

CS4400 Database Project

Fall Semester 2016

Version 1.2

SLS (Service, Learn, Sustain)

Look at the last page for edits made for each version. Please read the entire description of the project before starting to work on it.

Purpose of the Project

Analyze, specify, design, implement, document and demonstrate an online system. You are required to use the classical methodology for database development. The system should be implemented using a relational DBMS that supports standard SQL queries. Class administrators will provide you with information about how to access a college- managed MySQL server in order to implement your database and the application. The professors must approve any other alternative implementations. ***In no circumstances can you use a tool that automatically generates SQL or automatically maps programming objects into the database. You also cannot use any other software like Access.*** Ask professors or TAs if you have doubts in which tools/languages/software are allowed.

Project Phases

The three phases of the project cover the following work-processes from the Classical Methodology for Database Development (see notes on T-square under resources). Slides on database design methodology will be useful for phases I and II: All slides have been posted on T-square.

Regrade Policy

Once graded phases and/or quizzes are returned, there is a one-week deadline during which you can contest your grade. This clock starts not when you personally get your returned paper, but when the papers are returned to the class.

Groups

Project groups may have 3 or 4 members. Groups of more than 4 or less than 3 will **NOT** be allowed. You are allowed to form groups across the two sections (A & B) of the class. A group may remove a member from further participation in the group when Phase I is turned in or when Phase II is turned in. A written notification with a proper justification must be provided to the professor and the head TA at that time in hard-copy form.

Deliverables

Phase I (submit on TSquare and bring a hard copy to class)

Deadline: September 29

The deliverables include:

1. A cover page. Your cover page **MUST** include all information on the template. (Template is under Resources/Project)
2. Enhanced Entity Relationship (EER) Diagram
3. Information Flow Diagram
4. A list of logical constraints (at least 3). **Constraints are things that mentioned in the description but cannot be shown on the EER.** You are required to include at least three constraints, although a fully-specified system will probably have more than that.
5. Any assumptions made with explanations.

Notes:

1. The EER must capture the functionalities of the system as many as possible whenever applicable, i.e. total participation, super/sub class, weak entities.
2. The design of your system must have all functionalities. You are allowed to make up additional assumptions as long as they do not conflict with the specified constraints and requirements. You must list all your assumptions; otherwise TA would mark your ER diagram wrong since they would not be able to know you had made your own assumptions.
3. Constraints that can be specified directly using ER notation will not count towards the three required. Constraints related to data type or value are not accepted as constraints.

Each group needs to turn in one hard copy (only one for the entire group), and each group member should upload an electronic copy on T-Square individually. You will receive -5 penalty if you do not submit an electronic copy. **Please write down your Group Number clearly on cover page.** If you do not know your group number, please email the head TA.

Phase II (submit on TSquare and bring a hard copy to class)

Deadline: Oct 27

The deliverables include:

1. A cover page same as phase1

2. Copy of the ER Diagram (either from phase I (with any revisions) or from the solution provided)
3. Relational Schema Diagram (Identify primary and foreign keys and show referential integrity using arrows)
4. Create Table statements, including domain constraints, integrity constraints, primary keys, and foreign keys.

Note:

1. **Only one hard copy** should be turned in for the entire group, and each group member should upload an electronic copy on T-Square **individually**. **You will receive -5 penalty if you do not submit an electronic copy. Please write down your Group Number clearly on cover page.**

Phase III (Submit on TSquare)

TSquare Submission Deadline: Dec 4

Project Demo Dates: Dec 5 & 6

(It is due on Dec 4. You should not modify your project after Dec 4)

The electronic deliverables include:

1. A cover page same as phase1 & 2.
2. A text file with all SQL statements for each task (follow the template in the phase II design methodology)
Note: A set of SQL statements may be required in order to complete one task. However, in such cases, the last SQL statement should show the output according to the specification. Views and nested queries may be used to support the tasks.
3. For heavy weight option, you also need to submit your source code.

Note: Prior to the demo, the TAs will give guidelines for populating the database with data. The database has to be populated with this data set prior to the demo.

Each group member should upload an electronic copy on T-Square individually. You will receive -5 penalty if you do not submit an electronic copy.

On demo day:

Bring your laptop and make sure you have a text file on your laptop with all your SQL queries just in case your application does not work. More details about demo will be discussed later this

semester.

Grading

The project will consist of three phases (deliverables) as well as a final demo to the TA.

Phase I and Phase II of the project are each worth 10% of your final grade.

Phase III (20% for heavy-weight or 5% for light-weight, depending on option):

Heavy Weight Option (20 %): The students would be required to use the embedded SQL feature of MySQL which allows you to embed SQL statements in a standalone application.

Light Weight option (5%): The students would be required to demo the SQL queries on the MySQL console. Those who choose the light weight option would be required to take the Final exam.

Note that you can always change your option until the demo starts. Once TA starts to demo your project, you cannot change heavy-weight option to light-weight or vice versa.

Final Exam (15%): This would be only taken by students who have opted for the lightweight phase III. Under no circumstances would a heavy weight option student be allowed to take the Final.

Project

For this project, you will create a tool that stores projects and courses which are related to SLS (Service, Learn, Sustain).

The following sections contain a functional description of the system along with some mockup screens. Each section would explain a particular functionality and then present an example screen about it. **You don't have to follow the UI designs, but your program needs to support all the functionalities.** These mockups are just for helping you to understand all the functionalities. A complete reorganization of the user interface is permissible as long as your application supports all the functionality listed below. The sections have been grouped by customer's functionalities and managers' functionalities.

For heavy option, you may implement the project as a traditional standalone application (e.g., using Java GUIs) or as a web application (e.g., using a web scripting language like PHP). There is no restriction on the choice of language (e.g., Java, Python, Javascript). We will also send an announcement about which languages/tools/software/platforms are allowed later this semester. (Do ask the professors for permission if in doubt.)

1. Log In

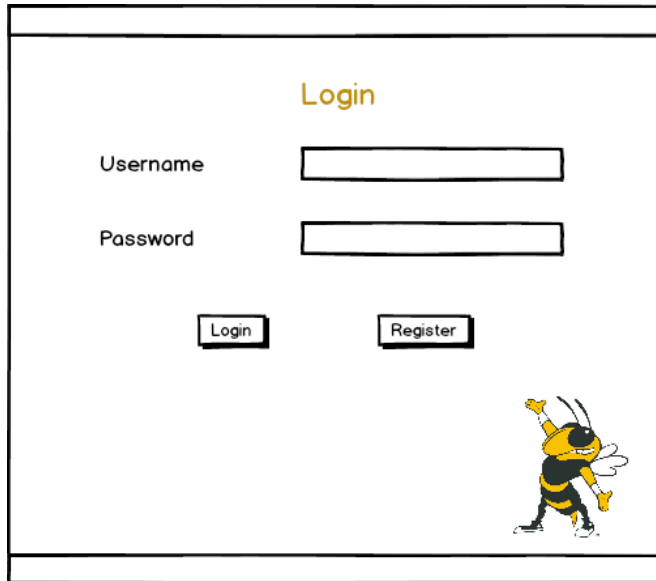
A mockup of a login screen. At the top center, the word "Login" is written in a yellow, sans-serif font. Below it, there are two input fields: the first is labeled "Username" and the second is labeled "Password". Both labels are in a black, sans-serif font. Below the input fields, there are two buttons: "Login" on the left and "Register" on the right. Both buttons are black with white text. In the bottom right corner, there is a cartoon bee character with a yellow and black striped body, wearing white gloves and shoes, and holding a white object.

Fig 1: Log in

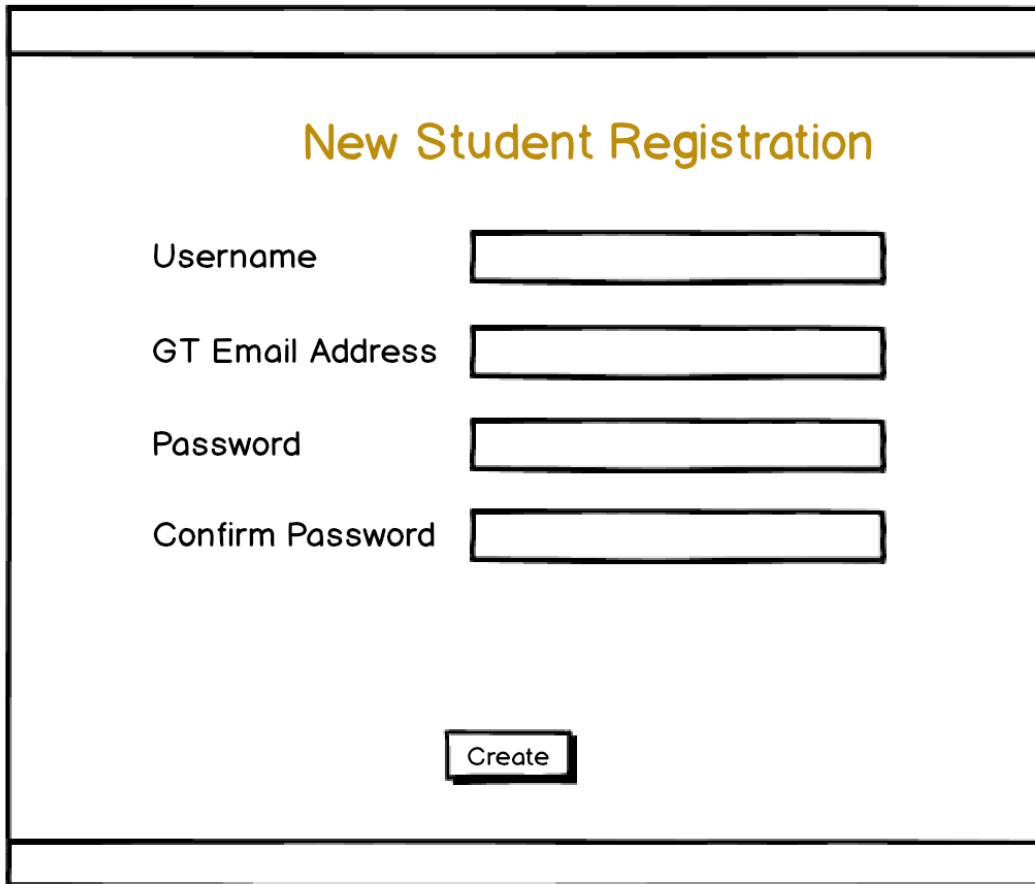
Fig 1 shows the login screen. All users must login before using this application. There are two types of users: students and admin. To login, a valid username and password combination is required. If users provide invalid login credentials, an error message should be shown on the screen.

If users do not have account yet, they can click on the register button to create an account. (this is for students only)

Note:

1. Username is **unique** for every user.
2. A user must be either a student or an admin.
3. Since students and admin share the same login screen, you need to check if the user is a student or an admin.

2. New User Registration



The image shows a web form titled "New Student Registration" in orange text. The form is enclosed in a black border. It contains four input fields, each preceded by a label: "Username", "GT Email Address", "Password", and "Confirm Password". Each label is aligned to the left of its corresponding input field. Below the input fields, centered, is a button labeled "Create".

New Student Registration

Username

GT Email Address

Password

Confirm Password

Fig 2: New Student Registration

After clicking register button in Figure 1, users will be directed to this new student registration page.

To register a student account:

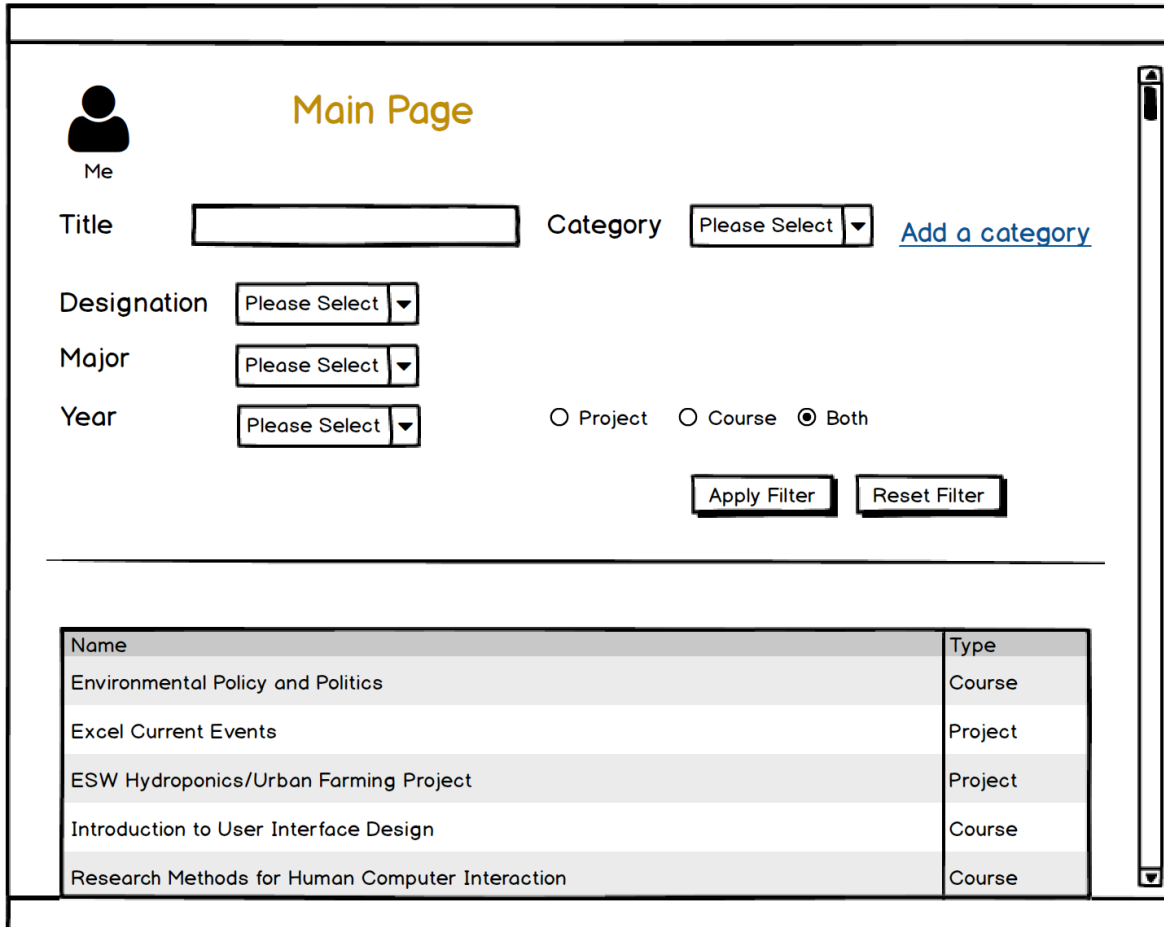
Only GT students can use this application, so students must enter their official GT email address to register accounts. Each student can only register one account, and hence you need to check if the GT email address is already in the database.

An error message should be displayed if any requirement is not met.

We assume all admin accounts have already been stored in the system, so no registration is needed for admin.

Student Functionalities

1. Main Page



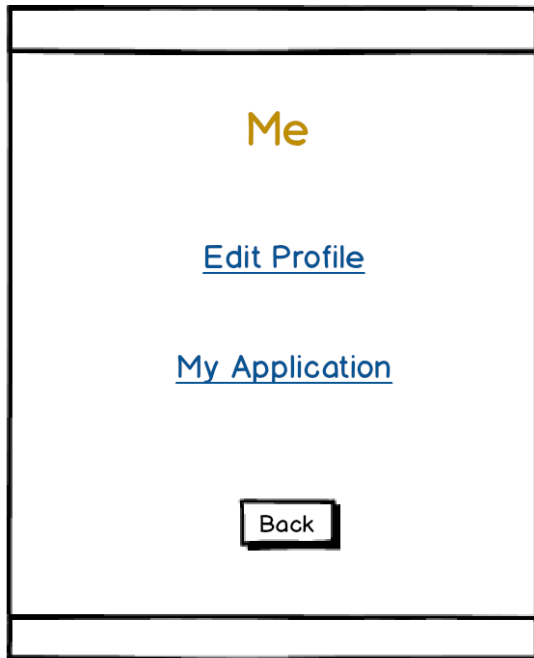
The Main Page UI mockup features a header with a user profile icon labeled 'Me' and the title 'Main Page'. Below the header, there are input fields for 'Title' and 'Category' (a dropdown menu), along with a link 'Add a category'. Further down, there are dropdown menus for 'Designation', 'Major', and 'Year'. To the right of these are radio buttons for 'Project', 'Course', and 'Both' (which is selected). At the bottom of the filter section are 'Apply Filter' and 'Reset Filter' buttons. Below a horizontal separator is a table with two columns: 'Name' and 'Type'.

Name	Type
Environmental Policy and Politics	Course
Excel Current Events	Project
ESW Hydroponics/Urban Farming Project	Project
Introduction to User Interface Design	Course
Research Methods for Human Computer Interaction	Course

Figure 3: Main Page

After logged in as students, students would be taken to this Main Page where they can browse and search projects and courses. Search and filter function will be discussed later.

2. Me

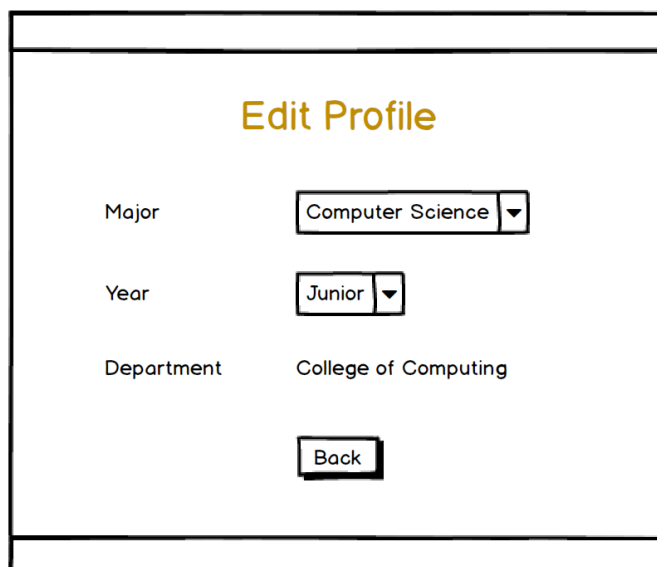


A wireframe of a web page titled "Me". The page has a header bar at the top and a footer bar at the bottom. The main content area contains the word "Me" in a large, bold, orange font. Below it are two blue, underlined links: "Edit Profile" and "My Application". At the bottom of the main content area is a rectangular button labeled "Back".

Figure 4: Me

On the left upper corner of Main Page, there is a Me button. After clicking on that button, students will be directed to Me page, where they can choose to edit their profile or view their applications.

3. Edit Profile



A wireframe of a web page titled "Edit Profile". The page has a header bar at the top and a footer bar at the bottom. The main content area contains the title "Edit Profile" in a large, bold, orange font. Below the title are three form fields: "Major" with a dropdown menu showing "Computer Science", "Year" with a dropdown menu showing "Junior", and "Department" with a text input field containing "College of Computing". At the bottom of the main content area is a rectangular button labeled "Back".

Figure 5: Edit Profile

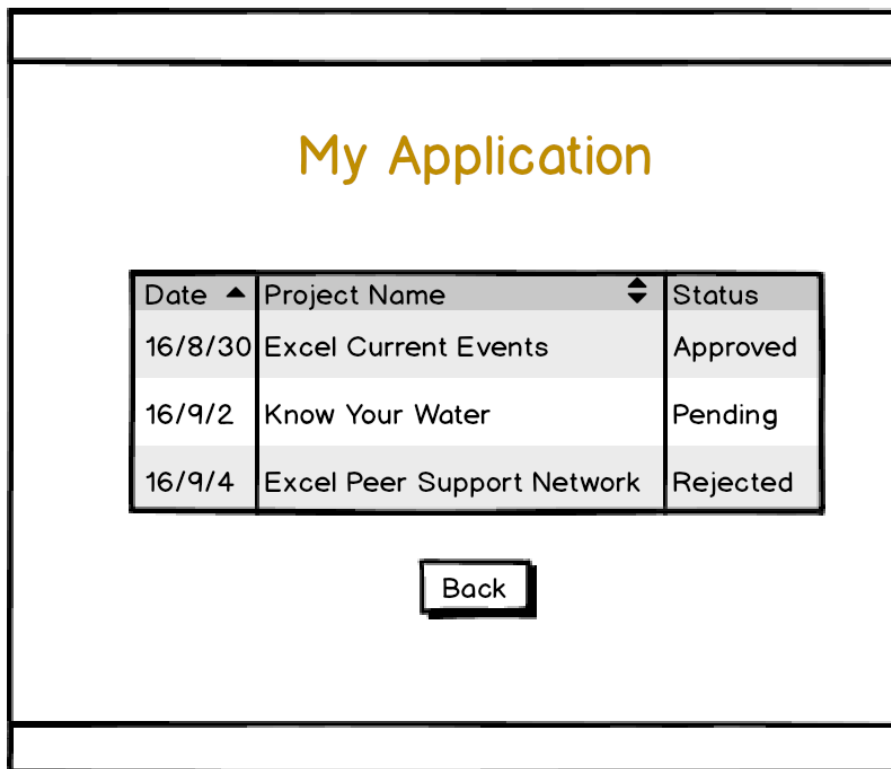
Students can change their major and year on this page. (You need to use dropdown, not textfield.) The system will update department based on the major. New student needs to complete this step before applying to join a project team. (We will discuss the application part later.)

We assume one student can only have one major, and one major belongs to only one department.

Major: use GT undergrad major data (You need to store these data in the database. You also need to store the departments that the majors belong to)

Year: choose one from freshman, sophomore, junior, senior

4. My Application



Date ▲	Project Name ▼	Status
16/8/30	Excel Current Events	Approved
16/9/2	Know Your Water	Pending
16/9/4	Excel Peer Support Network	Rejected

[Back](#)

Figure 6: My Application

On this page, students can view their applications. More information about applications will be discussed later.

5. View and Apply Project

Know Your Water

Advisor: Neha Kumar (neha.kumar@cc.gatech.edu)

Description:
This project will allow students to be part of a large, crowd-sourced study – at little cost to themselves – to contribute to a knowledge bank of how different communities treat and track their water quality. If you are interested in participating in this study, please let us know.

Designation: Sustainable Community

Category: Sustainable, crowd-sourced

Requirements: CS students only; Senior only

Estimated number of students: 40

[Back](#)[Apply](#)

Figure 7: View and Apply Project

Students can click on a project on the Main Page to view its details. A project has its name, advisor information, description, designation, category, requirements and estimated number of students.

Note: Students cannot reapply if they got rejected.

Data:

Name: project name

Advisor information: name and email address (One advisor for each project; Advisor is not a user)

Description: A short paragraph that describes this project

Category: One or more categories chosen from: *“computing for good”, “doing good for your neighborhood”, “reciprocal teaching and learning”, “urban development”, “adaptive learning”,*

“technology for social good”, “sustainable communities”, “crowd-sourced” and “collaborative action”

Designation: *“Sustainable Communities”* or *“Community”*

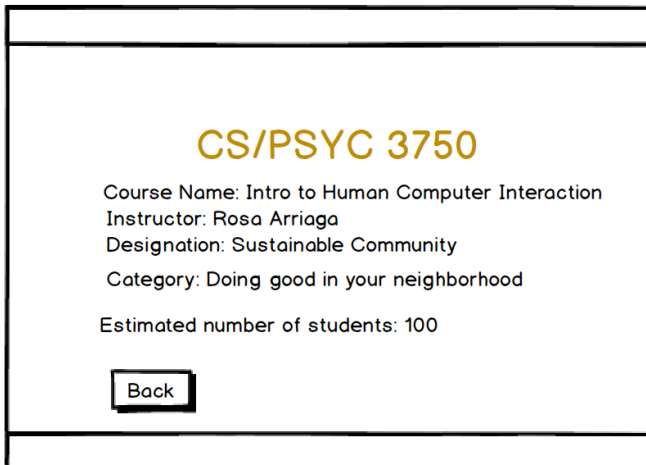
Requirements: major restriction, year restriction, department restriction

Note: You can assume requirements will always look like “xxx students only.” For instance: major restriction could be: CS students only; year restriction: Junior only; department restriction is “COC students only”.

If students are interested in this project, they can apply to join the project team by clicking on ‘Apply’ button. If students’ major/year/department cannot meet the requirements, an error message should be displayed.

An admin will review the application and make decisions.

6. View Course



CS/PSYC 3750

Course Name: Intro to Human Computer Interaction
Instructor: Rosa Arriaga
Designation: Sustainable Community
Category: Doing good in your neighborhood
Estimated number of students: 100

[Back](#)

Figure 8: View Course

Similar to view project, students can also click on a course on Main Page to view its details. However, they cannot apply to take the course.

Everything is the same as project, except:

- Course has a unique course number (For instance CS4400)

- Course has instructors rather than advisors, and you don’t need to store their email address.

- Course does not have description nor requirement.

7. Search/Filter

Students can use the search bar and filter tool on Main Page to find a course/project.

Title: search project name/ course name.

Category: choose one or multiple categories from a dropdown menu. (You must use a dropdown)

Designation: choose designation from dropdown.

Major: choose major from dropdown.


Year: choose year from dropdown. (For this one, you do not to pull data from the database to fill the dropdown options. You are allowed to hardcode them. We recommend you to use dropdown for Year, because it would be easier for you to apply filter later. However, other than year, you cannot hardcode options for dropdown.)

Note: Students can leave any of these options blank.

Example:

Let's say a CS junior is interested in "computing for good "and "doing good for your neighborhood", and he wants to join a project team which designation is community. So he chooses CS, junior, community, computing for good and doing good for your neighborhood, clicks project radiobutton and applies filter.

Main Page after applying filter


Me

Main Page

Title

Category

Computing for good

Designation

Community

Doing good for your[Add a category](#)

Major

CS

Year

Junior

☒ Project ☐ Course ☐ Both

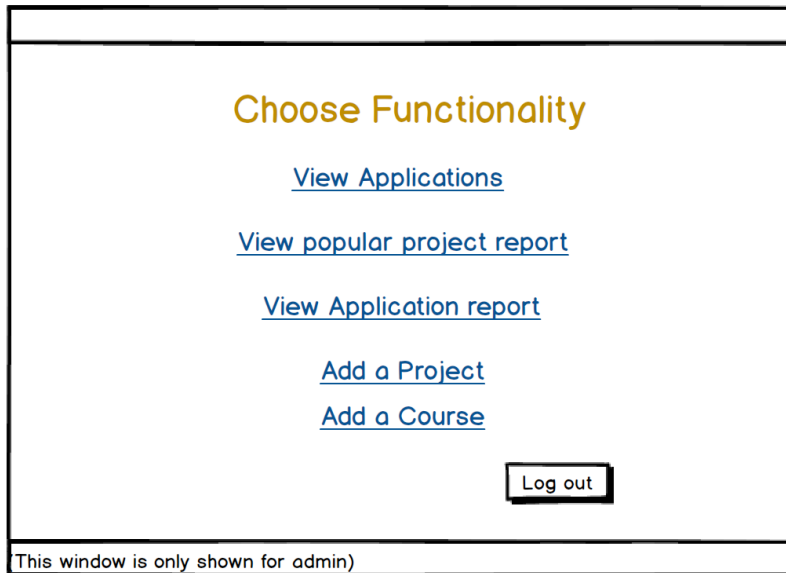
Apply Filter

Reset Filter

Figure 9: Search

Admin Functionalities

1. Choose Functionality (admin view)



A screenshot of a web application window titled "Choose Functionality". The window has a white background with a black border. At the top, the title "Choose Functionality" is displayed in a large, bold, orange font. Below the title, there are five blue, underlined links: "View Applications", "View popular project report", "View Application report", "Add a Project", and "Add a Course". At the bottom right, there is a black button with the text "Log out" in white. At the bottom left, there is a small black box containing the text "This window is only shown for admin)".

Choose Functionality

[View Applications](#)

[View popular project report](#)

[View Application report](#)

[Add a Project](#)

[Add a Course](#)

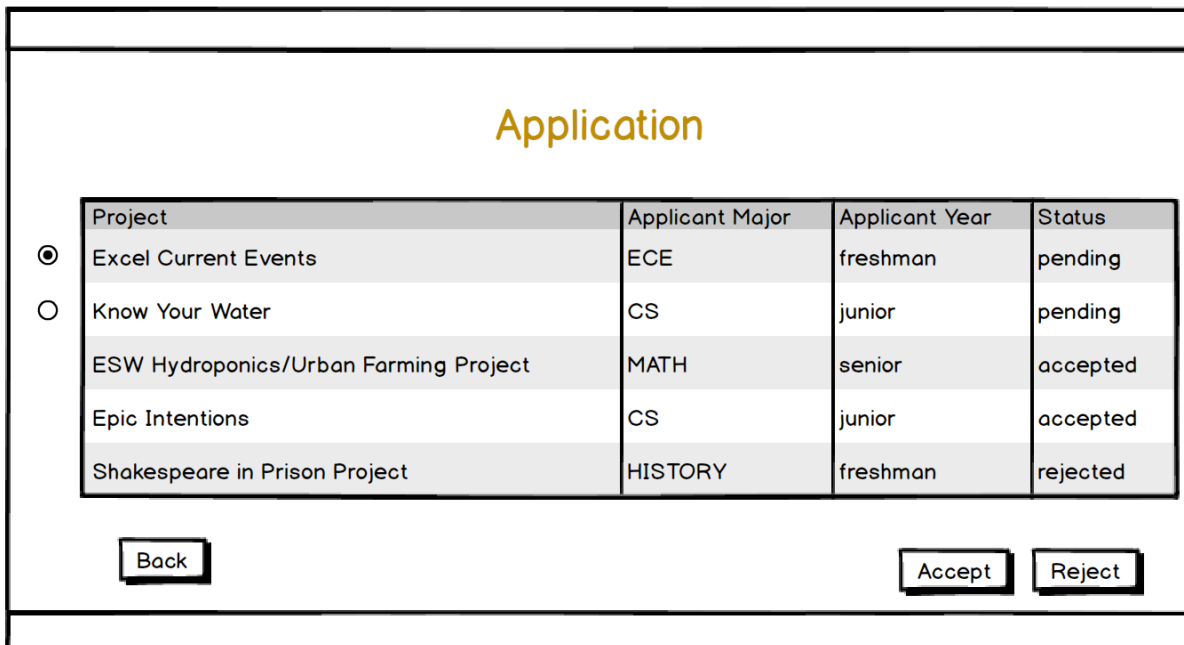
Log out

This window is only shown for admin)

Figure 10: Choose Functionality (admin view)

If users log in as admin, they will be taken to this window where they can choose to view applications, view popular project report or view application report.

2. View Applications



A screenshot of a web application window titled "Application". The window has a white background with a black border. At the top, the title "Application" is displayed in a large, bold, orange font. Below the title, there is a table with four columns: "Project", "Applicant Major", "Applicant Year", and "Status". The table has five rows of data. To the left of the table, there are two radio buttons: the first is selected (indicated by a filled circle) and the second is not (indicated by an empty circle). Below the table, there are three buttons: "Back", "Accept", and "Reject".

Application

Project	Applicant Major	Applicant Year	Status
Excel Current Events	ECE	freshman	pending
Know Your Water	CS	junior	pending
ESW Hydroponics/Urban Farming Project	MATH	senior	accepted
Epic Intentions	CS	junior	accepted
Shakespeare in Prison Project	HISTORY	freshman	rejected

Back

Accept

Reject

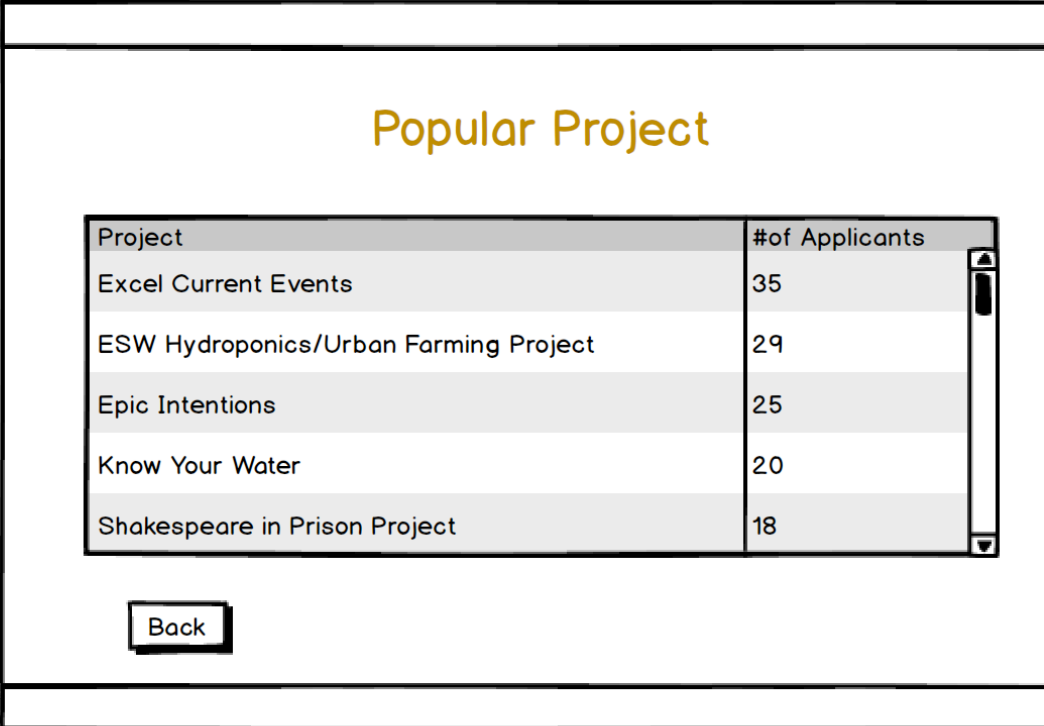
Figure 11: View Applications

Admin can view all applications on this page. For each application, it should have project name, applicant's major, applicant's year and application status.

Status: *pending, accepted, rejected*

If status is *pending*, that means admin has not made a decision yet. To accept or reject an application, admin could click on the radio button besides the application, and then choose "accept" or "reject".

3. View Popular Project Report



Project	#of Applicants
Excel Current Events	35
ESW Hydroponics/Urban Farming Project	29
Epic Intentions	25
Know Your Water	20
Shakespeare in Prison Project	18

Back

Figure 12: View Popular Project Report

This report shows the top 10 projects with most applications.

Hint:

1. Do you need to store reports in the database?

4. View Application Report

<h3>Application Report</h3> <p>152 applications in total, accepted 89 applications</p> <table><tr><th>Project</th><th>#of Applicants</th><th>accept rate</th><th>top 3 major</th></tr><tr><td>Epic Intentions</td><td>25</td><td>72%</td><td>CS/MATH</td></tr><tr><td>Shakespeare in Prison Project</td><td>18</td><td>72%</td><td>HIST/LMC/ENGL</td></tr><tr><td>ESW Hydroponics/Urban Farming Project</td><td>29</td><td>62%</td><td>EE/ARCH</td></tr><tr><td>Excel Current Events</td><td>35</td><td>57.1%</td><td>ECE/CM</td></tr><tr><td>Know Your Water</td><td>20</td><td>30%</td><td>CS</td></tr></table> <div>Back</div>				Project	#of Applicants	accept rate	top 3 major	Epic Intentions	25	72%	CS/MATH	Shakespeare in Prison Project	18	72%	HIST/LMC/ENGL	ESW Hydroponics/Urban Farming Project	29	62%	EE/ARCH	Excel Current Events	35	57.1%	ECE/CM	Know Your Water	20	30%	CS
Project	#of Applicants	accept rate	top 3 major																								
Epic Intentions	25	72%	CS/MATH																								
Shakespeare in Prison Project	18	72%	HIST/LMC/ENGL																								
ESW Hydroponics/Urban Farming Project	29	62%	EE/ARCH																								
Excel Current Events	35	57.1%	ECE/CM																								
Know Your Water	20	30%	CS																								

Figure 13: View Application Report

This report shows projects sorted by acceptance rate. It should also show number of applications and the top 3 majors of applicants. On the very top of the report, it should have the total number of applications and how many of them have been accepted.

5. Add a Project

Add a Project

Project Name:

Advisor:

Advisor Email:

Description:

Category: [Add a new category](#)

Designation:

Estimated # of students:

Major Requirement:

Year Requirement:

Department Requirement:

Figure 14: Add a Project

Admin can add projects to the database. All fields (except requirement) need to be filled.

6. Add a Course

Add a Course

Course Number:

Course Name:

Instructor:

Designation:

Category: [Add a new category](#)

Estimated # of students:

Figure 15: Add a Course

Admin can add courses to the database. All fields need to be filled.

Date	Version	Note
9/20	1.1	-Added functionalities: Admin can add projects and courses -Added regrade policy -Deleted requirements from course -Corrected some typos
9/27	1.2	-Fixed UI