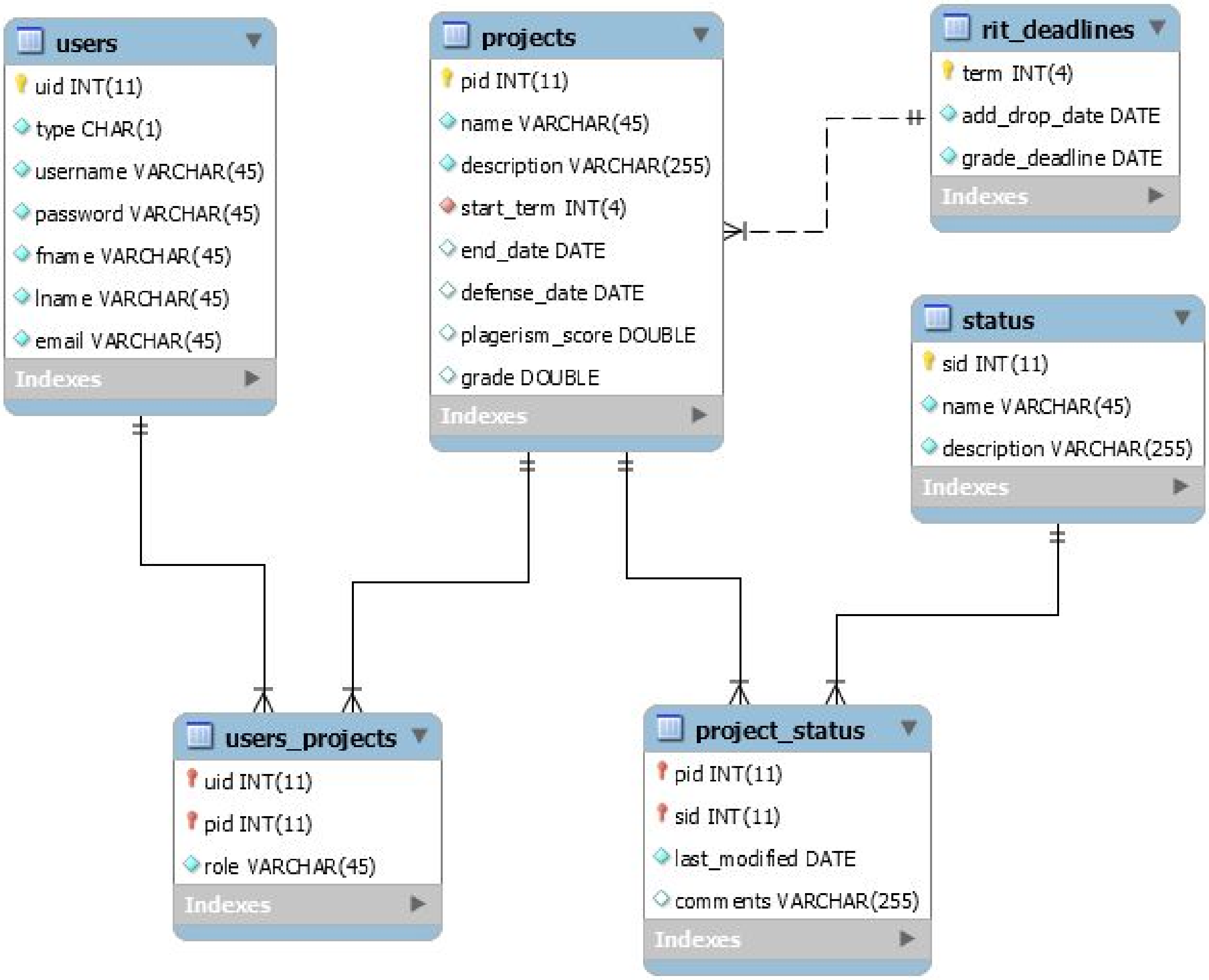
Milestone 1: Design Document

ISTE-330: Group 5

ER Diagram



Language Choice

Our group has chosen to use Java for a layers of the program as it is the programming language we are most familiar with and is optimal for retrieving, modifying, saving and presenting data as it is passed between the layers to the end user.

Passing Information Between Layers

There will be one database layer class that will handle queries passed to it from the data layer classes and return information to them either in the form of a boolean or a 2D Arraylist. The data layer classes (one of each major table in the database) will then parse this info, store it as global variables if need be, and pass them up to their corresponding business layer classes who then reformat the data to be displayed by the presentation layer.

Key Functionality

Users

- All users should be able to
  - Register an account
  - Set or change their name (first name, last name)
  - Change their email
  - Change their password
- All student users should be able to
  - Create a new project
  - Update project information (name, description, end_date)
- All faculty users should be able to
  - Modify any associated project
  - Update project information (name, description, end_date, plagiarism_score, grade)

Projects

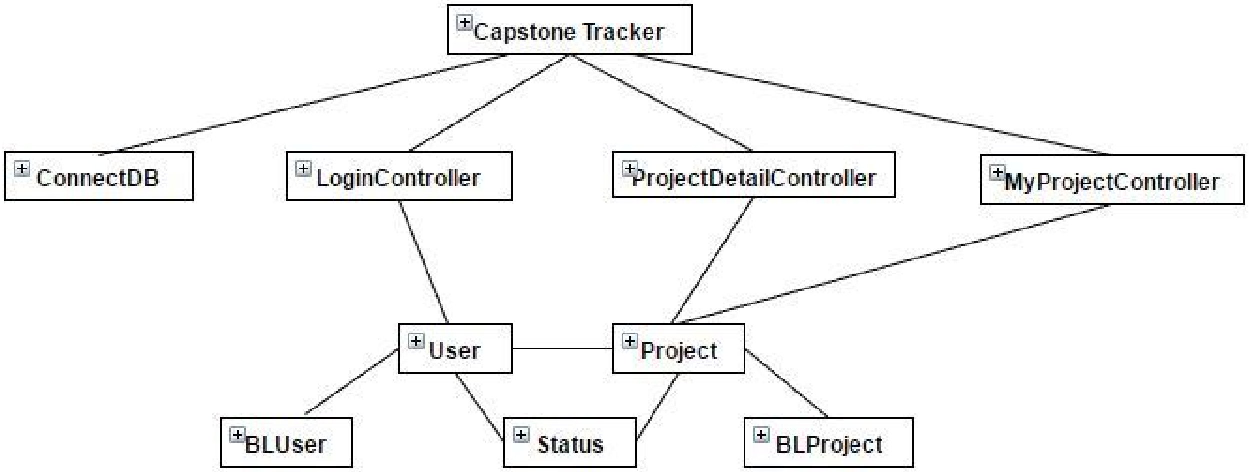
- All projects should…
  - Be able to be registered for consideration (with required project information)
  - Be able to be monitored for progress (e.g. status / comments available)
  - Students can modify any field in their project except their plagiarism_score

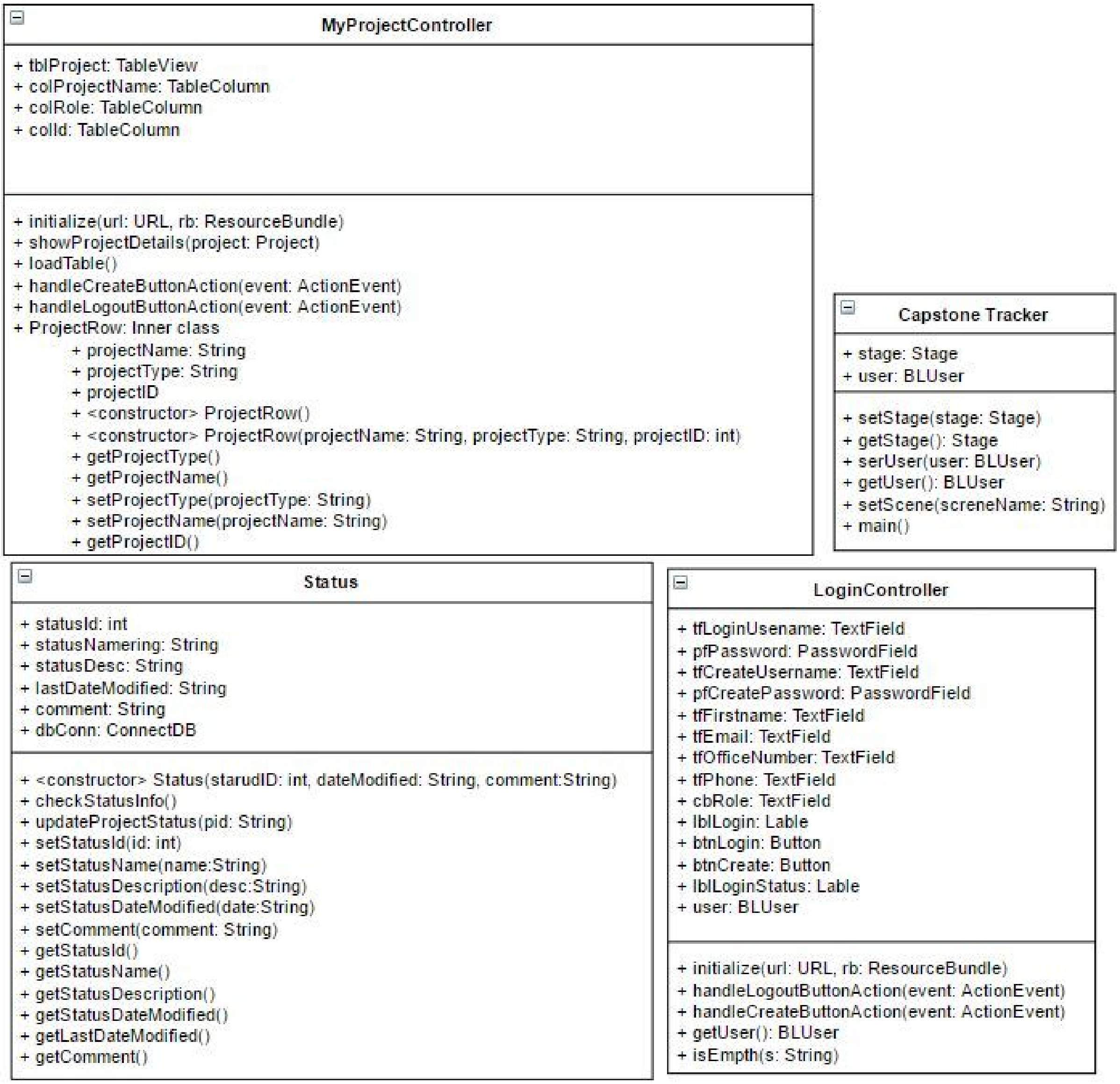
  - ○ Students need to be able to add faculty members as advisors for a project.

Areas of Particular Note or Concern

- We will need to be sure to keep track of which user roles have which permissions. For instance, grad students must not have permissions to edit their plagiarism score.
- The UI will need to have a global navigation system so that users can navigate between the various screens.

UML Diagram



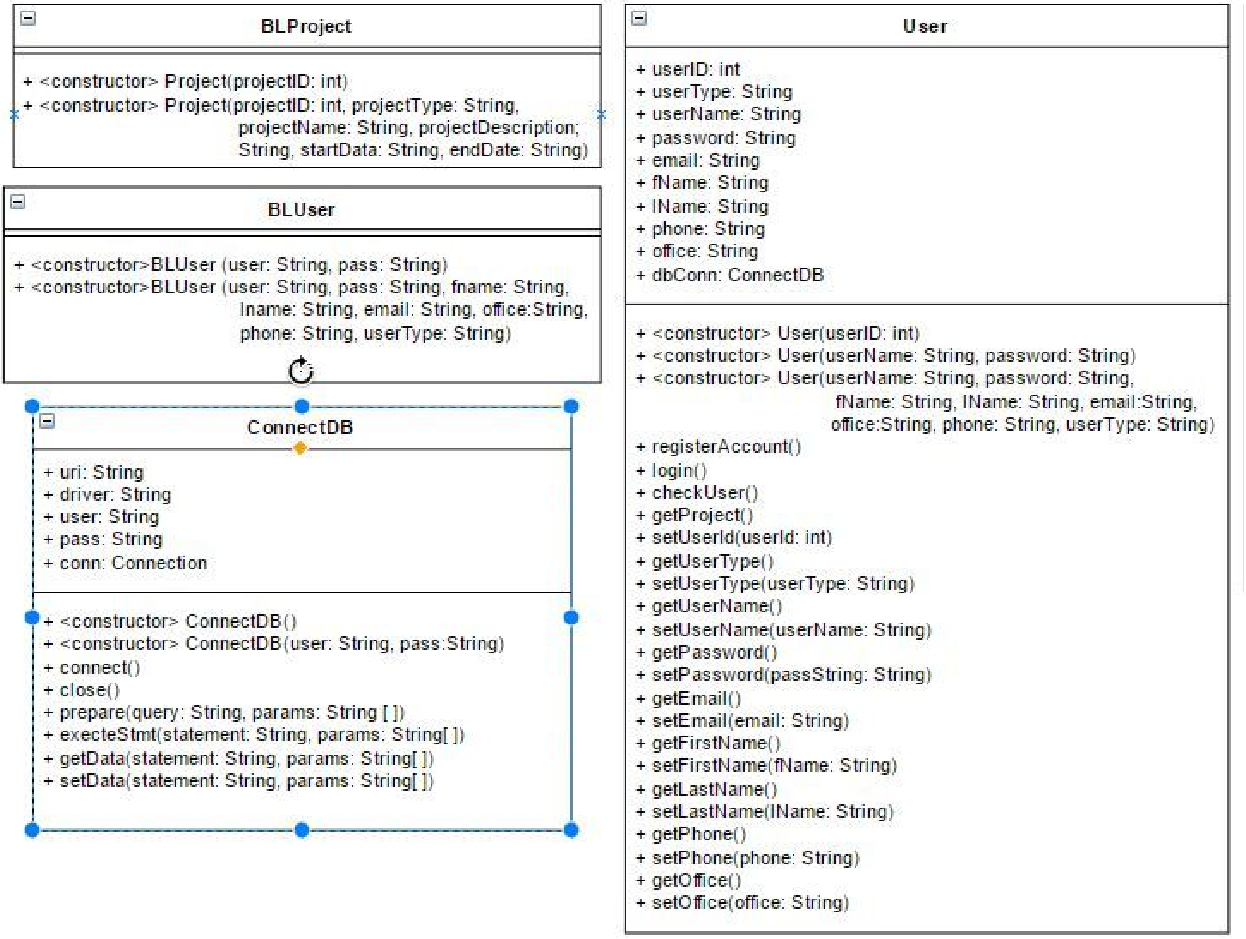


```
MyProjectController

+ tblProject: TableView
+ colProjectName: TableColumn
+ colRole: TableColumn
+ colId: TableColumn

+ initialize(url: URL, rb: ResourceBundle)
+ showProjectDetails(project: Project)
+ loadTable()
+ handleCreateButtonAction(event: ActionEvent)
+ handleLogoutButtonAction(event: ActionEvent)
+ ProjectRow: Inner class
        + projectName: String
        + projectType: String
        + projectID
        + <constructor> ProjectRow()
        + <constructor> ProjectRow(projectName: String, projectType: String, projectID: int)
        + getProjectType()
        + getProjectName()
        + setProjectType(projectType: String)
        + setProjectName(projectName: String)
        + getProjectID()
```

```
Capstone Tracker

+ stage: Stage
+ user: BLUser

+ setStage(stage: Stage)
+ getStage(): Stage
+ serUser(user: BLUser)
+ getUser(): BLUser
+ setScene(screneName: String)
+ main()
```

```
Status

+ statusId: int
+ statusNamering: String
+ statusDesc: String
+ lastDateModified: String
+ comment: String
+ dbConn: ConnectDB

+ <constructor> Status(starudID: int, dateModified: String, comment:String)
+ checkStatusInfo()
+ updateProjectStatus(pid: String)
+ setStatusId(id: int)
+ setStatusName(name:String)
+ setStatusDescription(desc:String)
+ setStatusDateModified(date:String)
+ setComment(comment: String)
+ getStatusId()
+ getStatusName()
+ getStatusDescription()
+ getStatusDateModified()
+ getLastDateModified()
+ getComment()
```

```
LoginController

+ tfLoginUsename: TextField
+ pfPassword: PasswordField
+ tfCreateUsername: TextField
+ pfCreatePassword: PasswordField
+ tfFirstname: TextField
+ tfEmail: TextField
+ tfOfficeNumber: TextField
+ tfPhone: TextField
+ cbRole: TextField
+ lblLogin: Lable
+ btnLogin: Button
+ btnCreate: Button
+ lblLoginStatus: Lable
+ user: BLUser

+ initialize(url: URL, rb: ResourceBundle)
+ handleLogoutButtonAction(event: ActionEvent)
+ handleCreateButtonAction(event: ActionEvent)
+ getUser(): BLUser
+ isEmpth(s: String)
```



BLProject

+ <constructor> Project(projectID: int)
+ <constructor> Project(projectID: int, projectType: String,
  projectName: String, projectDescription;
  String, startData: String, endDate: String)

BLUser

+ <constructor>BLUser (user: String, pass: String)
+ <constructor>BLUser (user: String, pass: String, fname: String,
  lname: String, email: String, office:String,
  phone: String, userType: String)

ConnectDB

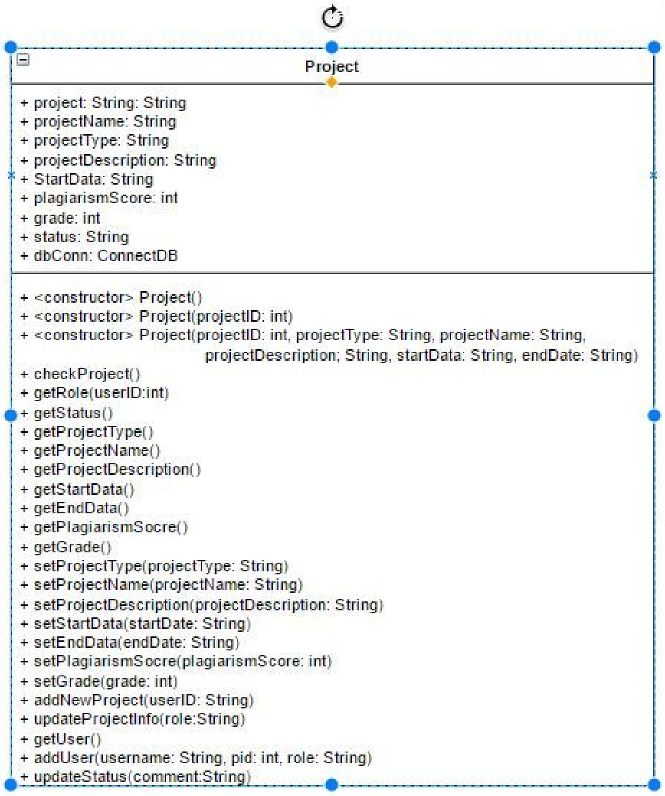
+ uri: String
+ driver: String
+ user: String
+ pass: String
+ conn: Connection

+ <constructor> ConnectDB()
+ <constructor> ConnectDB(user: String, pass:String)
+ connect()
+ close()
+ prepare(query: String, params: String [ ])
+ execteStmt(statement: String, params: String[ ])
+ getData(statement: String, params: String[ ])
+ setData(statement: String, params: String[ ])

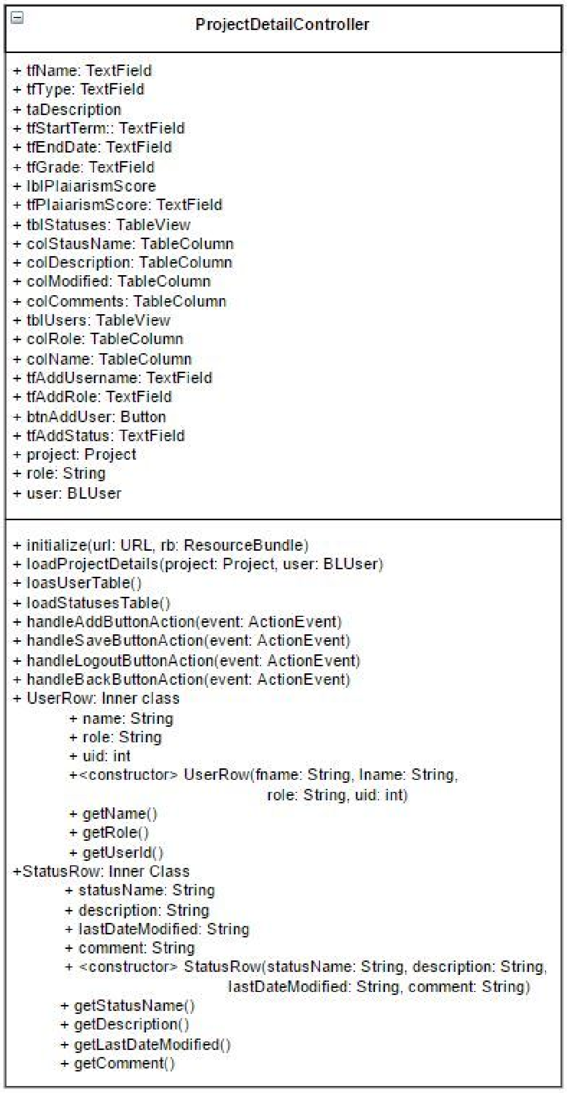
User

+ userID: int
+ userType: String
+ userName: String
+ password: String
+ email: String
+ fName: String
+ lName: String
+ phone: String
+ office: String
+ dbConn: ConnectDB

+ <constructor> User(userID: int)
+ <constructor> User(userName: String, password: String)
+ <constructor> User(userName: String, password: String,
  fName: String, lName: String, email:String,
  office:String, phone: String, userType: String)
+ registerAccount()
+ login()
+ checkUser()
+ getProject()
+ setUserId(userId: int)
+ getUserType()
+ setUserType(userType: String)
+ getUserName()
+ setUserName(userName: String)
+ getPassword()
+ setPassword(passString: String)
+ getEmail()
+ setEmail(email: String)
+ getFirstName()
+ setFirstName(fName: String)
+ getLastName()
+ setLastName(lName: String)
+ getPhone()
+ setPhone(phone: String)
+ getOffice()
+ setOffice(office: String)



```
Project
+ project: String: String
+ projectName: String
+ projectType: String
+ projectDescription: String
+ StartData: String
+ plagiarismScore: int
+ grade: int
+ status: String
+ dbConn: ConnectDB

+ <constructor> Project()
+ <constructor> Project(projectID: int)
+ <constructor> Project(projectID: int, projectType: String, projectName: String,
                        projectDescription; String, startData: String, endDate: String)
+ checkProject()
+ getRole(userID:int)
+ getStatus()
+ getProjectType()
+ getProjectName()
+ getProjectDescription()
+ getStartData()
+ getEndData()
+ getPlagiarismSocre()
+ getGrade()
+ setProjectType(projectType: String)
+ setProjectName(projectName: String)
+ setProjectDescription(projectDescription: String)
+ setStartData(startDate: String)
+ setEndData(endDate: String)
+ setPlagiarismSocre(plagiarismScore: int)
+ setGrade(grade: int)
+ addNewProject(userID: String)
+ updateProjectInfo(role:String)
+ getUser()
+ addUser(username: String, pid: int, role: String)
+ updateStatus(comment:String)
```



```
ProjectDetailController
+ tfName: TextField
+ tfType: TextField
+ taDescription
+ tfStartTerm:: TextField
+ tfEndDate: TextField
+ tfGrade: TextField
+ lblPlaiarismScore
+ tfPlaiarismScore: TextField
+ tblStatuses: TableView
+ colStausName: TableColumn
+ colDescription: TableColumn
+ colModified: TableColumn
+ colComments: TableColumn
+ tblUsers: TableView
+ colRole: TableColumn
+ colName: TableColumn
+ tfAddUsername: TextField
+ tfAddRole: TextField
+ btnAddUser: Button
+ tfAddStatus: TextField
+ project: Project
+ role: String
+ user: BLUser

+ initialize(url: URL, rb: ResourceBundle)
+ loadProjectDetails(project: Project, user: BLUser)
+ loasUserTable()
+ loadStatusesTable()
+ handleAddButtonAction(event: ActionEvent)
+ handleSaveButtonAction(event: ActionEvent)
+ handleLogoutButtonAction(event: ActionEvent)
+ handleBackButtonAction(event: ActionEvent)
+ UserRow: Inner class
        + name: String
        + role: String
        + uid: int
        +<constructor> UserRow(fname: String, lname: String,
                               role: String, uid: int)
        + getName()
        + getRole()
        + getUserId()
+StatusRow: Inner Class
        + statusName: String
        + description: String
        + lastDateModified: String
        + comment: String
        + <constructor> StatusRow(statusName: String, description: String,
                               lastDateModified: String, comment: String)
       + getStatusName()
       + getDescription()
       + getLastDateModified()
       + getComment()
```

User

- Class that represents user's that can access the system; stores data from the people table
- Mutators and Accessor Methods for userID, userName, userType, password, email , first and last name, phone and office numbers
- registerAccount(); used to add a new user account to the table
- login(); checks that a user existing using username and password and populates the rest of the variable with data from the DB
- checkUser(); performs the same function as login() but checks using userID
- getProject(); returns an ArrayList of Projects that the user is involved in

Project

- Class the represents the projects that have been registered in the system
- Mutators and Accessors for projectID, projectName, projectType, projectDesc, startDate, endDate, grade, and plagiarism score
- checkProject(): performs the same function as User.checkUser() for a Project object
- getRole(): retrieves the role assigned to a user involved in a project
- getStatus(): returns all the status' that have been applied to a project
- addNewProject(): inserts new project and people_project records
- updateProjectInfo(): updates the information for a project
- getUsers(): retrieves users that are associated with this project
- addUser(): adds users to a project
- updateStatus: assigns a new status to the project by calling Status.updateProjectStatus()

Status

- Class used to store status info as it relates to a project
- Mutators and Accessors for statusID, statusName, statusDesc, date_modified and comment
- checkStatusInfo(): retrieves name and description for a status to add to the date and comment
- updateProjectStatus(): inserts a new record in the project_status table when a new status is assigned to a project

ConnectDB

- DB connection object
- connect() and close self-explanitory
- prepare(): prepares a query for execution
- executeStmt(): executes prepared UPDATE, INSERT and DELETE statements
- getDate(): returns a 2d ArrayList for used by the DL classes
- setData(): checks that an update statement worked
- setResultingData(): Iterates through a ResultSet and uses MetaData to record info to a 2d Array returned by getData()

DLExcepetion

- Exception reporting object; same as the one used in the PEs so no info really need to be given

Business Layer Classes

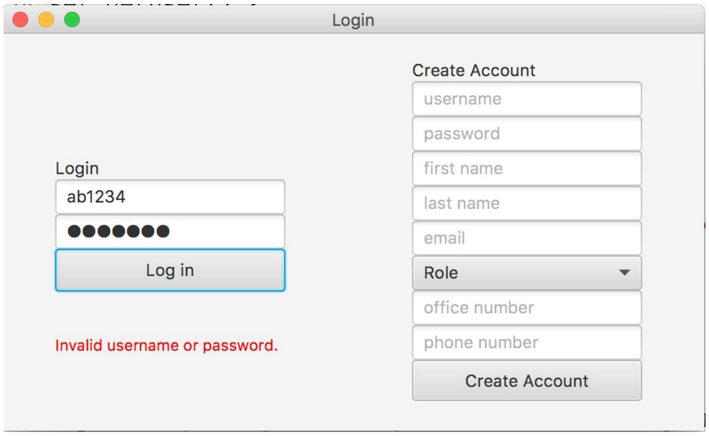
- Obscures the methods of the Data Layer classes from the Presentation Layer

Controller Classes

- MyProjectsController
  - Loads the fxml file for the "MyProjects" scene. Loads the projects table with all users associated with a project.
- LoginController
  - Loads the fxml file for the "Login" scene. Gets user input from text fields. If valid login credentials are entered, a user is logged in and directed to the "ProjectDetail" scene.
- ProjectDetailController
  - Loads the fxml file for the "ProjectDetail" scene. This controller

CapstoneTracker

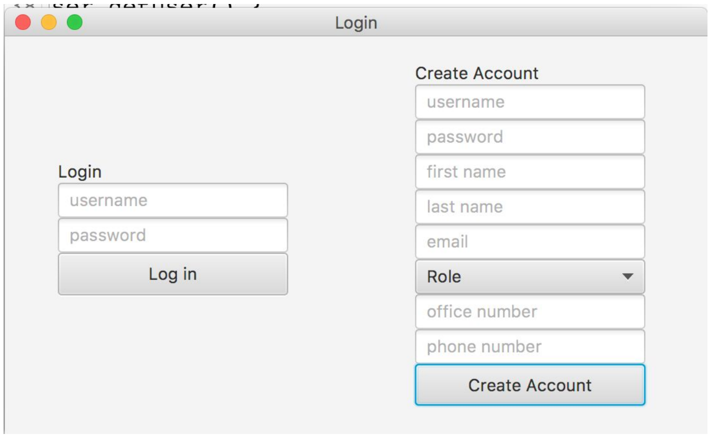
User Interface



ab1234

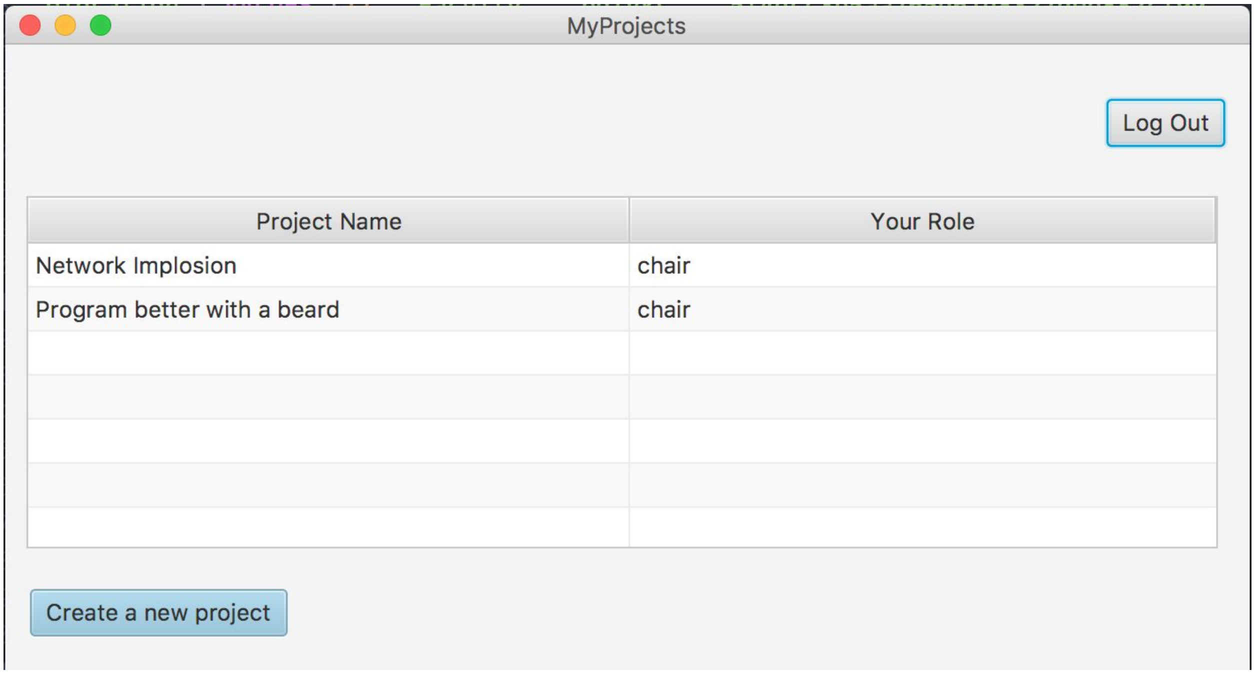
●●●●●●●

Invalid username or password.



Login Screen

The Login Screen will have two sections: a login form, and a form to create an account. The login form will have two text fields–one for a username and one for a password–and a button to login. The "create account" form will have fields to collect all the necessary information for the "people" table–username, password, email, type/role, first name, and last name. When invalid login credentials are entered, an error message is displayed to the user.



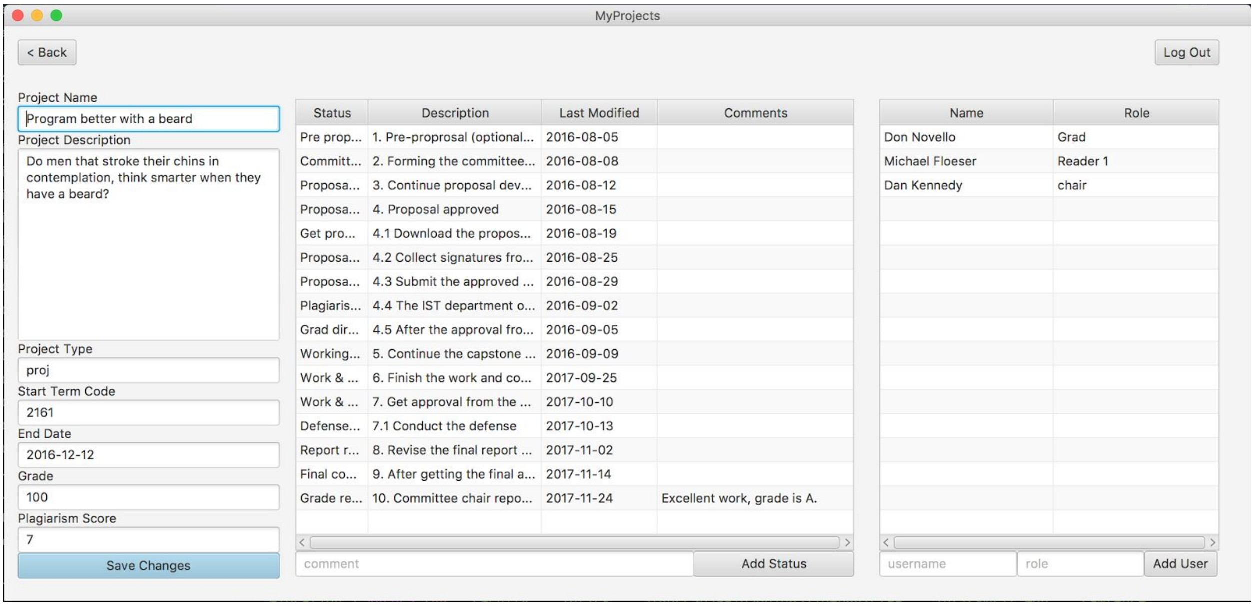
Project Name	Your Role
Network Implosion	chair
Program better with a beard	chair

My Projects Screen

This screen displays a list of any and all projects that the registered user is involved with. Double click on a row to view the details for that project.

The "Create a new project" button takes the user to the "Project Overview Screen" and initializes a new project.

The "Log Out" button takes the user back to the "Login" scene.



Project Name
Program better with a beard

Project Description
Do men that stroke their chins in contemplation, think smarter when they have a beard?

Project Type
proj

Start Term Code
2161

End Date
2016-12-12

Grade
100

Plagiarism Score
7

Status	Description	Last Modified	Comments
Pre prop...	1. Pre-proposal (optional...	2016-08-05	
Committ...	2. Forming the committee...	2016-08-08	
Proposa...	3. Continue proposal dev...	2016-08-12	
Proposa...	4. Proposal approved	2016-08-15	
Get pro...	4.1 Download the propos...	2016-08-19	
Proposa...	4.2 Collect signatures fro...	2016-08-25	
Proposa...	4.3 Submit the approved ...	2016-08-29	
Plagiaris...	4.4 The IST department o...	2016-09-02	
Grad dir...	4.5 After the approval fro...	2016-09-05	
Working...	5. Continue the capstone ...	2016-09-09	
Work & ...	6. Finish the work and co...	2017-09-25	
Work & ...	7. Get approval from the ...	2017-10-10	
Defense...	7.1 Conduct the defense	2017-10-13	
Report r...	8. Revise the final report ...	2017-11-02	
Final co...	9. After getting the final a...	2017-11-14	
Grade re...	10. Committee chair repo...	2017-11-24	Excellent work, grade is A.

Name	Role
Don Novello	Grad
Michael Floeser	Reader 1
Dan Kennedy	chair

Project Detail Screen

This screen displays the fields from the "project" table. This screen is used by any user type to view a selected project. This screen also displays the information from the "status" table for a project. On this screen, a student can edit the fields of their project except the grade, the plagiarism score, and the id. If a student does not have a project, going to this screen will initialize a new project with empty fields. Also on this screen, students can add faculty to their project by username. The username must already exist in the database for this to work.

This screen contains a table with each status, the status description, comments, the date the status was last modified, and the dates the statuses were completed. This section of the screen may be a scrollpane, depending on how many statuses there are and how much room they take up.

The "Log Out" button takes the user back to the "Login" scene.