Study Group Finder Application Design Document

Group Members: Nathan Mitchell, Christian Blankenback

APIs

Authentication API

- Endpoints:
    - POST /login - Authenticate a user.
    - POST /register - Register a new user.

User API

- Endpoints:
    - GET /user/profile - Retrieve the profile of the authenticated user.
    - PUT /user/profile - Update user profile information.

Study Group API

- Endpoints:
    - POST /studygroup/create - Create a new study group.
    - GET /studygroup/search - Find study groups based on criteria.
    - GET /studygroup/mygroups - Retrieve groups that the user is a part of.

Course API

- Endpoints:
    - GET /courses - List all courses.
    - POST /courses - Add a new course.

Beans (Entity Classes for JPA Mapping)

UserEntity

- Fields:
    - Long id
    - String email
    - String password
    - String name

StudyGroupEntity

- Fields:
    - Long id

- String name
- String weekday
- String time
- String location
- Long ownerId
- CourseEntity course

CourseEntity

- Fields:
  - Long id
  - String name

Web Pages

Color pallet used:

Start

- More about the website, Login Button and a Create Account button that link to their pages respectively.

Login

- Email and password fields, login button, link to Create Account page. Login button links to index page

Create Account

- Fields for name, email, password, and a register button. A link to login in case the client clicks the wrong button from the start page.

Index

- Page containing the navigation buttons to finding study groups or to create a new study group.

MyFeed

- Page containing a list of all study groups that have been created on the application, each will contain the course that the group is for along with the number of users registered in that group. From this page users will be able to click a join button to join the group.

CreateStudyGroup

- Page used to create a new study group for others to join, containing fields name, week day, time, location and course.

CreatedStudyGroup

- Page containing a list of all study groups that the user has created.

JoinedStudyGroup

- Page containing a list of all study groups that the user has joined.

Search Results

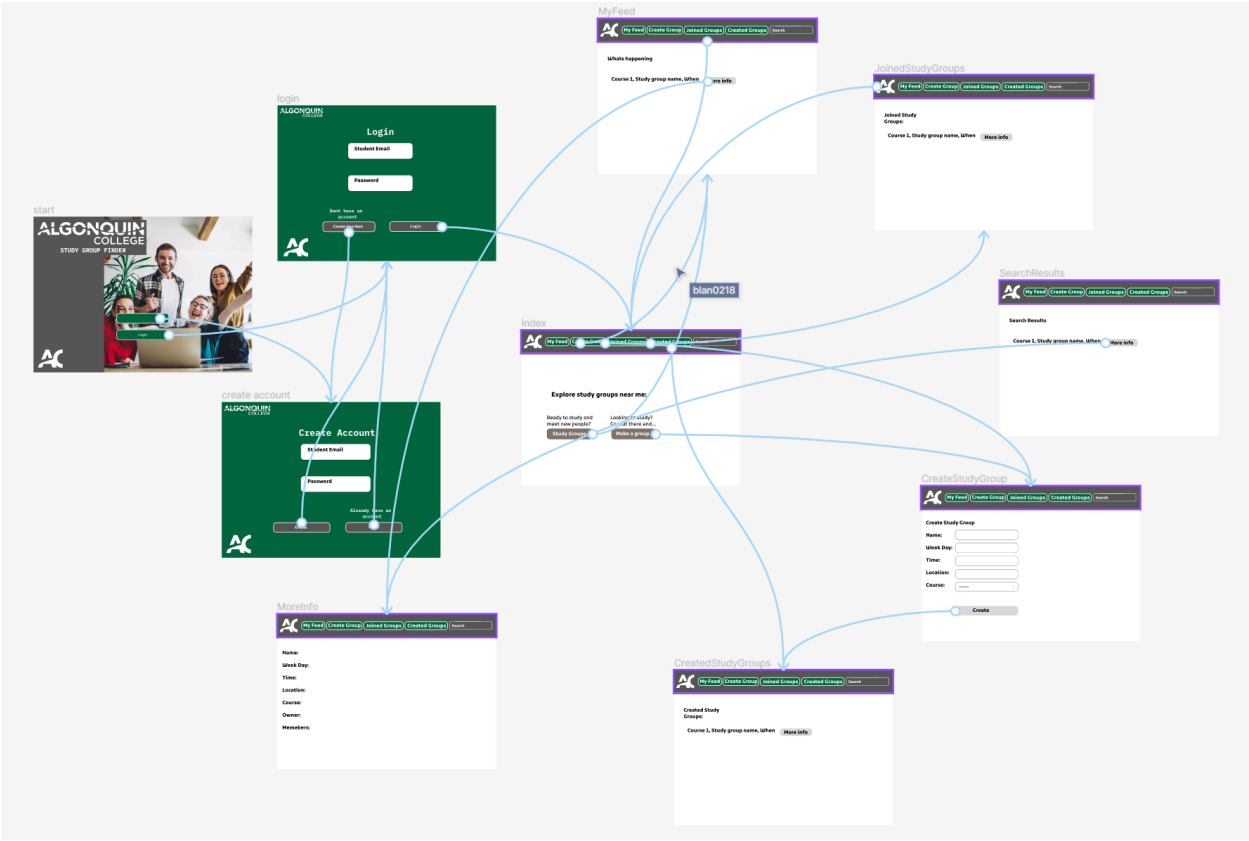
- Window containing the study groups that match the searched value.

More Info

- Window containing more information on the application along with some helpful tips to get started.

Wireframes (Textual Description)

- Login Page: A simple interface with the application's logo, fields for email and password, a login button, and a text link to navigate to the registration page.
- Registration Page: Similar to the login page but includes additional fields to input the user's full name and a password confirmation field.
- Create Group Page: A form with dropdowns for selecting a course, input fields for group name, location, and a date-time picker for scheduling.
- Find Group Page: A search bar at the top with advanced filters that dynamically update a list of study groups below.
- My Groups Page: A dashboard view listing all groups the user is a part of, with options to view details or leave the group next to each listing.



ERD

student table:

- Contains details of students.
- Fields include studentID (Integer), studentEmail (Varchar of length 50), Studentpassword (Varchar of length 50), and studentName (Varchar of length 50).
- studentID is the primary key.

studygroup table:

- Contains details about study groups.
- Fields include SGID (Integer), SGName (Varchar of length 50), SGWeekDay (Varchar of length 50), SGTime (Varchar of length 50), SGLocation (Varchar of length 50), OwnerID (Integer), and a foreign key course_CourseID (Integer) that references the CourseID from the course table.
- SGID is the primary key.

course table:

- Contains details of courses.
- Fields include CourseID (Integer) and CourseName (Varchar of length 50).
- CourseID is the primary key.

studentstudygroup table (associative table):

- Represents a many-to-many relationship between students and study groups.

- Fields include student_studentID (Integer) and student_group_SGID (Integer).
- Both fields together form a composite primary key and each field is a foreign key referencing studentID from the student table and SGID from the studygroup table, respectively.

studentcourses table (associative table):

- Represents a many-to-many relationship between students and courses.
- Fields include course_CourseID (Integer) and student_studentID (Integer).
- Both fields together form a composite primary key and each field is a foreign key referencing CourseID from the course table and studentID from the student table, respectively.