

Heart of Gold



"StudyBuddy"

Technical Design Specification

Database Modules/Components:

package.json

This Json file will contain the necessary information about our application, as well as keep track of our dependencies:

- body-parser
- connect-flash
- cookie-parser
- debug
- ejs
- express
- express-session
- morgan
- serve-favicon
- mysql
- path

create1.sql

This mySQL database file will create a user called 'studybuddy'@'localhost' with the password 'password'. It will create a database called main with a table for users which includes:

- username (uid/primary key)
- password (gets filtered out in routes before passing user object to client)
- name (Real first name)
- age (an integer)
- school (user's college, used for matching)
- courses (list of courses the user has selected on account creation or edit)
- study_pref (either quiet or talkative)
- online (boolean)
- matches_array (people who go to the same school and have ≥ 1 common course)

It also creates a table for chats which includes:

- sender (a column tracking who sent what message)

- receiver (a column tracking who received what message)
- time_sent (datetime type, tracks the time when the messages were sent)
- message (the actual content of the message as a varchar)

drop1.sql

This mySQL database file will

users.js

This javascript file will export the following fuctions:

- get
 - Returns a specific user from the database (self - based off of unique auth id)
- put
 - Allow for existing database entry to be updated
 - Update ripples user changes through to all dependencies in the database (matches are updated)
- add
 - Allow for non pre-existing user to be entered into the database
 - Update for all pre-existing user matches is performed
- query
 - Returns all users from the database matching a set of key-value pairs representing the users' fields
- delete
 - Deletes a user from the db

Brenton Chasse and Grayson Kempster, 04/10/15, Tyler Caldwell 5/2/15 & 5/7/15

Information Transfer/Data Translation Components:

index.js

A router which will provide GET methods for

- for login/splash screen,
- for profile creation

as well as POST methods for

- authentication
- processing information for user creation, from the profBuilder form

users.js

A router which will (after users have signed in) provide GET methods for

- the home/main user view,
- logout,
- user editing/deletion, (RESTful)
- matching, (RESTful)
- and chat. (RESTful)

Tyler Caldwell and Main Khan, 4/10/15

Frontend/UI Components:

style.css

This will be a universal stylesheet for the website that will create a universal look-and-feel

This is a base level component and has no dependencies

nav.css

This will control the styling of the navbar

This is a base level component and has no dependencies

Jscript.js and script.js

These files will have scripting that is used for a wide range of functionality on the website, and there may be other .js files that are made for the login and registration screens.

Use of the dojo toolkit will provide a collective of client-side functions enabling (but not limited to)

- manipulation of the DOM tree
 - (Dojo's Core Library)
- data to be brought in and out
 - (Dojo's Core Library)
- creation of UI templates
 - can be viewed as a layout engine for creating dynamic and interactive layouts
 - (Dojo's Dijit Library)

This multi-purposed .js file will include support for:

- controlling a unifying theme across all of user's application pages:
 - index.html, main.ejs
- updating application titles through DOM traversal (using dojo/dom-attr, dojo/html)
- updating application headers through DOM traversal (using dojo/dom-attr, dojo/html)
- updating application footers through DOM traversal (using dojo/dom-attr, dojo/html)
- updating application body through DOM traversal (using dojo/dom-attr, dojo/html)

This is a base-level component and has no dependencies

One of the most central uses of this javascript file will involve its ability to inject new/updated content into the user's current .html page client-side.

login.ejs

This ejb document will be the first thing a user sees when they visit the application, and it will have a login option, with an included integration for password recovery and creating a new login registration if it necessary.

This module will require:

style.css script.js/Jscript.js backend management for verifying user membership Data translation and parsing for the small form

adduser.ejs

This document will be the the screen a user uses to create a profile, and will contain a form with a post method which allows a new user's information to be added to the mySQL database.

This module will require:

style.css script.js/Jscript.js backend management for verifying user input information Data translation and parsing for the small form

profBuilder.html (deprecated due to REST)

This document will basically be the long-form version of being able to build a profile for a user. The data will be submitted to the database by form after the user completes the document.

This module will require:

style.css script.js/Jscript.js backend management for adding a new user/checking if they exist Data translation of this information, ajax protocols most likely

main.ejs

This document will have a List of new matches, old matches, classes that are being taken etc.

This module will require:

style.css script.js/Jscript.js backend protocol/Data translation for the information on the page

matcher.html (deprecated due to ejs middleware)

This will be the actual "matching" portion of the application that will present matches based on attributes the user has added in their profile.

This module will require:

style.css script.js/Jscript.js backend protocol/Data Translation for the information on the page

chat.html (deprecated due to ejs middleware)

This will be a dynamic view that will allow someone to chat with their matches and set up meetings to study with them. It will allow you to choose a match, which will open a chat window. The user will only be able to chat with one other user at a time.

This module will require:

style.css script.js/Jscript.js backend protocol/Data translation for the information on the page

edituser.html (deprecated due to ejs middleware)

Since the idea is to be able to add multiple classes to study for, there will be a document that allows a user to add another class to their profile. The user will also be able to edit their profile from this view.

This module will require:

style.css script.js/Jscript.js backend protocol/Data translation for the information on the page

Philron Hozier, 4/10/15

Gary White, 4/10/15