Gaurav Srivastava • Naveen Ganessin • **Alex Rosenberg** • Michael Schloss

ABC

Sprint 1 Plan

Team Members

Gaurav Srivastava — srivast6@purdue.edu
Naveen Ganessin — nganessi@purdue.edu
Alex Rosenberg — rosenbea@purdue.edu
Michael Schloss — mschlos@purdue.edu

Sprint Overview

This sprint's primary focus is on initializing the database and its proper API, build the login UI and APIs, and build rudimentary versions of User object and UserQuery JSON translation class. We want to be able to have the ability to log into the application.

Michael Schloss is the SCRUM master. We will be meeting on Monday's from 10:30a - noon. If necessary, we will meet on Wednesdays (at the same time) to finish up any documents.

There are minimal risks to be experienced during this sprint. Perhaps the most riskiest process would be communicating with the database. However, we are using API from a previous Java project that did not have issues with this.

Another risk would be making sure the UI rendered properly on any OS (for now, focusing on macOS and Windows). This will be more difficult as code can be different among the others.

Finally, the User object will be pretty detailed. A risk will be keeping its code load low and efficient while making sure all features get implemented.

Current Sprint Detail

Backlog ID: 0
User Story: As a user, I would like to log in to the application.

Task Description	Expected Hours	Owner
Create database tables to support logins of the three different type of users (instructors, students, admin)	2	Gaurav Srivastav
Create APIs that can check if a user's login information is correct or wrong	8	Gaurav Srivastav
Create the User object class, including login-related constructor(s), fields, accessors, and mutators as shown in the Design Document Objects Layer UML	10	Alex Rosenberg
Create UserQuery JSON translation class that allows for addition, modification and removal of Users	15	Naveen Ganessin
Create UI code for a login screen and direct the correct UI for the type of user logging in	5	Michael Schloss
Create UI code for a home screen that the user will see immediately after successful login	5	Michael Schloss
Create UI code for an alert if login is incorrect	5	Michael Schloss

ACCEPTANCE CRITERIA

- Given that a user opens the application, they should be able to view a login screen
- Given that a user enters their credentials correctly, they should be able to view their main page
- Given that a user enters their credentials incorrectly, they should be prompted to reenter their credentials

Backlog ID: 3
User Story: As an admin, I can add/delete instructors.

Task Description	Expected Hours	Owner
Create APIs that can create a new instructor in the database	**10**	**Gaurav Srivastav**
Continue work on the User object class such as related constructor(s), fields, accessors, and mutators as shown in the Design Document Objects Layer UML	**10**	**Alex Rosenberg**
Create UserQuery JSON translation class that allows for addition, modification and removal of Users	**5**	**Naveen Ganessin**
Create UI for removing and confirmation of deletion of teachers	**5**	**Michael Schloss**
Create UI for entering teacher's username, temp password and usertype	**5**	**Michael Schloss**
Create UI for list of teachers in database	**5**	**Michael Schloss**

ACCEPTANCE CRITERIA

- Given that an admin adds an instructor, the app should return a success prompt and show the userid and password
- Given that an admin adds a duplicate instructor, the app should return an error message claiming the instructor already exists in the course
- Given that an admin deletes an instructor, the app should return a success prompt saying the instructor was removed from the course
- Given that an admin tries to delete an instructor that does not exist, the app should return an error message saying the instructor does not exist or is not part of the course currently

Backlog ID: 4
User Story: As an admin, I can add/delete students.

Task Description	Expected Hours	Owner
Create APIs that can create a new student in the database	**8**	**Gaurav Srivastav**
Continue work on the User object class such as related constructor(s), fields, accessors, and mutators as shown in the Design Document Objects Layer UML	**10**	**Alex Rosenberg**
Create UI for removing and confirmation of deletion of student	**5**	**Michael Schloss**
Create UI for entering student's username and temp password	**5**	**Michael Schloss**
Create UI for list of students in database	**5**	**Michael Schloss**
Create UI for entering user's user type and class standing	**5**	**Michael Schloss**

ACCEPTANCE CRITERIA

- Given that an admin adds an students, the app should return a success prompt and show the userid and password
- Given that an admin adds a duplicate student, the app should return an error message claiming the student already exists in the course
- Given that an admin deletes an student, the app should return a success prompt saying the student was removed from the course
- Given that an admin tries to delete an student that does not exist, the app should return an error message saying the student does not exist or is not part of the course currently

Backlog ID: 6
User Story: As an admin, I can get course-wide or school-wide grade stats.

Task Description	Expected Hours	Owner
Create the Grade object class, including related constructor(s), fields, accessors, and mutators as shown in the Design Document Objects Layer UML	**10**	**Alex Rosenberg**
Create GradeQuery JSON Translation class that allows for addition, modification and removal of grades	**20**	**Naveen Ganessin**
Create database tables that can store different courses, assignments and grades	**2**	**Gaurav Srivastav**
Create APi that returns information about for every course	**10**	**Gaurav Srivastav**
Create UI code for admin tab bar and home layout	**5**	**Michael Schloss**
Create UI code for grades view with stats and charts	**5**	**Michael Schloss**
Create UI code for list of assignments grades per user	**5**	**Michael Schloss**

ACCEPTANCE CRITERIA

- Given that an admin opens the statistic and charts tab, admin should be able to see school wide grade statistics

Total Hours

Member	Total Hours
Alex Rosenberg	**40**
Naveen Ganessin	**40**
Gaurav Srivastava	**40**
Michael Schloss	**60**

Remaining Backlog Items

ID	Requirement
1	As an admin, I can add/delete classes
2	As an admin, I can change class storage quota
3	
4	
5	As an admin, I can post school-wide announcements
6	
7	As an Instructor, I can add/delete assignments, quizzes, and tests
8	As an Instructor, I can view file submissions from students
9	As an Instructor, I can set time-limits on assignments, quizzes, and tests
10	As an Instructor, I can post class-wide announcements
11	As an Instructor, I can send these announcements as emails should they wish
12	As an Instructor, I can host content on their class
13	As an Instructor, I can specify whether their hosted content is private or public
14	As an Instructor, I can teach multiple courses
15	As an Instructor, I can submit students' grades for assignments, quizzes, and tests
16	As an Instructor, I can view grades for any of their classes
17	As an Instructor, I can generate grade reports
18	As an Instructor, I should be able to generate a random text for students to enter on their account for attendance. (if time allows)
19	As a Student, I can see school-wide announcements
20	As a Student, I can see class-wide announcements
21	As a Student, I can see their grades from every class to which they're enrolled

ID	Requirement
22	As a Student, I can submit assignments
23	As a Student, I can take quizzes or tests
24	As a Student, I can send email to their teacher from the application
25	As a Student, I can view/consume public class files
26	As a Student, I should be able to receive an email notifications about assignments and exams due (if time allows)
27	As a Student, I should be able to enter the exact teacher's random text for attendance. (if time allows)