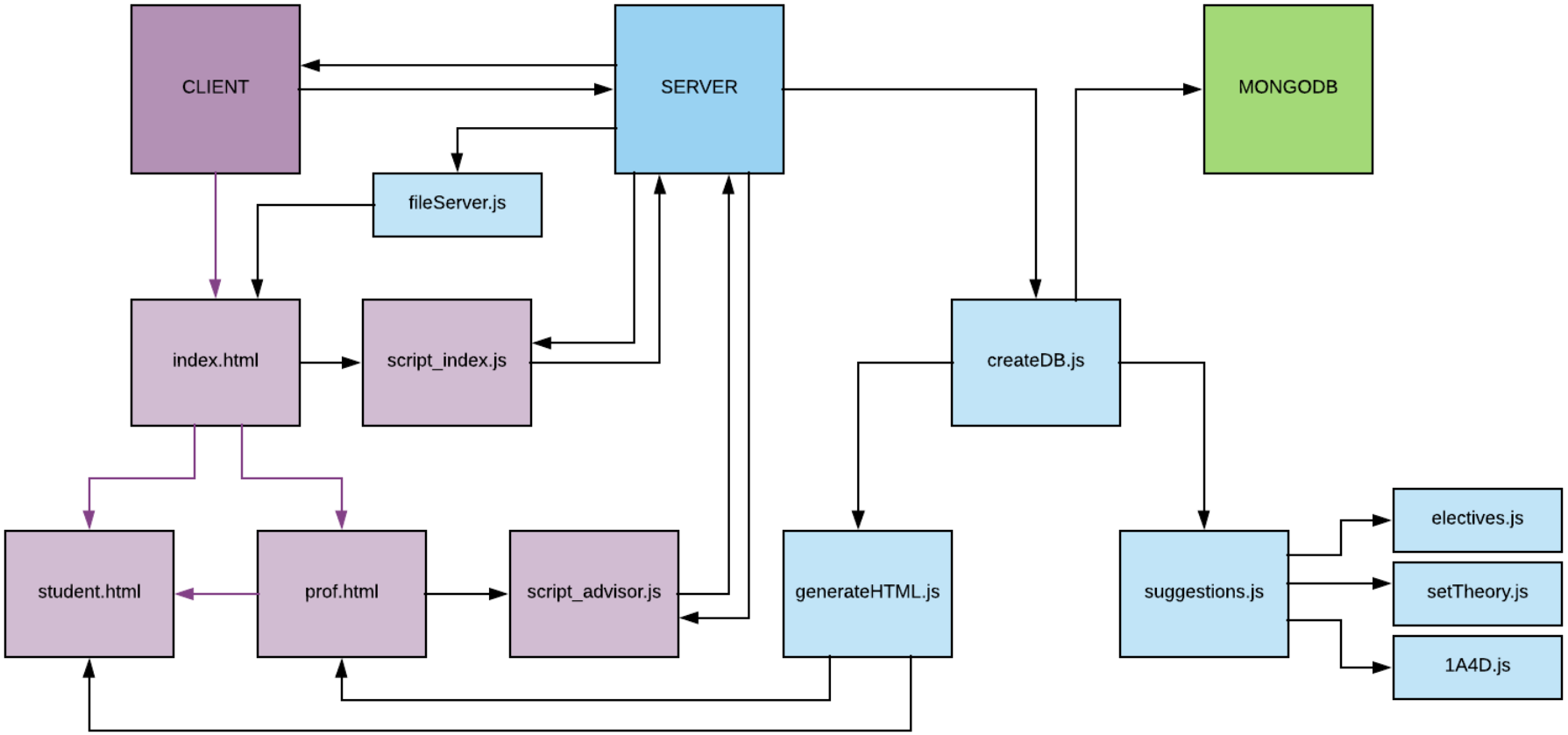
CLIENT
SERVER
MONGODB
fileServer.js
index.html
script_index.js
createDB.js
student.html
prof.html
script_advisor.js
generateHTML.js
suggestions.js
electives.js
setTheory.js
1A4D.js

CSC 220 Final Project: CS Online Major Advisor
By: Lorraine Lyu, Rebecca Rohrlich, and Alex Larosee

data.js

- Not meant to run. Before running the server package, open a new window in terminal, start a mongo shell, copy and paste this code in the shell.
- This creates the database with the student data that will be accessed by the code

server.js

- Main program, creates server and defines request handler
- Modules: http, url, createDB.js, fileserver.js, generateHTML.js
- Processes URL to determine how to handle request and uses modules to complete requests accordingly
    - Serves static homepage file index.html if requested or if nothing specified using fileserver.js module.
    - Checks if user is requesting to get the advisor or student page and sends request to createDB.js module.
    - Checks if user is requesting to filter data and sends request to createDB.js module.

createDB.js

- MongoDB interface module
- Modules: mongodb, generateHTML.js, suggestions.js
- Creates a new MongoClient to access the majorAdvisor database upon being required in the main program
- searchPerson(identity, input, keyword, httpRes)
    - connects to the database
    - accesses the “students” collection
    - checks identity of the client to determine whether to check the advisor field or the student field of the documents.
    - Searches the collection using find(inputRegExp) and parses the matching documents into an array
    - Informs client if no results are found
    - If student identity, calls suggestedCourses function in the suggestions module to determine and add the course suggestions to the student object, sends request to generateStudentPage in generateHTML.js to create and get the student html page.
    - If professor identity, checks if the user has requested a filter of the student data by including a key to indicate the type of filter. Performs the requested type of filter and returns a list of matching documents. Sends server response as the stringified list of matching documents.
    - If professor identity but no keyword, sends request to generateProfessorPage in generateHTML.js to create and get the professor html page

Suggestions.js

- Uses course list to determines whether a student has satisfied the CS major requirements, check if student is in good standing, and make suggestions as to what courses to take.
- Modules: electives.js, setTheory.js,1A4D.js
- algorithm for determining the satisfaction of the electives requirements is written in the electives.js module, algorithm for the 1A4D requirements is written in the 1A4D.js module.
- applies set theory, using setTheory.js module, to find all possible courses that work to satisfy major requirements while accounting for the fact that many courses can satisfy multiple requirements
- isOnFire(studentObj)
    - compares the number of requirements left for the student to complete to the number that is deemed reasonable for a given class year
    - checks student onhold field
    - returns true if too many outstanding requirements or if student is on hold
- suggestedCourses(studentObj)
    - determines what courses the student can take to satisfy the remaining requirements
    - creates new “suggestions” field which contains an array of courses
    - returns updated student object

generateHTML.js

- uses a package called “swig” to generate html pages according to a formula
- Modules: swig
- generateStudentPage(studentInfo, res)
    - creates an html page for the given student using swig
    - sends the html page to the client in server response
- generateProfessorPage(obj, res)
    - creates an html page for the given professor using swig
    - sends the html page to the client in server response

script_index.js

- handles the toggle of the drop down menu to select identity as student or advisor
    - displays drop down when button is clicked
    - closes drop down menu when user clicks outside of it
- handles event when user clicks on one of the identities in the drop down menu with studentOrAdvisorLogin(identity)
    - reformats the page to ask user for ID with a text input and submit button
- handles event when user submits an ID with init()
    - sends a request in the browser to the server with the pathname “identity:userID.html”

script_advisor.js

- handles the event of user clicking on a student button with getStudentPage(name)
    - sends a request in the browser to the server with the pathname "identity:name.html", where the identity is "student" and the name is the name of the student that was clicked on.
- Handles the event of user clicking on the dropdown menu to filter advisees with filter(key)
    - Sends an AJAX request to the server with query string containing: request = filter, identity = advisor, the userID, and the key, which is a single-letter string corresponding to the type of filter that is requested.
    - Handles the server response to the request with filterAJAXHandler()
        - Receives the filtered result as a stringified JSON list of students
        - Parses the string into a JSON object
        - Removes the currently displayed list of students on the HTML page
        - Displays the list of students in filtered order on the HTML page