System Design

Table of Contents

CRC CARD

User	
- ID - Username (Display Name) - Last Name - First Name - Email - Description - UTorId - Profile Pic - Register as a user - Log in - Update personal information - Upload profile picture - Search Classroom by course code, semester, year - Join a Classroom - Create Classroom for a course - Set as a manager of a Classroom - Set as a teacher of a Classroom - (Default role – student) - Check chatting histories - Chat in chatroom - Send messages anonymously - Post questions - Answer questions - Adopt an answer to resolve posts - View all questions posted in joined classroom - View all files uploaded in joined classroom - Upload a file - View profile of classmates - Send private messages to friends - Send friend request to other users - Have a list of friends - Search for posts by keywords	AccountService ClassService Classroom Post File Role

- Search for posts by tags - File complaints on inappropriate posts - Cannot hack in using CSS Injection	

Role	
- ID - Role Name - Description - If this role can send message - If this role can post - If this role can upload file - If this role can delete own message - If this role can delete own post - If this role can delete own uploaded file - If this role can delete other's message - If this role can delete other's post - If this role can delete other's file - If this role can delete a classroom	

Classroom	
- ID - Course Code - Semester - Year - Title - Description - All [enrolled] Users and their roles - Keep track of historic chatting messages - Keep track of list of posts - Keep track of all files uploaded - Notify all users about alerts/important posts	User Post File ClassController

- Keep track of posts that are alerts/important posts - Collect tags from collection of posts	

Post	
- Post ID - Post Title - Post Content - Date posted - User association of post - Classroom association of post - If it’s a follow-up post or not and if yes, the parent post id - Save post in database - Resolved post with utilization and adoption of an answer	User Classroom ClassController

File	
- File Name - DateTime uploaded - Location of the file - Uploader - Classroom it belongs to	User Classroom ClassController

IndexController	
- Link to Login - Link to SignUp - Link to Logout if logged in - Search bar of Classrooms	User SignUpController LoginController ClassService

SignUpController	
- Obtain username - Obtain first name - Obtain last name	User AccountService

- Obtain password - Obtain UTORid - Use AccountService to register a new user with info obtained	

LoginController	
- Obtain username - Obtain password - Authenticate within databases using username and password - Redirect user to profile page after logged in	User AccountService

ProfileController	
- Upload and set profile picture - Set first name and last name - Change first name and last name - Upload resume - Set major - Change password - Communicate with AccountService to update DB with changes	User AccountService

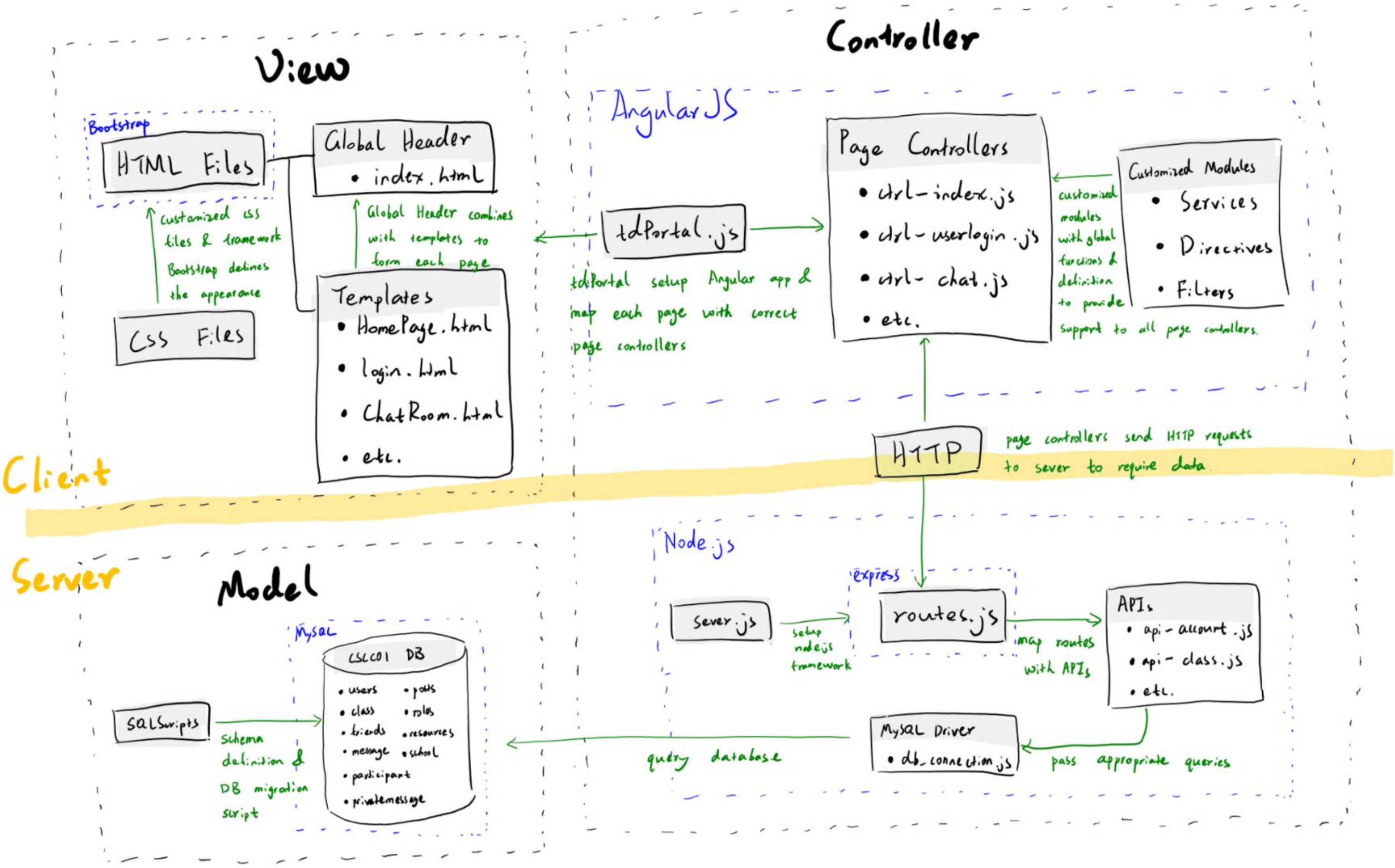
ClassController	
- Get posts created by user and use ClassService to save to db - Get files uploaded by user and use ClassService to save to db - Get messages sent by user and use ClassSerivce to save to db	User Post File ClassService

AccountService	
- Check if user is logged in - Logout user	User

- Authenticate - Sign up a user - Retrieve user info from db - Save user profile pic into db - Save user info into db	

ClassService	
- Retrieve a class info from db - Record in db that a user joins a classroom - Create a classroom - Delete a classroom - Update a user's role in the classroom - Send out a chat message and record in db - Pull and update chat messages in real time	

Architecture Diagram


View
Bootstrap
HTML Files
Global Header
• index.html
customized css files & framework Bootstrap defines the appearance
Global Header combines with templates to form each page
CSS Files
Templates
• HomePage.html
• login.html
• ChatRoom.html
• etc.
Controller
AngularJS
tdPortal.js
tdPortal setup Angular app & map each page with correct page controllers
Page Controllers
• ctrl-index.js
• ctrl-userlogin.js
• ctrl-chat.js
• etc.
Customized Modules
• Services
• Directives
• Filters
customized modules with global functions & definition to provide support to all page controllers.
HTTP
page controllers send HTTP requests to sever to require data
Client
Server
Model
MySQL
CSCC01 DB
• users
• posts
• class
• roles
• friends
• resources
• message
• school
• participant
• privatemessage
SQLScripts
Schema definition & DB migration script
Node.js
express
sever.js
setup nodejs framework
routes.js
map routes with APIs
APIs
• api-account.js
• api-class.js
• etc.
MySQL Driver
• db_connection.js
pass appropriate queries
query database