MCO Documentation
‘Budol Finds’

CCAPDEV S15/S16

Group 10

Escalona, Jose Miguel A.

Batista, Ralf Taruc

Fontanilla, Carlos Emilio

Romero, Ayiana Louse Garcia

04/06/2022

Introduction

Our proposed web application is for shoppers to share their bought items to recommend to fellow shoppers. The idea was taken from the private Facebook group, Home Buddies, with over 3 million members. The Facebook group's posts cover different topics and purposes, however, the group decided to take on the idea of sharing "budol finds", which are items that were bought and enjoyed by home shoppers. The web application's main aim is to provide a platform of consolidated "budol finds" for shoppers to make sharing and finding items easier, rather than a general Facebook group page. Through the web application, shoppers are given a better shopping experience as items they wish to buy are items suggested by fellow shoppers. It also makes it easier to shop as items are organized by categories and provided with a link or destination on where it can be bought.

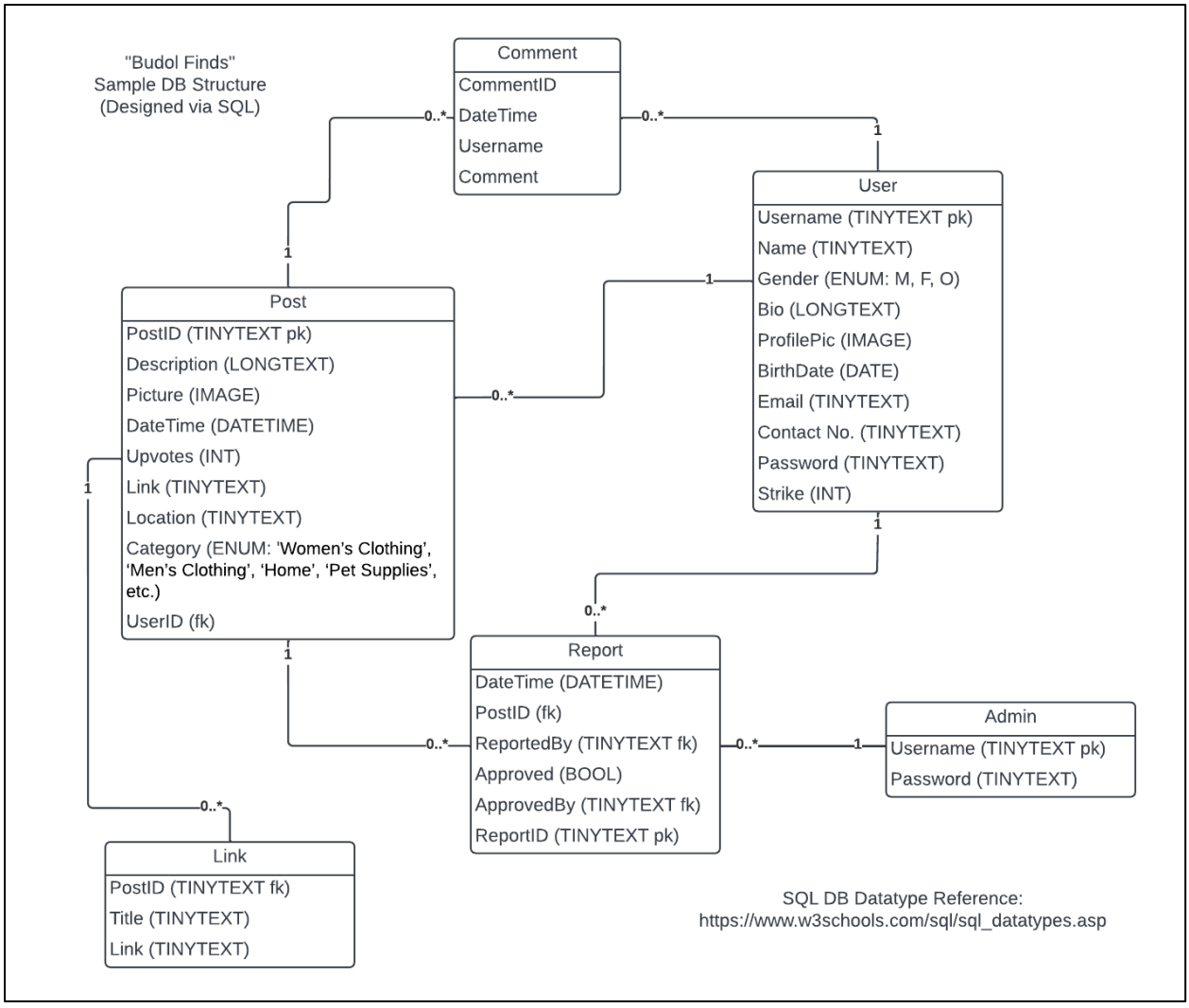
The web application would require a user to create an account before they are able to share their finds. The user can then create a post containing what they bought, where to find it, and a media attachment of the item. The website would have a user page, home page/timeline, and different pages organized by item category (e.g.: 'Women's Clothing', 'Men's Clothing', 'Home', 'Pet Supplies', etc.).

Description of the Web Application (Features)

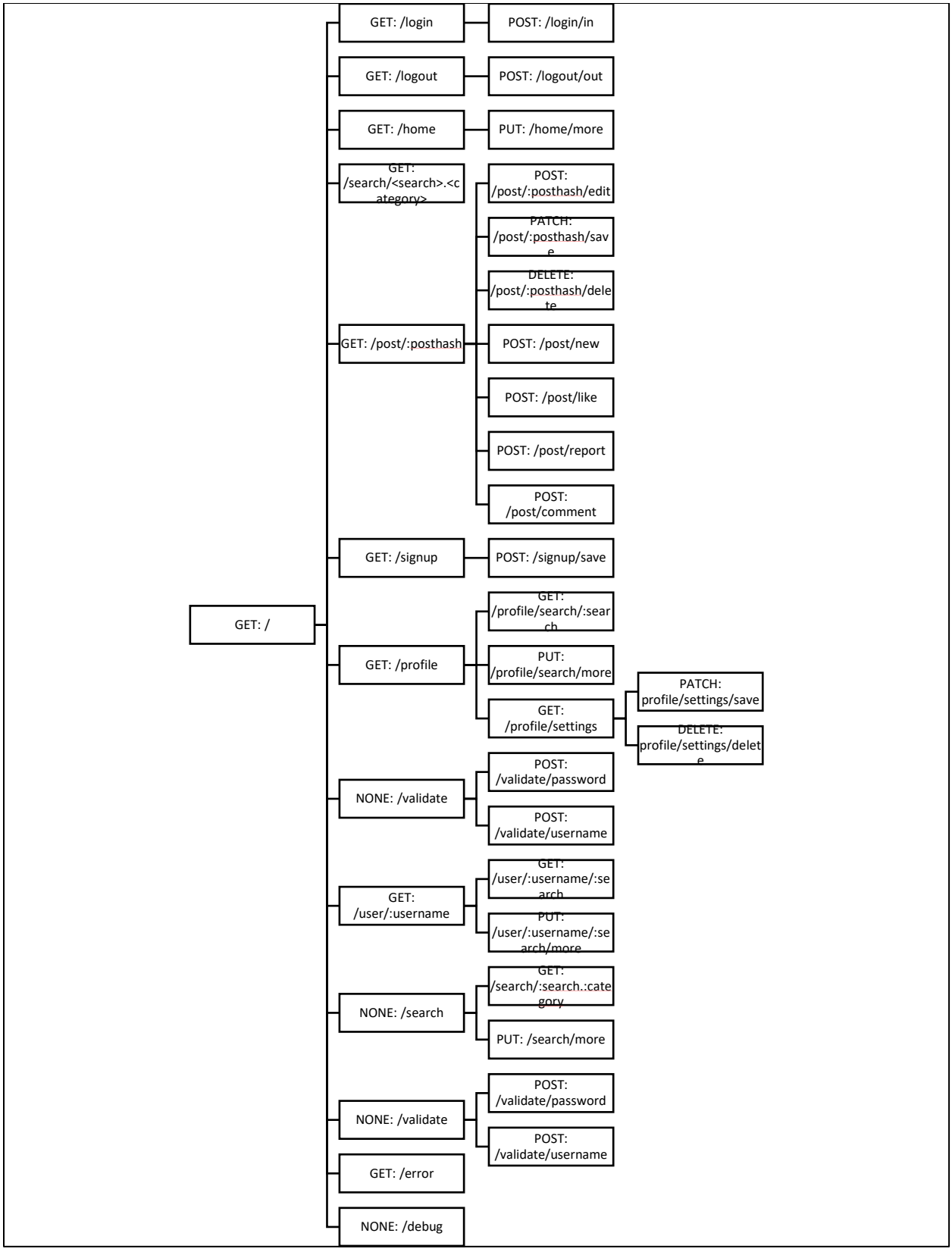
1. User Account
    a. Users must be able to log-in and log-out according to their discretion
2. User Profile
    a. Users must be able to create an account by providing basic information about themselves such as:
        i. Username
        ii. Email
        iii. Name
        iv. Gender
        v. Birthdate
        vi. Bio
    b. Users will also have a three-strike policy to ensure no unwanted acts can be done on the platform. If the account hits three strikes, the account and its associated email will be flagged and banned. (Downgraded: The number of reports received are displayed on the user's page as viewed instead.)

    c. Users' respective posts can be found on their profile
    d. Users can also view other user's accounts.
3. Post
    a. Users can create posts that should contain
        i. Description
        ii. Photograph
        iii. Link or location where they bought it
        iv. Category
    b. Users can update or delete their posts
    c. Other users can view post individually or as part of the user's gallery, or home timeline.
    d. Users can search for posts based on their specified keyword(s)
    e. Users can upvote and comment on a post
2. Comment
    a. Users can create comment on posts
    b. Users can update or delete their comments (Deprecated)
    c. Users can view other user's comments on a post (Downgraded: Limited to latest 5 comments per posts)
3. Report
    a. Users can report posts that may violate the web app's TOS. (Downgraded: No specific definition/cause can be provided but a direct report count will be added to the user.)
    b. Each validated report is equivalent to one strike on the user. (Downgraded: Direct report counting is implemented instead).
4. Admin (Deprecated)
    a. Has a similar properties as a user, specifically it having a username and password.
    b. Admin username & password can be updated as needed.
    c. Admin accounts can be deleted by the administrator themselves.
    d. Admins can set approval of reports.

Initial Database Design Diagram (Sampled in RDB style)

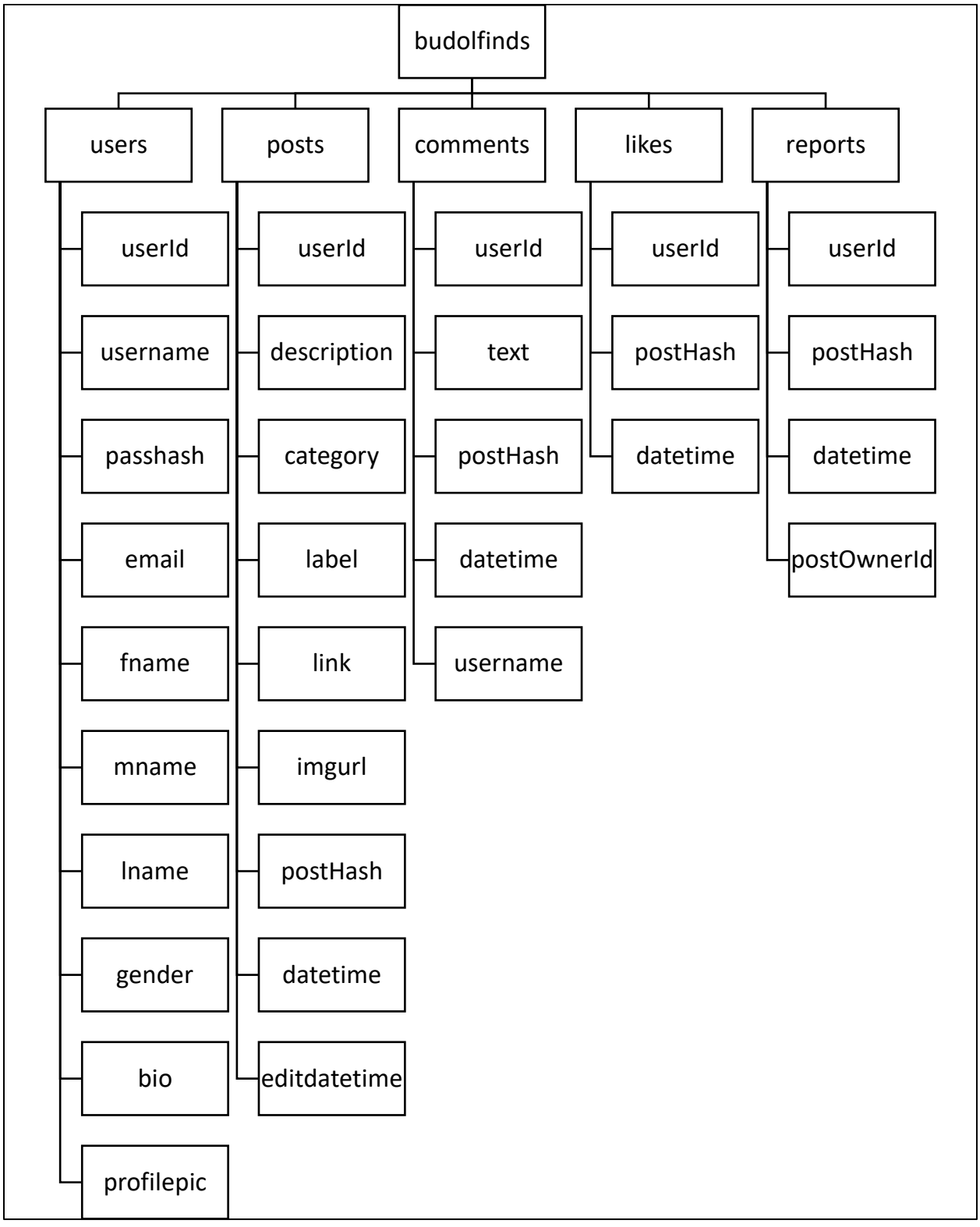


Page Structure – How the web application routes be structured?



Data Model – How the data in the DB is structured?

The database selected for the project is MongoDB which is a No-SQL or Non-Relational Database. Listed below is the structure of the contents of the database.



Technologies Used – What technologies are used to build the web application?

- Frontend
  - Vanilla JS
  - JQuery (via Cloudflare CDN)
  - Express-Handlebars
- Backend
  - NodeJS (v16.15.0)
  - Express
  - MongoDB (v5.0.3)
  - NPM (v8.11.0)
- Dependencies (Extracted from package.json)
  - bcrypt
  - cookier-parser
  - crypto-random-string
  - dotenv
  - express
  - express-handlebars
  - express-session
  - hashids
  - https-status-codes
  - mongodb
  - multer
  - nocache
  - path
  - server-favicon
  - url
- External Dependencies
  - validator.js (via unpkg.com)

Vulnerabilities Determined

- A vulnerability was determined on the underlying dependencies used by busboy and multer called dicer as of 4th of June 2022.
- GitHub released an advice regarding the matter stating that it affects the dicer package versions 0.3.1 and below with no patches being released as of 4th of June 2022.
- The vulnerability was determined to be of High Severity that could cause a DOS when a malicious form is sent in a loop.
- Link: https://github.com/advisories/GHSA-wm7h-9275-46v2