**College of Saint Benedict & Saint John’s University**

**Computer Science Department**

**CMC.com**

**Phase 3**

**Team Name: Holey Fan Club**

**Team Leader: Ben Xiong**

**Team Members: Abdi Jibril, Noah Hoberg, Moua Lee, Tommy Torbert**

**Date 4/10/19**

**[This page is intentionally left blank]**

**Database Controller**

**Central hub of all database accessing methods, methods are given by Imad**

**Logon Controller**

**Logout – logs a user out if they are already logged in**

**Login – logs a user in if they are found in the database by username, and if their password is correct, otherwise it throws illegalArgumentExceptions**

**AdminFunctionalityController**

**getListOfUsers – returns a 2d array of all users from the database**

**Public java.lang.String[][] gitListOfUsers()**

**deleteUniversity- deletes a university from the database by name**

**Public int deleteUniversity(String uniName)**

**resetUsersPassword – calls editUser in database controller with a username and new password**

**Public void resetUsersPassword(String username, String newPassword)**

**deleteUser – calls delete user from the database controller with a username**

**Public int deleteUser(String username)**

**editUniversity – edit university from the database controller with a given university**

**Public void editUniversity(String username)**

**addNewUser – calls add new user from the database controller, and if the user’s username is not unique throws an exception**

**Public int addNewUser(User user)**

**addNewUniversity – calls add new university from the database controller given a university, if the universities’ name is unique, otherwise throwing an exception**

**Public void addNewUniversity(University university)**

**editUser – calls edit user from the database library, if the account is found in the database, otherwise throws an exception**

**Public void editUser(String user)**

**deactivateUser – calls edit User from the database**

**Public void deactivateUser(String name)**

**UserFunctionalityController**

**LogIn- sets a Boolean field to the given parameter**

public void login(String username, String password

**logout – sets the Boolean field to false**

public void logout()

**setCurrentUser – sets the “currentUser” to the user that is passed (currently logged in)**

**edit profile – edits the profile to the info passed with the parameter User’s info**

public int editProfile(User user)

**getUniversity – calls the database getUniversity method**

public void getUniversity()

**saveSchool – calls the database controller save school to user method with a given username and password, if the first is not in the database, or the first is but the second is incorrect, throws an exception**

public int saveSchool(String username, String uniName)

**Search Controller**

**searchResults – prints the results of searchUniversities (from the instance field schoolClones)**

**Public void searchResults()**

**searchUniversities – searches the given university through multiple if statements, removing incorrect universities from the list and returning false if no university is found**

**Public void searchUniversities(String name, String state, String location, String control, int numOfStudents, double percentFemale, double SATVerbal, double satMath, double expenses, double finAid, int numApplicants, double perAdmitted, double perEnrolled, int academicScale, int socialScale, int lifeScale)**

**AdminUI**

**Interaction layer that the user actually uses**

**addNewUniversity – calls the admin functionality controller add new university**

**editUniversity – calls the admin functionality controller edit university**

**viewListOfUsers – calls the admin functionality controller view list of users**

**addNewUser – calls the admin functionality controller add new user**

**deleteUser – calls the admin functionality controller delete user**

**deactivateUser – calls the admin functionality controller deactivate user**

**changePassword – calls the admin functionality controller change password**

**deleteUniversity – calls the admin functionality controller delete university**

**login – calls the admin functionality controller login**

**logout – calls the admin functionality controller logout**

**public void confirmMessage()**

**public void failMessage()**

**public void viewListOfUsers()**

**public void addNewUser(User user)**

**public void deleteUser(String username)**

**public void changePassword(String username, String password)**

**public void deleteUniversity(String uniName)**

**public void login(String username, String password)**

**UserUI**

**Apparently got overlooked and is under implemented**

**public void displayUsersList()**

**public void returnList()**

**public void searchForUser(String username)**

**public void saveSchool(String username, String uniName)**

**public int editProfile(User user)**

**public void login(String username, String password)**

**public void logout()**

**Account2**

**Public account2(String username, String firstName, String lastName, String password, Char type, Char status)**

**GetUserName – simple accessor method for a field**

**Public String getUsername()**

**SetUserName – simple mutator method that sets a field**

**Public void setUserName(String name)**

**GetFirstName – simple accessor method for a field**

**Public String getFirstName()**

**SetFirstName – simple mutator method that sets a field**

**Public void setFirstName(String firstName)**

**GetPassword – simple accessor method for a field**

**Public String getPassword()**

**SetPassword – simple mutator method that sets a field**

**Public void setPassword(String password)**

**GetType – simple accessor method for a field**

**Public Char getType()**

**SetType – simple mutator method that sets a field**

**Public void setType(Char type)**

**GetStatus – simple accessor method for a field**

**Public Char getStatus()**

**SetStatus – simple mutator method that sets a field**

**Public void setStatus(Char status)**

**Admin**

**Super constructor that uses the account2 constructor**

**public Admin()**

**public String getPassword()**

**public char getType()**

**public char getStatus()**

**public void logout()**

**public void login(String userName, String password)**

**public void viewUniversities()**

**public void viewUsers()**

**public void addNewUser()**

**public void resetUserPassword()**

**public void addNewUniversity()**

**public void editUniversityInfo()**

**public void deleteSchool()**

**public void saveSchoolSearchStatistics()**

**public void deactivateUser()**

**public void deleteUser()**

**public void viewProfile()**

**User**

**Super constructor that uses the account2 constructor**

**University**

**University constructor - takes all instance fields for a university and initializes them**

**Public University(String name, String state, String location, String control, int numOfStudents, double percentFemale, double satVerbal, double satMath, double expenses, double finAid, int numApplicants, double perAdmitted, double perEnrolled, int academicScale, in socialScale, int lifeScale)**

**NAME AMOUNT Classes Tested**

|  |  |  |
| --- | --- | --- |
| **Tommy** | **4** | UC14, UC16, UC19, UC10 |
| **Ben** | **4** | UC1, UC2, UC3, UC4 |
| **Noah** | **4** | UC5, UC6, UC8, UC17 |
| **Moua** | **4** | UC21, UC13, UC11, UC14 |
| **Abdi** | **4** | UC7, UC8, UC20, UC12 |

**Meeting Minutes (4/10/19)**

The team worked together and finally figured out the search school.

Breakthrough: Moua and Noah fixed the pathing error on eclipse.

Struggle: Github keep giving problems regarding to pulling, pushing, and merging. Everyone was assigned cases when it came to making test classes.

Tommy worked on UC14, UC16, UC19, UC10. (testing)

Abdi worked on UC7, UC8, UC20, UC12. (testing)

Moua worked on UC21, UC13, UC11, UC14. (testing)

Ben worked on UC1, UC2, UC3, UC4. (testing)

Naoh worked on UC5, UC6, UC8, UC17. (testing)

Some of the testing cases were not well set-up. There were errors in class communication.

**Task Decomposition**

Moua – 20%

Tommy – 20%

Ben – 20%

Abdi – 20%

Noah – 20%