TEAM: Michael Brughelli, Chris Flauta, Jing Guo, Stephen Rowell
TITLE: Classroom Scheduler
PROJECT SUMMARY:

A scheduling program for school administrators to easily assign classes to appropriately sized classrooms at times that do not conflict with the needs and spatial requirements of other classes.

REQUIREMENTS:

Business Requirements:

ID	Requirement	Topic Area	User	Priority
BR-1	All login userIDs must be associated with university	Authentication	All	High
BR-2	Only admin can grant or deny new permissions to users	Authentication	Admin	Low-Med
BR-3	Add/ Modify classrooms data.	Database	Admin	Critical
BR-3.1	Add/ Modify classroom capacity, room number, building	Database	Admin	Critical
BR-3.2	Only Admin can perform this action	Authentication	Admin	High
BR-4	Add/ Modify course data	Database	Department Head	Critical
BR-4.1	Add/ Modify course enrollment size, instructor of the course, department of course	Database	Department Head	Critical
BR-4.2	Only Department head can perform this action	Authentication	Department Head	Med

User Requirements:

ID	Requirement	Topic Area	User	Priority
UR-1	New Users can request/create an account	Accounts	New Users	Critical
UR-2	Instructors can view class schedules only	Authentication	Instructor	Low
UR-3	Students can view class schedules only	Authentication	Student	Low
UR-4	Department Head must be able to request classroom changes	Admin request	Department Head	High

Functional Requirements:

ID	Requirement	Topic Area	User	Priority
FR-1	Users must log-in before using the software	User Interface	All	Critical
FR-2	Classroom Schedule must be accessible to all Users	User Interface	All	High
FR-3	Department Heads must be able to submit course changes	User Interface	Department Head	High
FR-4	Admin must be able to run/rerun Scheduling Algorithm	User Interface	Admin	Critical
FR-5	Classroom Schedule must be sortable for easy searching	User Interface	All	Med
FR-6	User can request a password reset	User Interface	All	Med

Non- Functional Requirements:

ID	Requirement	Topic Area	User	Priority
NR-1	All login userIDs must be checked for permissions	Authentication	All	High
NR-2	Classroom Table must have: building, room number, capacity	Database	All	Critical
NR-3	Course Table must have: course id, title, department name, number of enrolled students, instructor name	Database	All	Critical
NR-4	User Table must have: user id, name, permission level, department name (if employee of university)	Database	All	Critical
NR-5	Department Table must have: department name	Database	All	Critical

USERS and TASKS:

Use Case ID:	UC-01
Use Case Name:	Create Course List
Description:	Department Heads use import option to create a list of Courses for the Scheduler.

Actors:	Department Head		
Pre-Conditions:	Department Head has access to the scheduler and Spreadsheet of Courses containing information on Enrollment Size, Credit Hours, Course ID, and Instructor.		
Post-Conditions:	Scheduling Software has full list of Courses for given Department.		
Frequency of Use:	One Time Setup		
Flow of Events:		**Actor Action**	**System Response**
	1	Log in to program	Verify Credentials and Load Client
	2	File→Import Course List	Opens Dialog Window for file Search.
	3	Select Course List File	Opens Selected File
	4	Click Import	Insert Course Data into Database
	5	Confirm Load	
Variations:	2. Use “Add Course” feature to manually add a Course to Database		
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-02
Use Case Name:	Add New Course
Description:	Department Heads use “Add Course” feature to add a new Course to their Course List.

Actors:	Department Head		
Pre-Conditions:	Department Head has access to the scheduler and Course List has already been created.		
Post-Conditions:	Course List contains newly added Course.		
Frequency of Use:	When Needed		
Flow of Events:		**Actor Action**	**System Response**
	1	Log in to program	Verify Credentials and Load Client
	2	Click “Add Course” on Toolbar	Opens Dialog Window to request Course Information
	3	Fill Out Course Information and click “Add”	Adds new Course Data to Course Database
Variations:	None		
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-03
Use Case Name:	Remove Course
Description:	Department Heads use “Remove Course” feature to remove a Course from their Course List.

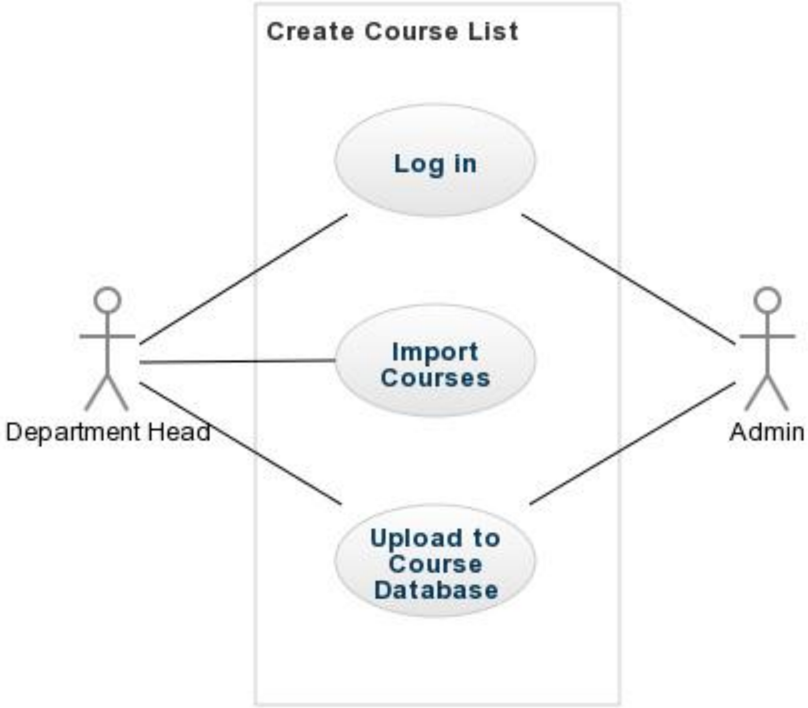
Actors:	Department Head		
Pre-Conditions:	Department Head has access to the scheduler and Course List has already been created.		
Post-Conditions:	Course List no longer contains Removed Course		
Frequency of Use:	When Needed		
Flow of Events:		**Actor Action**	**System Response**
	1	Log in to program	Verify Credentials and Load Client
	2	Click “Remove Course” on Toolbar	Opens Dialog Window to request Course Information
	3	Enters Course ID of Course to Remove	Finds all Courses with given ID and List them
	4	Select Course to be Removed and click “Remove”	Get Course Information and ask for Confirmation of Removal
	5	Click “Confirm” to confirm removal	Remove Course from Database
Variations:	4. Click “Cancel” to cancel Removal 5. Click “Cancel” to cancel Removal		
Notes and Issues:			
Developer Notes:			

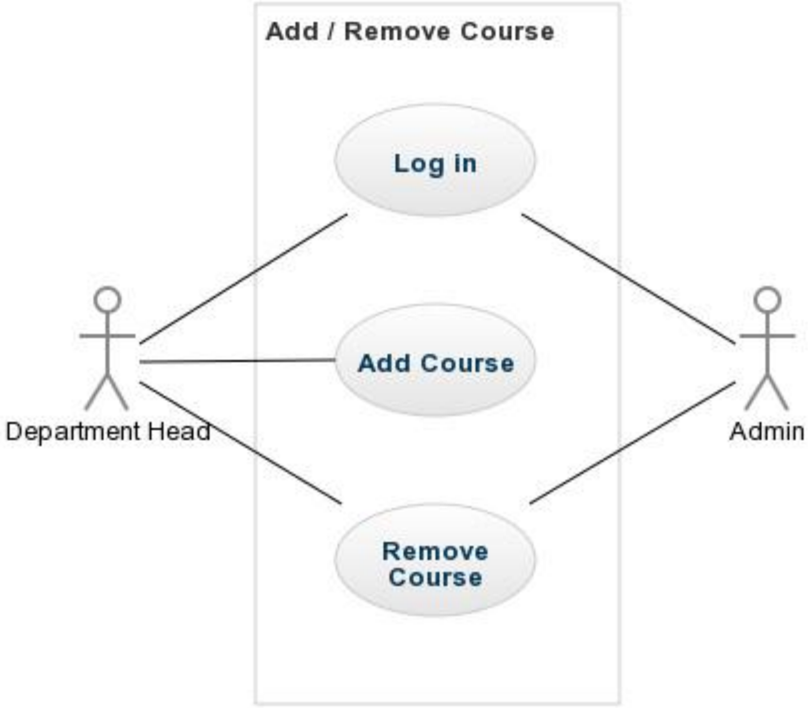
Use Case ID:	UC-04
Use Case Name:	View Course Schedule
Description:	Department Heads use “View Schedule” feature to view Course Schedule for their Department

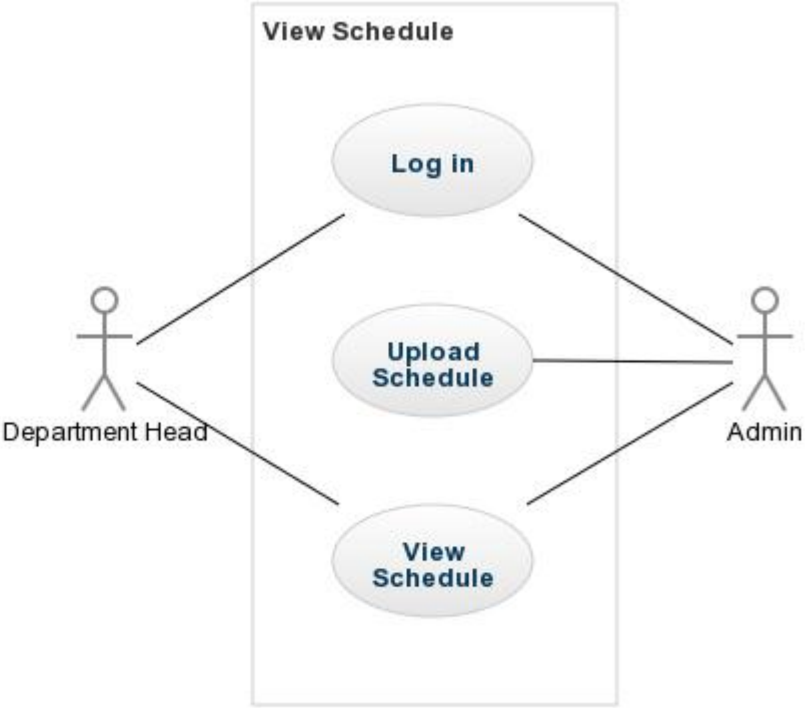
Actors:	Department Head		
Pre-Conditions:	Department Head has access to the scheduler and Course Schedule has already been created		
Post-Conditions:	Full Course Schedule available with Sort and Search capabilities		
Frequency of Use:	When Needed		
Flow of Events:		**Actor Action**	**System Response**
	1	Log in to program	Verify Credentials and Load Client
	2	Click “View Schedule” on Toolbar	Opens New Window with Course Schedule Information
	3	Click Columns to Sort	Sort Column in Ascending (first click) / Descending order
	4	Enter text into Search	Search Course Schedule for given text
	5	Click “Close”	Close Course Schedule Window
Variations:			
Notes and Issues:			
Developer Notes:			

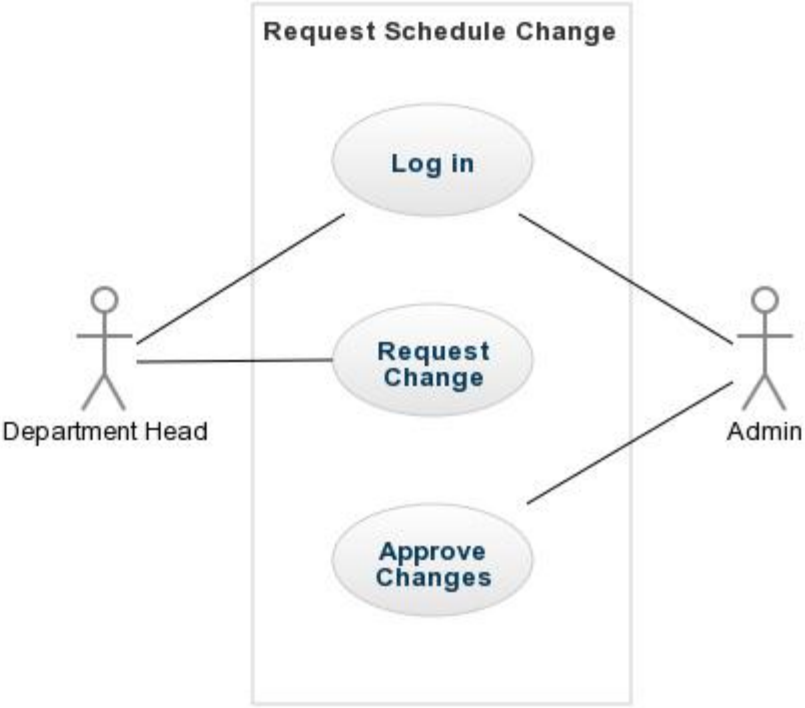
Use Case ID:	UC-05
Use Case Name:	Course Schedule Change Request
Description:	Department Heads use “Request Changes” feature to request changes after Class Schedule already created.

Actors:	Department Head		
Pre-Conditions:	Department Head has access to the scheduler and Course Schedule already created		
Post-Conditions:	Requested Schedule Change submitted, pending Admin Approval		
Frequency of Use:	When Needed		
Flow of Events:		**Actor Action**	**System Response**
	1	Log in to program	Verify Credentials and Load Client
	2	Click “View Schedule” on Toolbar	Opens New Window with Course Schedule Information
	3	Highlight Course and click “Request Changes” Button	Ask for Type of Change (Remove/Modify)
	4	Select “Modify” and Enter proposed modification	Ask for Confirmation
	5	Click “Confirm” to confirm request	Send request to Admin
Variations:	4. Click “Remove” to drop Course from Schedule		
Notes and Issues:			
Developer Notes:			


Create Course List
Log in
Import Courses
Upload to Course Database
Department Head
Admin


Add / Remove Course
Log in
Add Course
Remove Course
Department Head
Admin


View Schedule
Log in
Upload Schedule
View Schedule
Department Head
Admin


Request Schedule Change
Log in
Request Change
Approve Changes
Department Head
Admin

Use Case ID:	UC-06
Use Case Name:	Creating a new user
Description:	New user can request and create a new account

Actors:	Admin
Pre-Condition :	Click the create new user button and fill out information needed
Post-condition:	Submit the user information to database
Frequency of Use:	When needed
Flow of Events:	

	Actor Action	System Response
1	Click on new user button	Open a new window to let user type information
2	Read in email and password	Check the email address if is already exist and read in password
3	Click create button	Put information into database

Variations	
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC-07
Use Case Name:	Resetting a password
Description:	Let user to reset password if they need

<table>
<tr><td>Actors:</td><td>Admin</td></tr>
<tr><td>Pre-Condition:</td><td>User must already has an account</td></tr>
<tr><td>Post-condition:</td><td>User can submit the new password and log in</td></tr>
<tr><td>Frequency of Use:</td><td>When needed</td></tr>
<tr><td>Flow of Events:</td><td>
<table>
<tr><th></th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1</td><td>Log into the program first</td><td>Check if this a valid account</td></tr>
<tr><td>2</td><td>Click reset password button</td><td>Show up a page to let user enter new password</td></tr>
<tr><td>3</td><td>Enter the new password</td><td>Save the new password</td></tr>
</table>
</td></tr>
<tr><td>Variations</td><td></td></tr>
<tr><td>Notes and Issues:</td><td></td></tr>
<tr><td>Developer Notes:</td><td></td></tr>
</table>

Use Case ID:	UC-08
Use Case Name:	Manage classroom database
Description:	add or delete a classroom

<table>
<tr><td>Actors:</td><td>Admin</td></tr>
<tr><td>Pre-Condition:</td><td>The database must exist</td></tr>
<tr><td>Post-condition:</td><td>Database match the changes</td></tr>
<tr><td>Frequency of Use:</td><td>When needed</td></tr>
<tr><td>Flow of Events:</td><td>
<table>
<tr><td></td><td>Actor Action</td><td>System Response</td></tr>
<tr><td>1</td><td>Log into system</td><td>Verify account</td></tr>
<tr><td>2</td><td>Click the manage database bottom</td><td>Access into database</td></tr>
<tr><td>3</td><td>Click add or delete button to modify classroom</td><td>Add or delete classroom from database</td></tr>
<tr><td>4</td><td>Choose a certain classroom to modify the specific information</td><td>Change the information in the database</td></tr>
</table>
</td></tr>
<tr><td>Variations</td><td></td></tr>
<tr><td>Notes and Issues:</td><td></td></tr>
<tr><td>Developer Notes:</td><td></td></tr>
</table>

Use Case ID:	UC-09
Use Case Name:	Manage courses database
Description:	Add and remove courses

<table>
<tr><td>Actors:</td><td>Admin</td></tr>
<tr><td>Pre-Condition:</td><td>The database must exist</td></tr>
<tr><td>Post-condition:</td><td>Database match the changes</td></tr>
<tr><td>Frequency of Use:</td><td>When needed</td></tr>
<tr><td>Flow of Events:</td><td>
<table>
<tr><th></th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1</td><td>Log into system</td><td>Verify account</td></tr>
<tr><td>2</td><td>Click the “manage course” button</td><td>Access into database</td></tr>
<tr><td>3</td><td>Click “add” button to insert a new course, including the information about time and capacity</td><td>Check if the course is in the database, if not add into database</td></tr>
<tr><td>4</td><td>Click “delete” button to delete a course</td><td>Check if the course is in the database, if it is there then delete</td></tr>
</table>
</td></tr>
<tr><td>Variations</td><td></td></tr>
<tr><td>Notes and Issues:</td><td></td></tr>
<tr><td>Developer Notes:</td><td></td></tr>
</table>

Use Case ID:	UC-10
Use Case Name:	Manage user database
Description:	Add and remove users

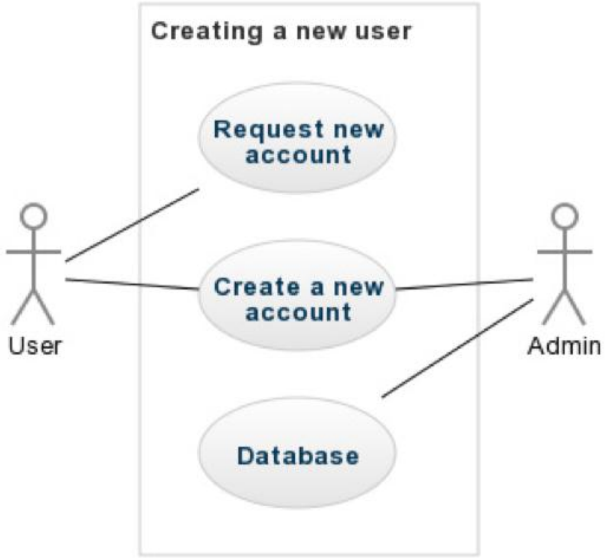
<table>
<tr><td>Actors:</td><td>Admin</td></tr>
<tr><td>Pre-Condition:</td><td>The database must exist</td></tr>
<tr><td>Post-condition:</td><td>Database match the changes</td></tr>
<tr><td>Frequency of Use:</td><td>When needed</td></tr>
<tr><td>Flow of Events:</td><td>
<table>
<tr><th></th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1</td><td>Log into system</td><td>Verify account</td></tr>
<tr><td>2</td><td>Click the “manage user” button</td><td>Access into database</td></tr>
<tr><td>3</td><td>Read in a request from user to create a new account</td><td>Insert the information into database</td></tr>
<tr><td>4</td><td>Remove user information</td><td>Delete the information the request to remove</td></tr>
</table>
</td></tr>
<tr><td>Variations</td><td></td></tr>
<tr><td>Notes and Issues:</td><td></td></tr>
<tr><td>Developer Notes:</td><td></td></tr>
</table>

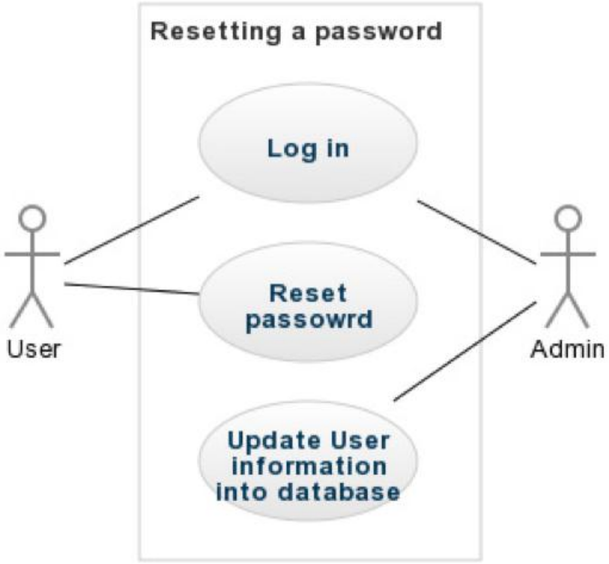
Use Case ID:	UC-11
Use Case Name:	Manage schedule database
Description:	Add, change or delete schedule from database

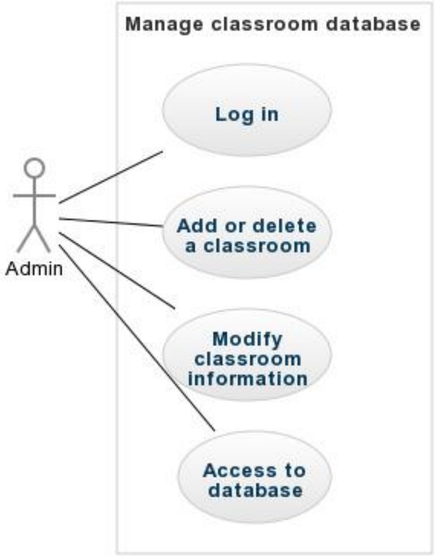
Actors:	Admin
Pre-Condition:	The database must exist
Post-condition:	Database match the changes
Frequency of Use:	When needed
Flow of Events:	

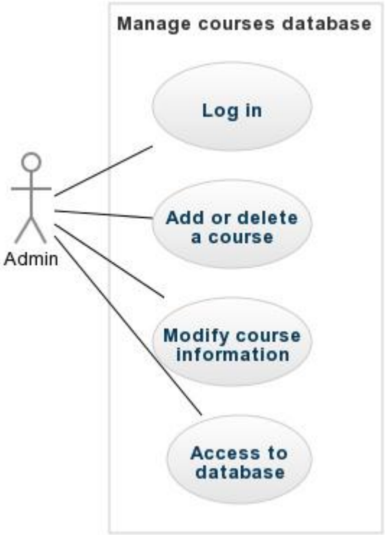
	Actor Action	System Response
1	Make sure log in first	Log into the system
2	Click the manage schedule bottom	Access into database
3	Click “add” to insert a new schedule	Run the schedule algorithm to add in to database
4	Click ”change” to change a schedule	Run the schedule algorithm to change in database
5	Click ”delete” to delete a schedule	Delete a schedule

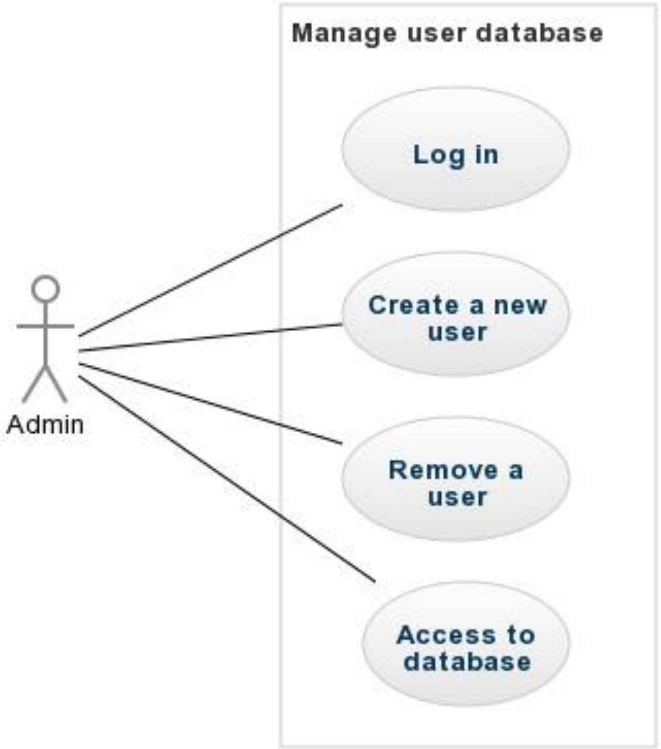
Variations	
Notes and Issues:	
Developer Notes:	

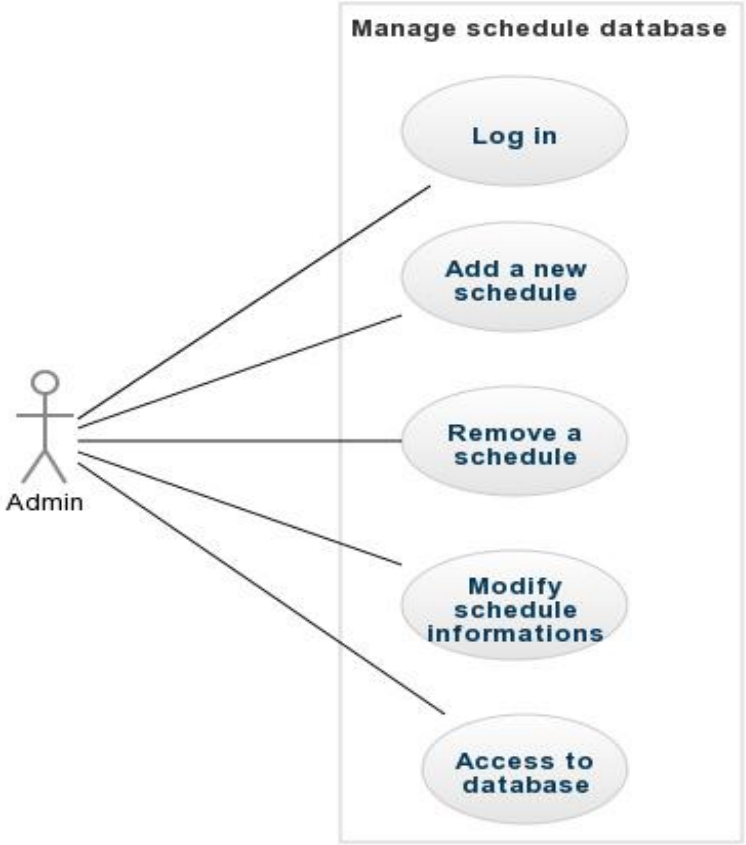

Creating a new user
Request new account
Create a new account
Database
User
Admin

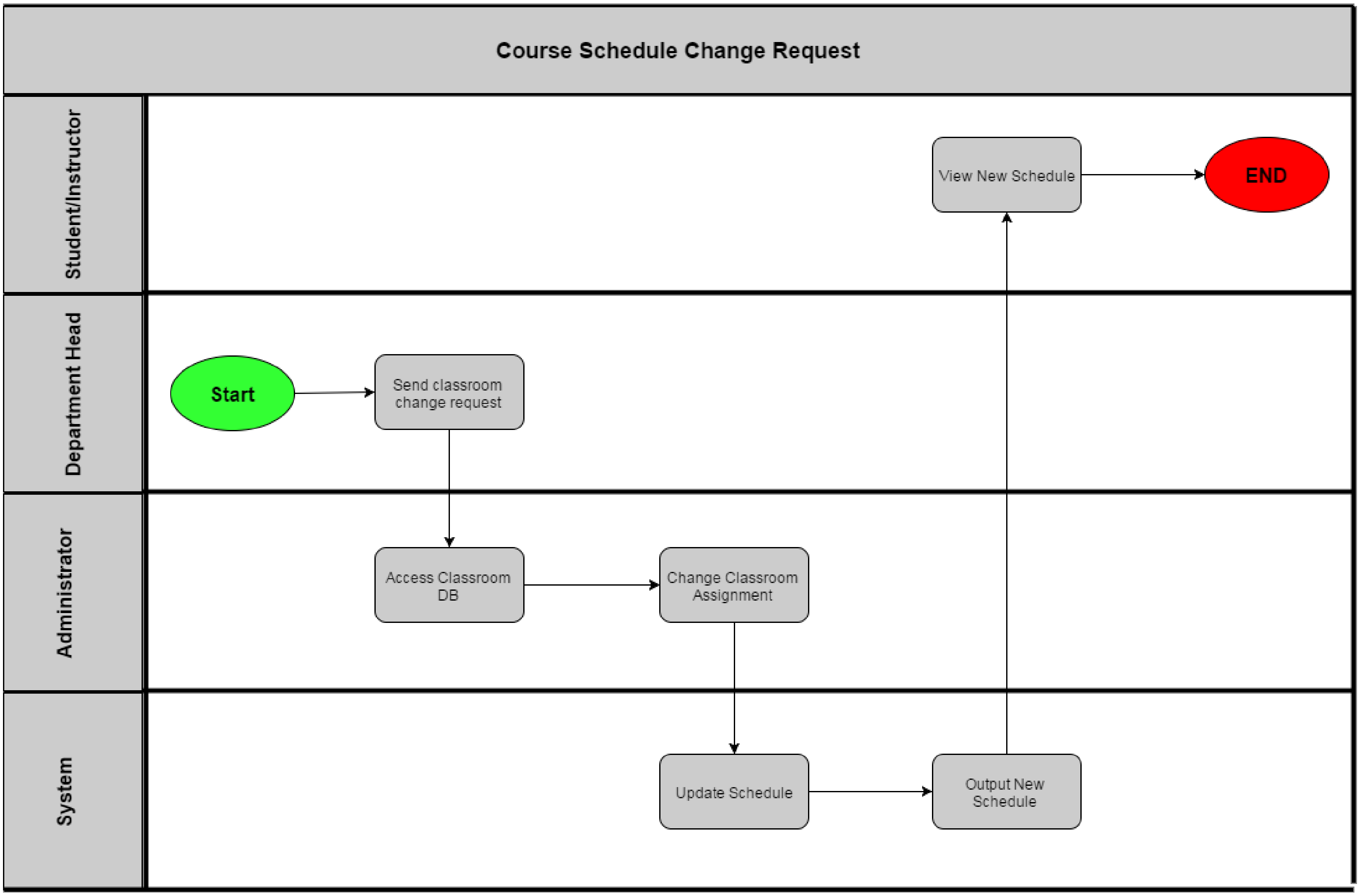

Resetting a password
Log in
Reset passowrd
Update User information into database
User
Admin


Manage classroom database
Log in
Add or delete a classroom
Modify classroom information
Access to database
Admin


Manage courses database
Log in
Add or delete a course
Modify course information
Access to database
Admin


Manage user database
Admin
Log in
Create a new user
Remove a user
Access to database

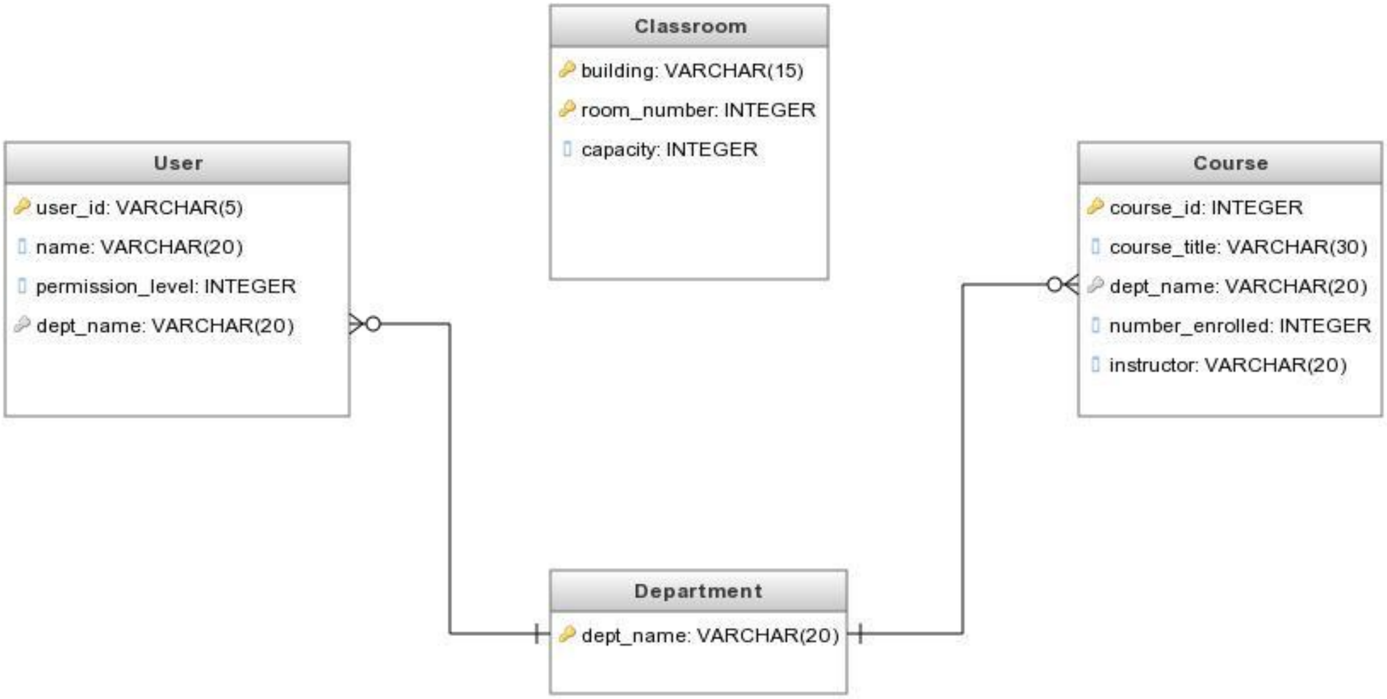

Manage schedule database
Log in
Add a new schedule
Remove a schedule
Modify schedule informations
Access to database
Admin



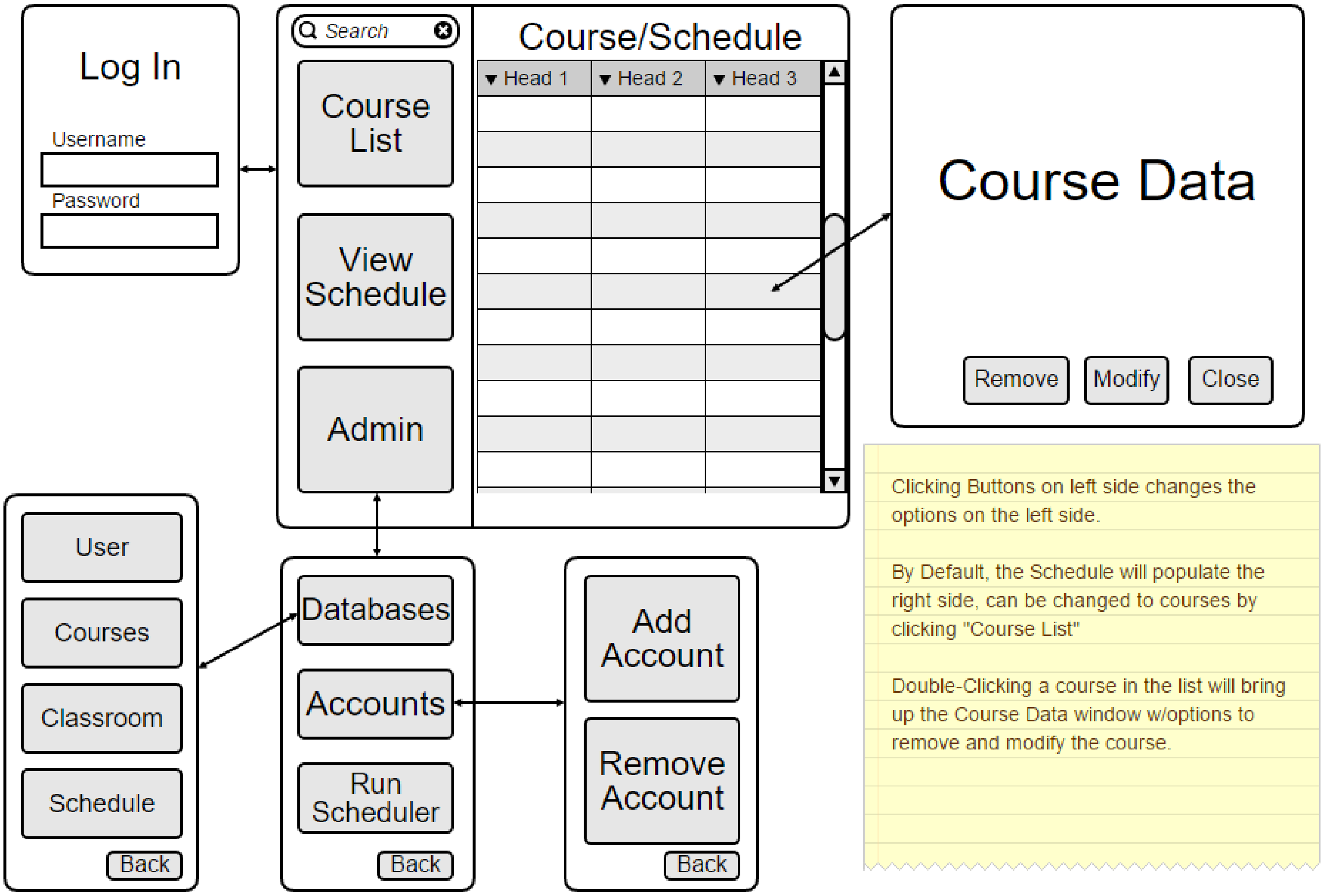
ACTIVITY DIAGRAM:

DATA STORAGE:

We will be using a MySQL database. See the schema below.



UI MOCKUPS:



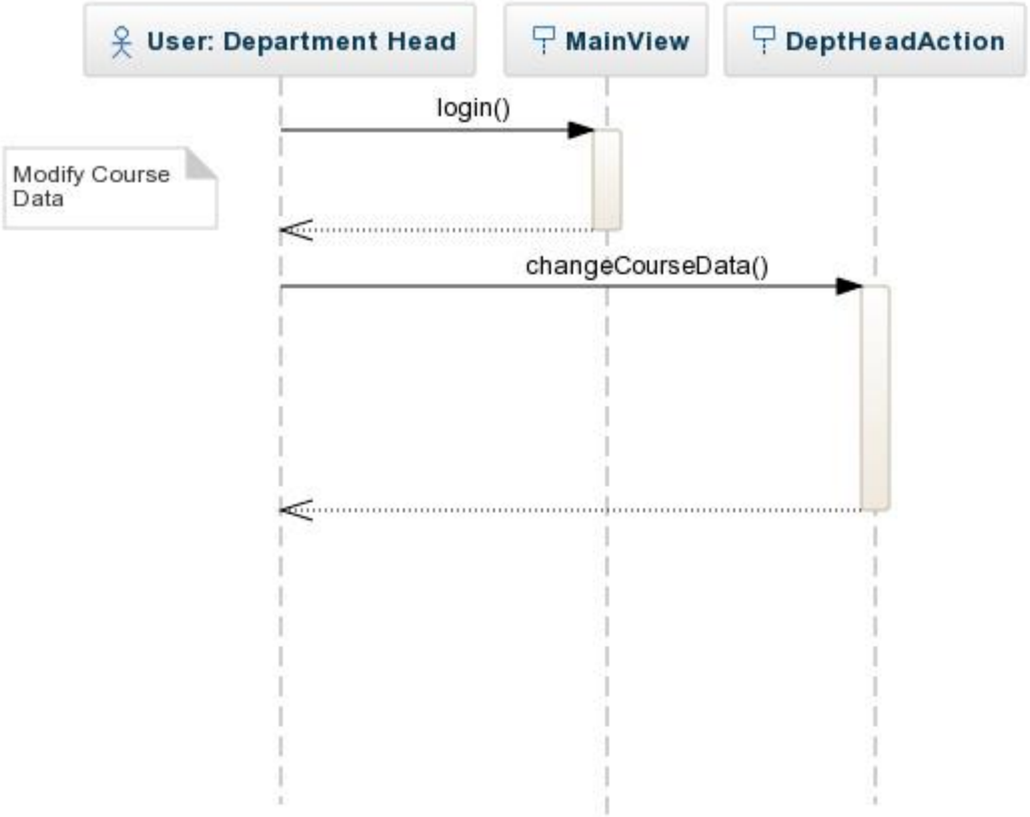
Clicking Buttons on left side changes the options on the left side.

By Default, the Schedule will populate the right side, can be changed to courses by clicking "Course List"

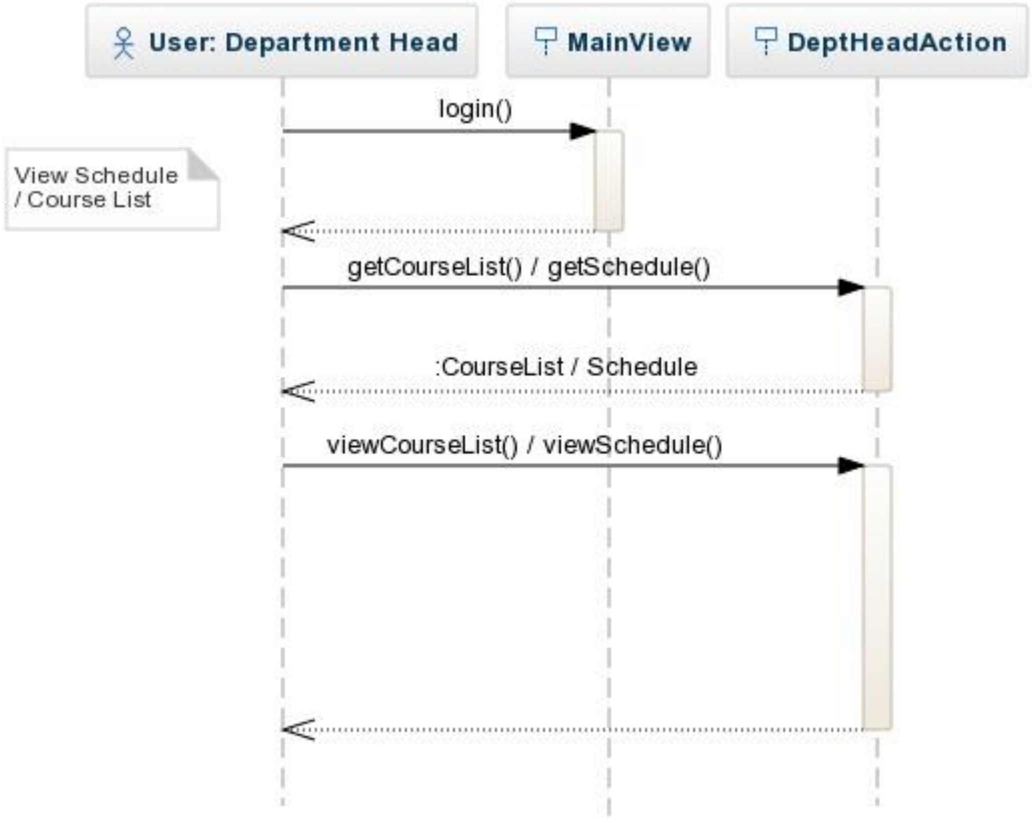
Double-Clicking a course in the list will bring up the Course Data window w/options to remove and modify the course.

USER INTERACTIONS:

1. **Modifying Course List:** Once the Department Head has logged in, a call to getCourseList() is made to populate the course window. The user then finds the course he wishes to change and opens its Data Panel. From there they can click the “Modify” button, making a call to changeCourseData(). Once the function completes the course data is changed.

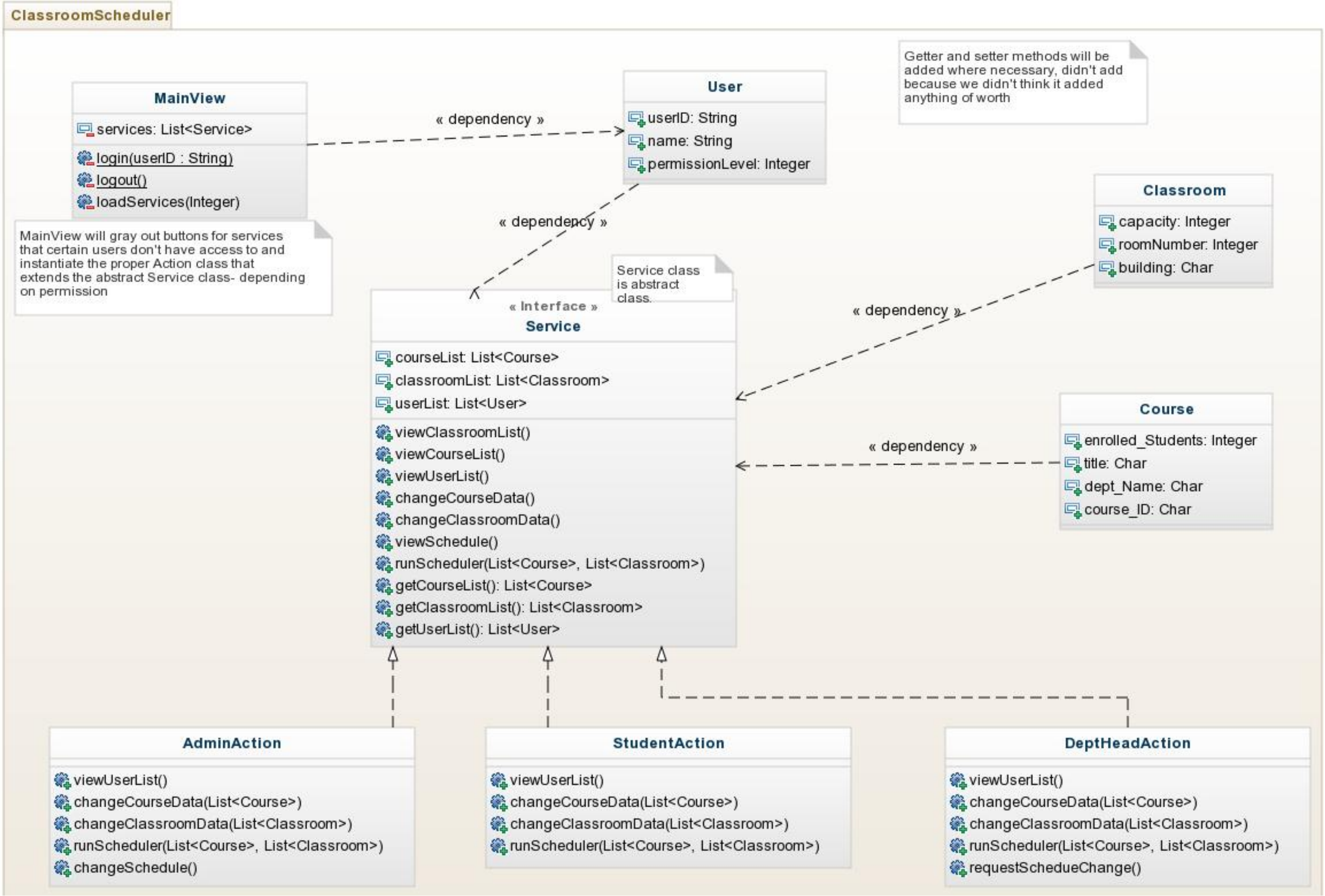


2. **View Schedule/Course List:** Department Heads have to option to view their own course list, as well as viewing the Schedule (once it's been created). To do this, they simply login, then click either the Course List button (calling getCourseList() and viewCourseList()), or the View Schedule button (calling getSchedule() and viewSchedule()).


User: Department Head
MainView
DeptHeadAction
login()
View Schedule / Course List
getCourseList() / getSchedule()
:CourseList / Schedule
viewCourseList() / viewSchedule()

3. **Run Scheduler:** The Admin, once course lists from each department has be populated, can run the scheduler to schedule courses into classrooms. To do this, he simply logs in, clicks the Admin button, the clicks “Run Scheduler”. The program will use data from the Classroom and Courses databases then populate the Schedule.

   TODO: Sequence Diagram



Getter and setter methods will be added where necessary, didn't add because we didn't think it added anything of worth

MainView will gray out buttons for services that certain users don't have access to and instantiate the proper Action class that extends the abstract Service class- depending on permission

Service class is abstract class.

CLASS DIAGRAM: