|  |
| --- |
| agile phases: Start Date => (27-4-2023) |
| - CONCEPTION AND REQUIERMENTS  - DESIGN  . UML Design  -Data Structure Design  Error Handling Design  Performance Design  Security Design |

|  |
| --- |
| Function Requirments |
| 1. comments page to manage it. => validate the ressolver + Exceptions  * get one comment with id * get all comments with pagination * get the latest comments and the count them * get all comments within interval of time * get the reported comments =====🡺 need to add the feature of reporting on comments * get all comments on that blog ordered by the most comment has replies * get all the comments for that user * add comment * remove comment * edit comment * get the count of comments of blog  1. Reply page => validate the ressolver  * Get one reply with id * Add * Update * Delete * Exceptions  1. Blogs page // all queries and mutations that uses Blog model  * one blog with id * get the blogs related to one writer * get all blogs related to that category * get the invisible blogs for that user who disappear them * display the writer blogs that need confirm * scheduling The Posts //event * get all blogs with pagination * get the most popular blogs with comments and likes // how to paginate the result? * get the blogs in interval of time * add blog * update blog * soft delete blog => when delet the blog the   media deleted too  blogtags  comments  reply  read by  reports  likes  share   * search for blog in title, body, description and with time created in it * delete blog after 10 reports from users //event * undelete the blog * invisible the blog * Visible the blog  1. media page  * display all media for that blog. * if I need to add vedio from youtupe < optional > * add * upload images * update * delete  1. tags page   • display all tags   * display one tag with id * add * update * delete and cascading with Blog Tags * restore * force delete did not deleted  1. Blogtags page  * display all blogs in that tag with pagination * display all Tags in that Blog with pagination * Add * Update * Delete (force)  1. User page  * display the users * search for user with name * the event is the blocking and have several listeners: 1) notify the writer 2) if it the tenth Be User * Scheduling // block for 1 month if it 10, and delet the blocks to recount again * Ask for be writter * Sign in * Log out * Register new user * Update user data * Delete user softly | * Restore | => the deletion of the user will delete everything related to him. * Force Delete | * Authentication  1. Admins page  * Get all * Get with name * Get with id * Search for admin * Get current admin => needs remember tokens tables like user * Login * Logout => needs remember tokens tables like user * Register new admin => email\_verified\_at column * Upload admin photo after registration * Disactivate admin // cannot log in * Activates admin * Last log in time ⌚ * Who is active (log in) now? * Update admin * Delete admin // who is gona do this?  1. Category page  * Display all category * Get one category * Search for category with name * Add category * updates * delete category softly * cascading the deletion * restore category * Force Delete category  1. News page  * Get all * Get one * Search * Add, show, update and delete news * appear or disappear news * scheduling The news  1. Blocks page  * Get number of blocks * Add block => this not a custom resolver so how to add this rule CheckUserIfIsWriter * Unblock  1. Follow page 2. Reade by 3. Reports page  * get the reported blogs * count the # of reports on that blog  1. Likes page 2. Shares page 3. Comments page  * block user from commenting // method that contain some words if it found in his comment  1. Chates page  * user can chat with admin with messaging system  1. Messages page   BUGS:   * + - 1. Cascading on delet, restore and force delete Blog and its Relations       2. Cascading on delet, restore and force delete Category and its Relations       3. Cascading on delet, restore and force delete Tags and its Relations       4. Cascading on delet, restore and force delete User and its Relations   systems need:   1. notification 🔔  * for writers if some one  1. Comment 2. Replay on his comment 3. liked 4. shared 5. reported 6. Blocked 7. read 8. followed him  * for User if some one  1. Replay on his comment 2. His followed writter write new blog  * for Admin if  1. user reported blog 2. user blocked writter 3. Emails 4. Caching 5. Chatting |
| In the next version   1. Contributed Blogs => some blogs will be able to modified by some users in another branch as open source blog. |