Phase 3

Team Leader: Fue

Team Members: Devin, Charlie, Leo, Hongtao, Fue, Bennet

Date: 4/14/2020

[This page is intentionally left blank]

https://github.com/crolfs001/CMC.git

Table of Contents

Meeting Minutes

April 3

- Minute Taker: Hongtao
- Attendees: Devin, Charles, Fue, Hongtao
- Discuss the content of phase 2, check everyone’s task.
- Assign tasks of phase 3
- Meet again on April 6

April 6

- Minute taker : Leonardo Yang
- Attendees: Devin, Fue, Hongtao, Bennet, Leonardo
- Discussed the phase 3 progress.
- Split up entity and controller classes to do
- Discussed How to set up J-Unit classes
- Split up use cases to do.
- Will meet again on wednesday April 8 at 4:00 pm

Detailed Task Decomposition

- Devin:
    - Created system tests for login, logout, top 5 recommended schools, view list of users functionalities, create user functionalities, and delete user functionalities.
    - Coded view list of users functionality.
    - Coded database methods(createUser(), deleteUser()).
    - Created unit tests for Account Controller (testlogin(), testLogout(), testViewListofUsers()), DBController(testFindByUsername(), testcreateUser(), testdeleteUser(), testSetLoginStatus(), testGetTopFiveClosestSchools()), User(testSetLoginStatus()), UniversityController(testGetTopFiveClosestSchools).
    - Helped others resolve errors
- Charlie: created all stub test methods including test suite.
    - Attached view and edit profile to database
    - Started and completed activating and deactivating account use case and attached to database
    - Created these tests
        - AccountController: testUpdateProfile(), testActivateAccount(), testDeactivateAccount()

        - DBController: testUpdateUser()
        - Functionality test: testEditProfile(), testActivateAndDeactivateAccount()
- Fue:
- Hongtao: coded UserSchool class: addSchool(), removeSchool()

- Bennet:
- Leo:

  Functionality test : ViewSavedSchoolsTest, ViewSchoolInfoTest,

  Created UserInteractionTest: testGetUser(), testSetUser(), Login(),

  Created UniversityControllerTest: testShowSchools().

Testing

Charlie’s Tests

- WBT tests
  - AccountController: testActivateAccount()
  - AccountController: testDeactivateAccount()
- BBT tests
  - DBController: testUpdateUser()
  - AccountController: testUpdateProfile()
- Simple tests
  - testEditProfile()
  - testActivateAndDeactivateAccount()

Devin’s Tests

- WBT
  - DBControllerTest: testCreateUser
  - DBControllerTest: testDeleteUser
- BBT
  - UserTest: testSetLoginStatus
  - DBControllerTest: testSetLoginStatus

Leonardo’s Tests

- WBT

  UserInteractionTest: testGetUser()

  UniversityControllerTest: testShowSchools()

- BBT

  UserInteractionTest: Login(), testSetUser(),

Fue’s Tests

- WBT
  - UserSchoolTest: removeSchool()
  - StudentInteractionTest: promptRemove()
- BBT
  - DBControllerTest: removeSavedSchool()
  - UniversityControllerTest: removeSavedSchool()

Classes

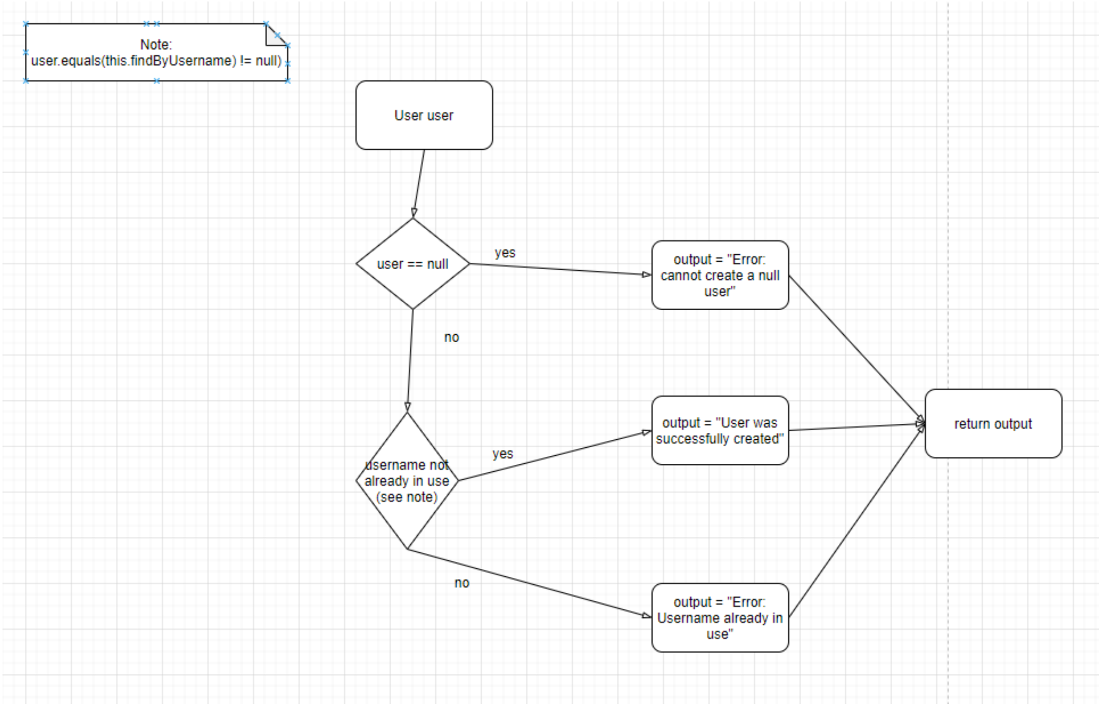
DBController: constructs a DBController object

- (Charlie) public void updateUser(User user): BBT
  - Equivalence classes: User- only need to create one user to validate that the database is updated properly
  - Test Cases: user = findByUsername(“juser”)
  - Test Case Table

Changes	**Expected**	**Actual**
Change first name	“Bill”	user.getFirstName()

Change last name	“Hader”	user.getLastName()
Change password	“Password”	user.getPassword()

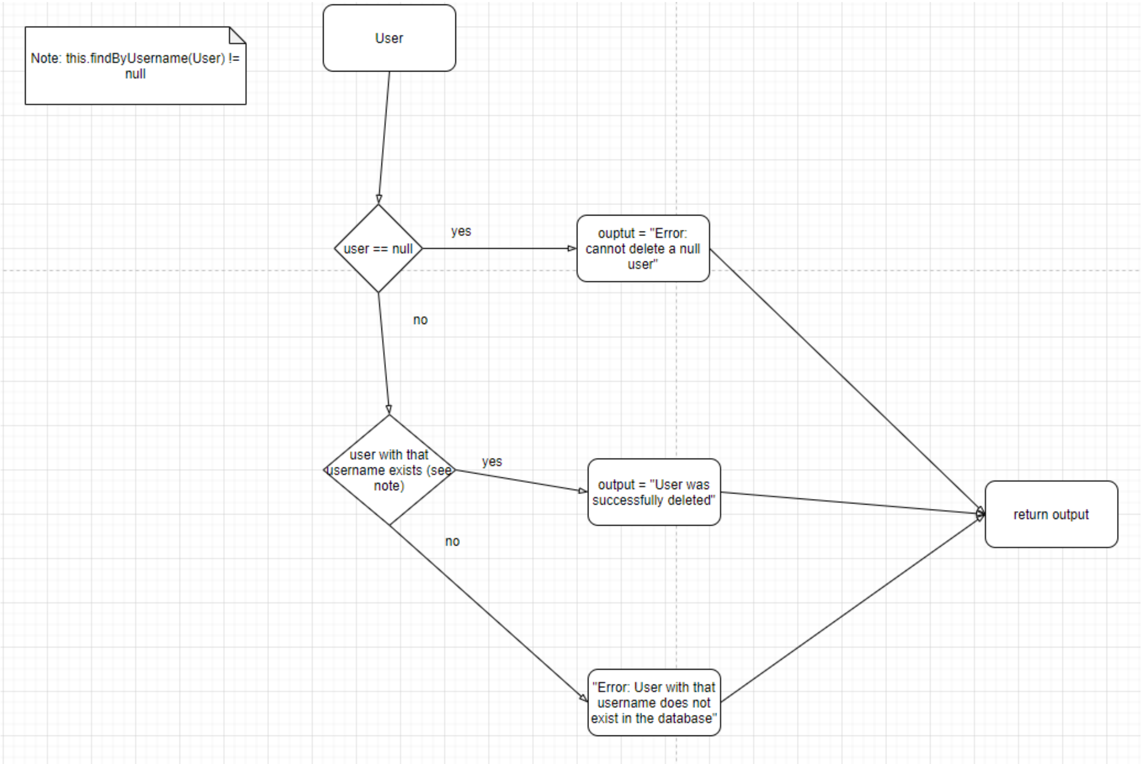
- Public void DBController()
- (Devin) Public String createUser(User)


Note:
user.equals(this.findByUsername) != null)
User user
user == null
yes
no
output = "Error: cannot create a null user"
username not already in use (see note)
yes
no
output = "User was successfully created"
output = "Error: Username already in use"
return output

Path	Expected	Actual
User == null	“Error: cannot create a null user”	database.createUser(User)
Username is not already in use	“User was successfully created”	database.createUser(User)
else	“Error: username already in use”	database.createUser(User)

- ■

- ○ (Devin) Public String deleteUser(User)


Note: this.findByUsername(User) != null
User
user == null
yes
ouptut = "Error: cannot delete a null user"
no
user with that username exists (see note)
yes
output = "User was successfully deleted"
no
return output
"Error: User with that username does not exist in the database"

- ■

- ■

Path	Expected	Actual
User == null	“Error: cannot delete a null user”	this.deleteUser(User)
User with that username was not found	“User was successfully deleted”	this.deleteUser(User)
else	“Error: User with that username does not exist in the database”	this.deleteUser(User)

- ■

- ○ Public User findByUsername(String)

- Private int getMaxApplicants()
- Private double getMaxExpenses()
- Private int getMaxNumStudents()
- Private int getMinApplicants()
- Private double get MinExpenses()
- Private int getMinNumStudents()
- Public University getSchool(String)
- Public ArrayList<University> getSchoolList()
- Public getTopFiveClosestSchool(String)
- Public ArrayList<User> getUserList()
- Public String removeSavedSchool(USer,University)
- (Devin) Public boolean setLoginStatus(User, boolean)
    - Equivalence Classes: inputs a User object whose login status we are changing and a boolean that tells us what the new status is.
    - Two possibilities, either we change the status to true (logging in) or false (logging out). These are the only two inputs possible when using black box testing. Output should be the boolean true if the loginstatus was successfully changed.
    -

Changes	Expected	Actual
Input true	**true**	this.setLoginStatus(User, **true**)
Input false	**true**	this.setLoginStatus(User, **false**)

    - ■

- Public void showSchoolList()
- Public String updateSavedUniversityList(User, University)
- Public void updateUser(User)

Driver(Do not need to test driver)

- Private void accountStatusTester(DBController, AccountController, UserInteraction)
- Public void addSchoolTester(DBController, AccountController, UserInteraction)
- Public void loginTester(DBController, AccountController, UserInteraction)
- Public void logoutTester(DBController, AccountController, UserInteraction)
- Public void topFiveRecommendedSchoolTester(DBController, AccountCOntroller, UserInteraction)
- Public void viewAndEditProfileTester(DBController, AccountController, UserInteraction)
- Public void ViewSchoolSearchResults()
- Public void ViewStudentInfoTester()

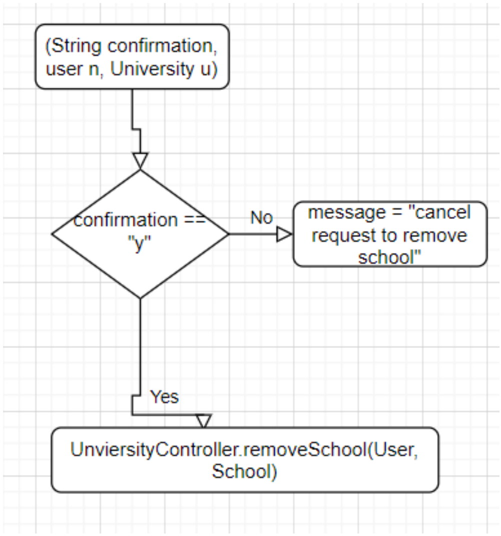
Student

StudentFunctionalityController

- Public void StudentFunctionalityController()
- Public String getSchool()
- Public void pressSaveButton(User, String)

StudentInteraction

- ○ Public Student cancelChange(Student)
- ○ Public ArrayList<String> displayTopFive(String)
- ○ Public String getSchool()
- ○ Public void makeEditAble()
- ○ Public void openViewSavedSchools()
- ○ Public void pressSaveButton(String)
- ○ Public String promptRemove(String, User, University)


(String confirmation, user n, University u)
confirmation == "y"
No
message = "cancel request to remove school"
Yes
UnviersityController.removeSchool(User, School)

Path	Expected	Actual
confirmation == “y”	“This school have been remove”	UniversityController.removeSchool(user, school)
confirmation != “y”	“Cancel request to remove school”	Invalid String

- Public void resetSearch()
- Public List<String> searchSchool()
- Public void viewProfilPage()

University

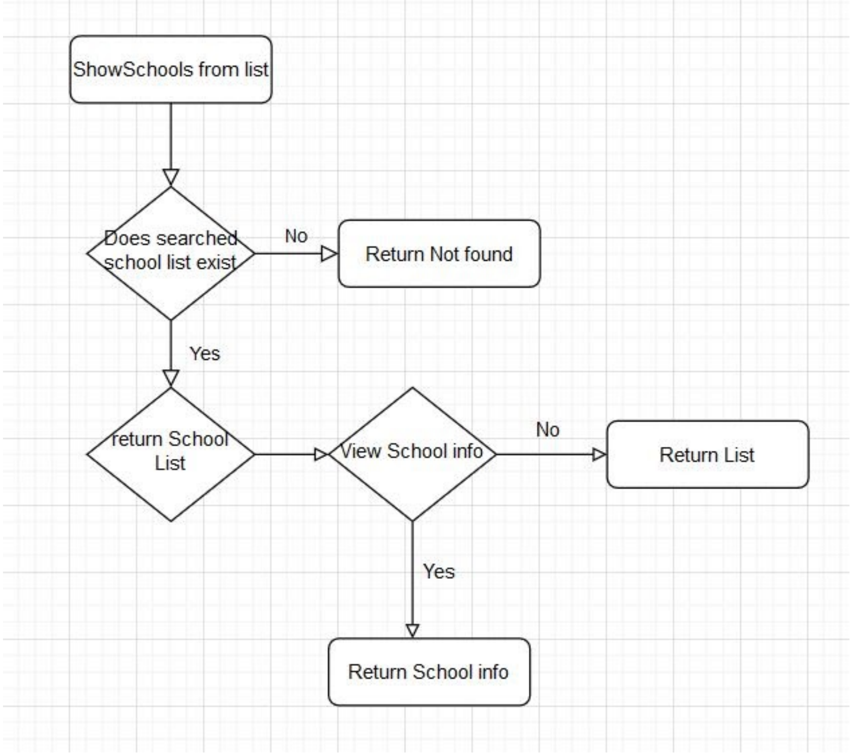
- Public void University(String, String, String, String, int, double, double, double, double, double, int, double, double, int, int, int)
- Public void addEmphasis(String)
- Public int getAcademcsScale()
- Public String getControl()
- Public ArrayList<String> getEmphases()
- Public double getExpenses()
- Public String getLocation()
- Public String getName()
- Public int getNumApplications()
- Public int getNumStudents()
- Public double getPercentAdmitted()
- Public double getPercentEnrolled()
- Public double getPercentFemal()
- Public double getPercentFinancialAid()
- Public int getQualityOfLifeScale()
- Public double getSatMath()
- Public double getSatVerbal()

- Public int getSocialScale()
- Public String getState()
- Public void setAcademicsScale(int)
- Public void setcontrol(String)
- Public void setExpenses(int)
- Public void setLocation(String)
- Public void setName(String)
- Public void setNumApplications(int)
- Public void setNumStudents(int)
- Public void setPercentAdmitted(int)
- Public void setPercentEnrolled(int)
- Public void setPercentFemal(int)
- Public void setPercentFInancialAid(int)
- Public void setQualityOfLifeScale(int)
- Public void setSatMath(int)
- Public void setSatVerbal(int)
- Public void setSocialScale(int)
- Public void setState(String)

UniversityController

- Public void UniversityController()
- Public void UniversityController(DBController)
- Public University getSchoolName(String)

- Public ArrayList<String> getTopFiveCLosestSchool(String)
- Public String removeSavedSchool(User, University)
- Public ArrayList<University> searchSchool()
- Public void showSchools(List<University>)

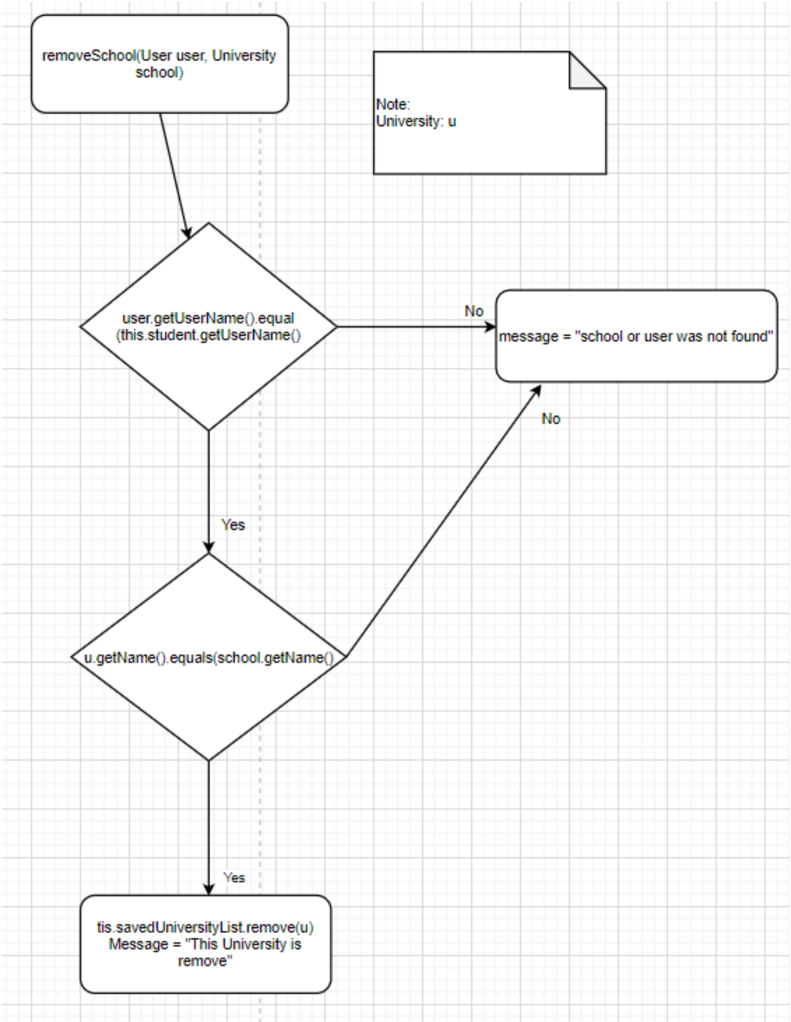


Path	Expected	Actual
Searched list does not exist	“List does not exist”	null
Return school list	“SJU”	“SJU”

UserSchool

- Public void UserSchool(User)

- Public String addSchool(User, University)
- Public String getLastSchool()
- Public String getUserName()
- Public String removeSchool(User, University)

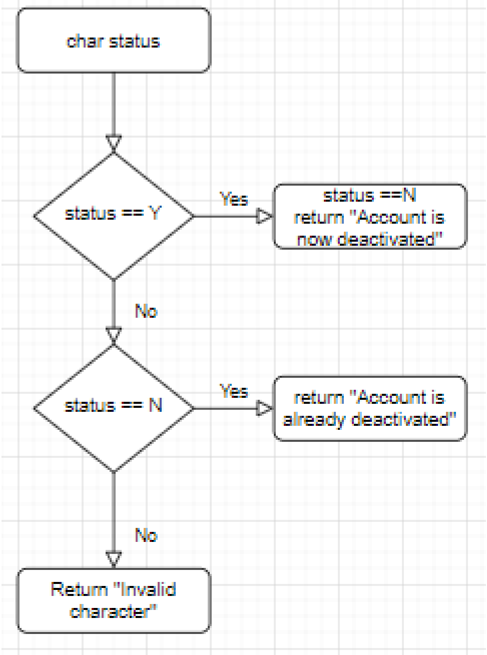

removeSchool(User user, University school)
Note:
University: u
user.getUserName().equal (this.student.getUserName()
No
message = "school or user was not found"
No
Yes
u.getName().equals(school.getName()
Yes
tis.savedUniversityList.remove(u) Message = "This University is remove"

path	Expected	Actual
user.getUserName().equal (this.student.getUserName() && u.getName().equals(school.getName()	“This university is remove	this.savedUniversityList.remove(u)

user.getUserName() != equal (this.student.getUserName()	“School or user was not found”	

- Public void showSaveSchoolList()

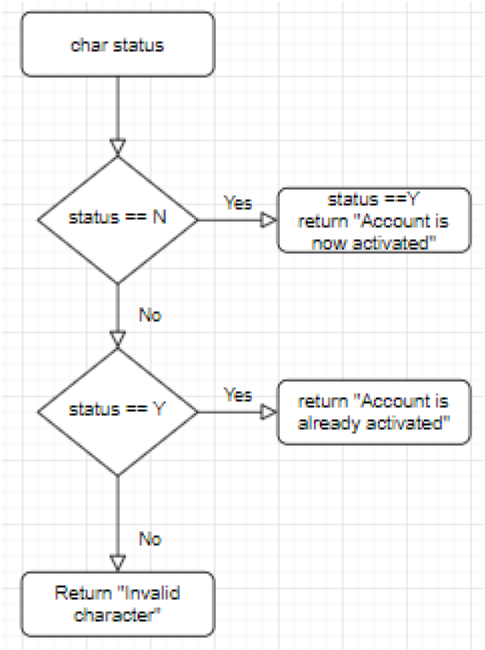
AccountController

- Public void AccountController(DBController)
- (Charlie) Public String deactivateAccount(String) WBT
- 
char status
status == Y
Yes
status ==N
return "Account is now deactivated"
No
status == N
Yes
return "Account is already deactivated"
No
Return "Invalid character"

Path	Expected	Actual
Status == Y	Status = N, return “Account is now deactivated”	user.getStatus(), “Account is now deactivated”
Status == N	Status = N, return “Account is already deactivated”	user.getStatus(), “Account is already deactivated”

Else	Invalid char	Invalid char

- Public String createUser(User)
- (Charlie) Public String activateAccount(String) WBT


char status
status == N
Yes
status ==Y
return "Account is now activated"
No
status == Y
Yes
return "Account is already activated"
No
Return "Invalid character"

Path	Expected	Actual
Status == N	Status = Y, return “Account is now activated”	user.getStatus(), “Account is now activated”
Status == Y	Status = Y, return “Account is already activated”	user.getStatus(), “Account is already activated”
Else	Invalid char	Invalid char

- Public User findByUsername(String)

- Public DBController getDatabase()
- Public User getUser()
- Public String login(String String)
- Public String logout()
- Public void setDatabase(DBController)
- (Charlie) Public String updateProfile(String, String, String, String): BBT
  - Equivalence classes: takes a User
  - Test Cases: test changing first name, last name, and password for a user
  - Test Case Table

Changes	**Expected**	**Actual**
Change first name	“firstName”	user.getFirstName()
Change last name	“lastName”	user.getLastName()
Change password	“password”	user.getPassword()

- Public ArrayList<User> viewListofUsers()

User

- Public void User(String, String, String, String, char)
- Public String getFirstName()
- Public String getLastName()
- Public boolean getLoginStatus()

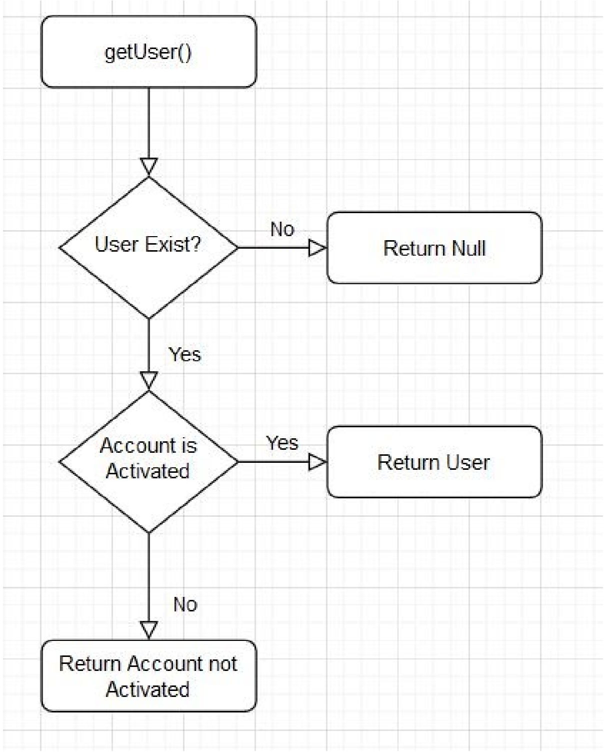
- Public String getPassword()
- Public char getStatus()
- Public char getType()
- Public String getUserName()
- Public void setFirstName(String)
- Public void setLastName(String)
- (Devin) Public void setLoginStatus(boolean)
    - Equivalence class: takes a boolean as the login status
    - Test the two different possibilities for true (user is logging in) and false (user is logging out). Ouptut is a boolean of whether it was successfully changed to the correct status
    -

changes	expected	actual
Input true	**true**	User.getLoginStatus()
Input false	**true**	User.getLoginStatus()

    -
- Public void setPassword(String)
- Public void setStatus(char)
- Public void setType(char)
- Public void setUserName(String)

UserInteraction

Public user getUser()



Path	Expected	Actual
User does not exist	null	null
Account is activated	user	UserInteraction.getUser()

public void setUser(User user)

Equivalence Class: User newName

Test cases: testSetUser()

Changes	Expected	Actual
Changed newName variable to LYang001	LYang001	LYang001

Changed newName variable to FVue001	FVue001	FVue001
Changed newName variable to CRolfs001	CRolfs001	CRolfs001

public AccountController getAccountController()

public void setAccountController(AccountController accountController)

public String login(String username, String password)

Equivalence class: newUser.getUsername, newUser.getPassword

Test case: testLogin()

Changes	Expected	Actual
Change UserName to new name	Failed login	Failed login
Change newUser to newname	Failed login	Failed login
Change Username to user1	Failed login	Failed login

public String logout()

- public String activateAccount(String username) basic test
    - Created ActivateAndDeactivateTest class
- public String deactivateAccount(String username)
    - Created ActivateAndDeactivateTest class

public UniversityController getUniversityController()

```
public void setUniversityController(UniversityController universityController)
public ArrayList<String> getTopFiveClosestSchool(String universityName)
public ArrayList<User> viewListofUsers()
public String createUser(User user)
```