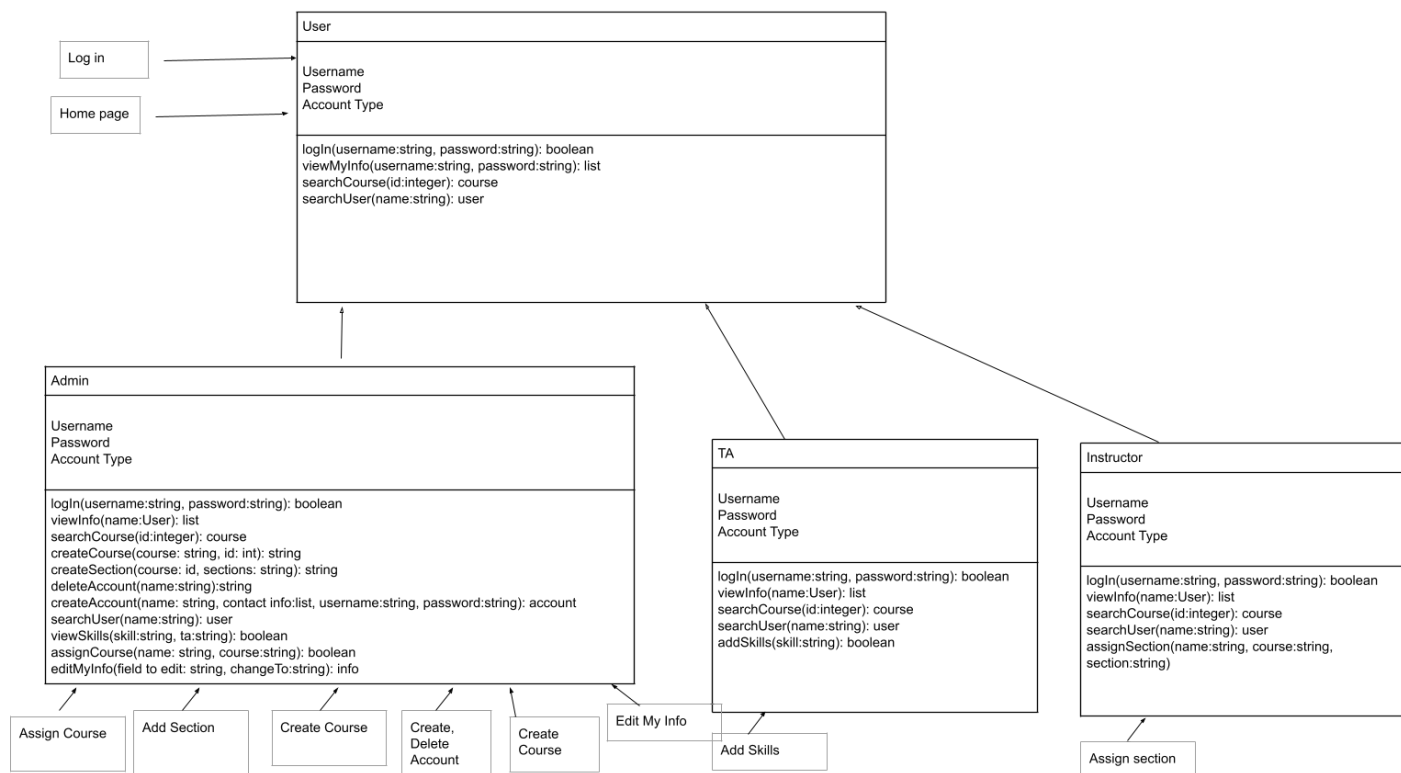GIVEN that I am on the "view skills" page as an Administrator

WHEN I search a registered TA

THEN the TA's skills should be prominently displayed for easy assessment.

Database UML

User
- username : string
- password: string
- name : string
- email : string
- phone : integer
- address: string
- Type: {admin, ta prof}

Section
- Id: integer
- Name: string

Course
- Name: string
- id: integer
- Time: string

Class UML

Log in

Home page

User
- Username
- Password
- Account Type

logIn(username:string, password:string): boolean
viewMyInfo(username:string, password:string): list
searchCourse(id:integer): course
searchUser(name:string): user

Admin
- Username
- Password
- Account Type

logIn(username:string, password:string): boolean
viewInfo(name:User): list
searchCourse(id:integer): course
createCourse(course: string, id: int): string
createSection(course: id, sections: string): string
deleteAccount(name:string):string
createAccount(name: string, contact info:list, username:string, password:string): account
searchUser(name:string): user
viewSkills(skill:string, ta:string): boolean
assignCourse(name: string, course:string): boolean
editMyInfo(field to edit: string, changeTo:string): info

TA
- Username
- Password
- Account Type

logIn(username:string, password:string): boolean
viewInfo(name:User): list
searchCourse(id:integer): course
searchUser(name:string): user
addSkills(skill:string): boolean

Instructor
- Username
- Password
- Account Type

logIn(username:string, password:string): boolean
viewInfo(name:User): list
searchCourse(id:integer): course
searchUser(name:string): user
assignSection(name:string, course:string, section:string)

Assign Course

Add Section

Create Course

Create, Delete Account

Create Course

Edit My Info

Add Skills

Assign section

Grey Boxes indicate pages which require directed classes

Admin.logIn(username, password)

Preconditions: user table exists in database, two arguments are provided, username must be string and password must be string, else throw illegal argument exception
Postcondition 1: Username and password are linked to one user in the database, return true
Side Effect 1: If user was not logged in, the user is now logged in and a print statement "log in successful"
Side Effect 1.2: If user is already logged in, print statement "already logged in"
Postcondition 2:Username or password are not part of a user in the database,, return false
Side-effects: Print statement "unsuccessful log in"
username (input): this username should be linked to the account user wants to log into password(input): password should be linked to username to admit log in

Admin.viewMyInfo(name)

Preconditions: User table exists in database
Postconditions:user is logged in, return all my user information from database
Side-effects: none
name (input): provided during successful log in

Admin.searchUser(user)

Preconditions: Usertable exists in database, user is a string
Postconditions: Given user is linked to a course, display user

Postconditions 2: user is not linked to account, throw exception
Side-effects: none
user (in): name of user we want to find

Admin.searchCourse(id)

Preconditions: Course table exists in database, id is an int
Postconditions: Given id is linked to a course, return true
Postconditions 2: id is not linked to course, return false
Side-effects: none
id (in): confirms that course exists

Admin.createCourse(name, id)

Preconditions: Course table exists in the database, name is a string, id is an integer
Postconditions: a new course is added to the database table of courses
Side-effects: none
name (in-out): the name of the course we want to create, will be saved to the course id(in-out): id reference to course being created

Admin.createAccount(username, password,type)

Preconditions: type,name, and password are strings, user table exists
Postconditions: an account containing the provided arguments is added to the user table, an object of type is created

Postcondition 2: name provided overlaps with another account, throw exception, account not created
Side-effects: none
name(in-out): name provided for owner of account being created which will be used for output account password(in-out): provided password which will be used for output account type(in): type of user account being created, supervisor, ta, prof

Admin.createSection(course, id)

Preconditions: Course table exists in the database, section table exists in database
Postcondition 1: a new section is added to the database table of sections and is linked to course whose id matches provided course
PostCondition 2: Course id provided doesn't exist, throw exception
Postcondition 3: Section id provided is already in use for section linked to provided course id, throw exception
Side-effects: none
course (in-out):id number of course we want to add section to id(in-out): id reference to section being added to course

Admin.editMyInfo(type, newEntry)

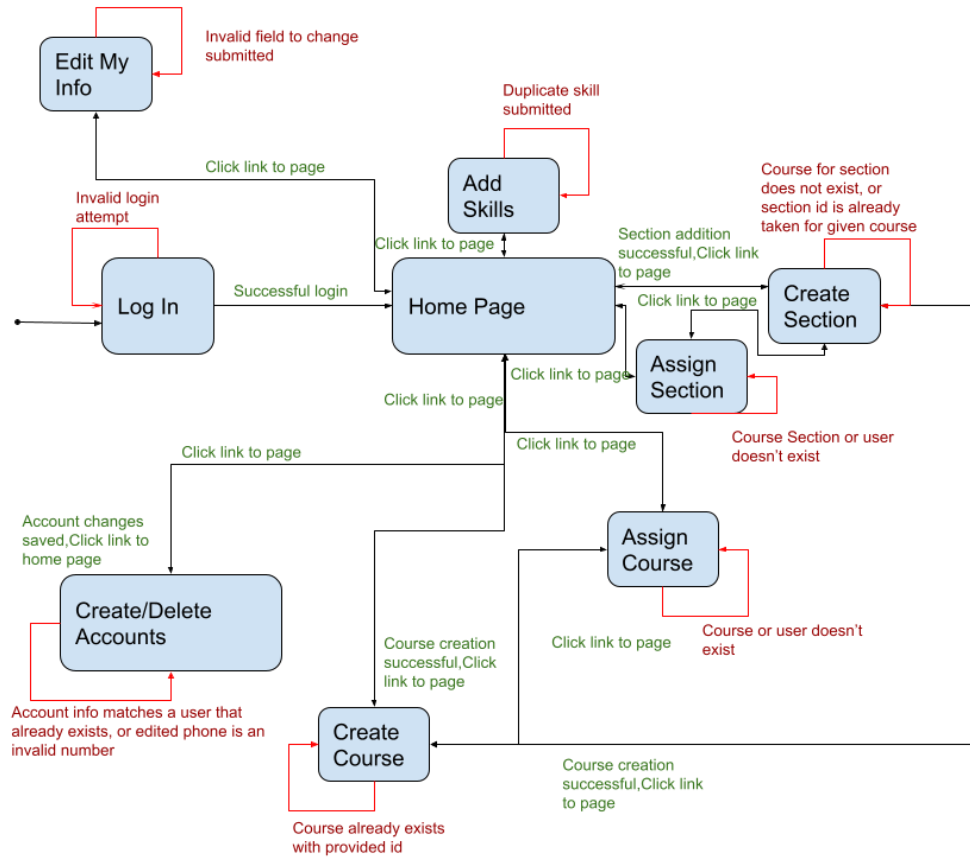| |
|---|
| Preconditions: User table exists in database,type and newEntry must be string |
| Postconditions: type provided matches one of the fields phone, name, email, or address, value of given field is changed to newEntry and is returned |
| Side-effects: none |
| type(in): Specifies which field of contact info will be changed
newEntry(in-out): updated field contents provided by user |

Admin.assignCourse(assignee, course)

Preconditions: prof must be string and course must be string, user table and course table exist in database, must be able to view ta skill info
Postconditions: return true, false if course or assignee does not exist
Side-effects: course is now linked to assignee and vice versa
assignee (in-out): assignee that will be assigned to course course(in-out): course that will be linked to prof

Instructor.assignSection(course, section, assignee)

Preconditions: assignee, section, and course must be a string, user table, course and section table exist in database
Postconditions: return true, false if any of the fields provided are not in db
Side-effects: section is now linked to ta
assignee (in-out): ta that will be assigned to section course(in-out): no change, only used to verify the section exists within the course section(in-out): specific section that will be linked to ta

Instructor.addSkills(ta, skill)

Preconditions: ta must be string, skill must be string, user table exists in db with a skill s category
Postconditions: return true, false if skill is duplicate
Side-effects: skill is added to user skill list
ta(in-out): ta who is adding to their skills skill(in-out): skill to be added to list

Edit My Info

Invalid field to change submitted

Duplicate skill submitted

Add Skills

Course for section does not exist, or section id is already taken for given course

Click link to page

Invalid login attempt

Section addition successful,Click link to page

Click link to page

Log In

Successful login

Home Page

Click link to page

Create Section

Click link to page

Assign Section

Click link to page

Course Section or user doesn't exist

Click link to page

Click link to page

Assign Course

Account changes saved,Click link to home page

Create/Delete Accounts

Course creation successful,Click link to page

Click link to page

Course or user doesn't exist

Account info matches a user that already exists, or edited phone is an invalid number

Create Course

Course creation successful,Click link to page

Course already exists with provided id

UW-Milwaukee TA Scheduling Application- Login

Username:

Password::

Log in

UW-Milwaukee TA Scheduling Application- Home Page

Account Info:
Name: Bob Password**** Phone, Email, Address

Create Course

Add Section

Create, Delete, Accounts

Edit my Info

Assign Section

Assign Course

UW-Milwaukee TA Scheduling Application- Create Course

Create Course

Course Name

Course ID

Time

Create Course

Add a section

All Courses List here

Go Home

Assign Course

UW-Milwaukee TA Scheduling Application- Create, Delete Account

Create

Username

Password

User Type

Create Account

Delete

Search User →

All Users list here

Go Home

UW-Milwaukee TA Scheduling Application- Edit My Info

Edit Information

Field editing

Change to

Save Changes

Go Home

UW-Milwaukee TA Scheduling Application- Add Skills

My Name

New Skill

Add skill

All Skills List here

Go Home

UW-Milwaukee TA Scheduling Application- Add Section

Add Section

Course ID

Section ID

Add Section

All Sections List here

Go Home

Assign section

UW-Milwaukee TA Scheduling Application- Assign Section

Assign Section

Course Name

Section Name

Assignee Name

Saven

Create Section

Go Home

Assign Course

Course Name

Assignee Name

Assignee skills: science, math

Assign

Go Home   Create Course