# System Design

Table of Contents

[System Design 1](#_Toc486883937)

[CRC CARD 3](#_Toc486883938)

[Architecture Diagram 8](#_Toc486883939)

## 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 * Search for posts by tags * File complaints on inappropriate posts * Cannot hack in using CSS Injection | AccountService  ClassService  Classroom  Post  File  Role |

|  |  |
| --- | --- |
| ***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 * Keep track of posts that are alerts/important posts * Collect tags from collection of posts | User  Post  File  ClassController |

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

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

|  |  |
| --- | --- |
| ***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