**PROJECT TWATTER**

**Software Design Document (SDD)**

Group 4

Ahror Abdulhamidov

San Jose State University

October 14th, 2020

**Table of Contents**

**Title Page** 1

**Table of Contents**  2

**1. Introduction** 3

1.1 Purpose 3

1.2 Scope 3 1.3 Overview 3

1.4 Reference Material 3

1.5 Glossary 3

**2. System Overview**  4

**3. System Architecture** 5

3.1 Architectural Design 5

3.2 Decomposition Description 6

3.3 Design Rationale 7

**4. Data Design** 7

**5. User Interface Design** 8

5.1 Overview of UI 8

5.2 Screen Images 8

5.3 Screen Objects and Actions 10

**6. Requirements Matrix** 11

**1. Introduction**

**1.1 Purpose**

This software design document describes the software architecture and system design of Twatter, the web application social network.

**1.2 Scope**

Twatter is a blog posting social network web application. In general, it is a Twitter clone, a more simplistic and minimalistic version of it. This project will benefit the public, helping them to spread the word across the world in this online platform, which runs on any modern browser from any place in the world.

**1.3 Overview**

This document will provide a detailed description of the architectural design of the software, data design and management, description of the key software components, design of the human interface (UI/UX), and the requirements matrix.

**1.4 Reference Material**

The software, described in this document, uses Laravel version 6., which is a PHP based framework, and therefore, some of the features / components / tools / key words are related to Laravel and PHP. For more information, check this website: <https://laravel.com/docs/8.x/>

**1.5 Glossary**

Front-end – the practice of converting data to a graphical interface

Back-end – server side of an application and everything that communicates between the database and the browser

HTML – Hyper Text Markup Language

CSS – Cascade Style Sheet

PHP – Hypertext Preprocessor, general purpose scripting language, used primarily for web-development

Laravel – popular open source PHP based web-development framework

**2. System Overview**

This software project “Twatter” is a short blog posting web application intended for public use. In general, it is a Twitter clone, a more simplistic and minimalistic version of it. It allows users to register / sign in to the system, post messages or short blogs online that can be viewed by others. In addition, users can follow other users, and therefore, have their posts on their personal feed. Users will also be able to like other people’s posts.

The application server side can be run on Windows / Linux / Mac OS computer, while for the users, it will be accessible through any smart device with modern browser and stable internet connection.

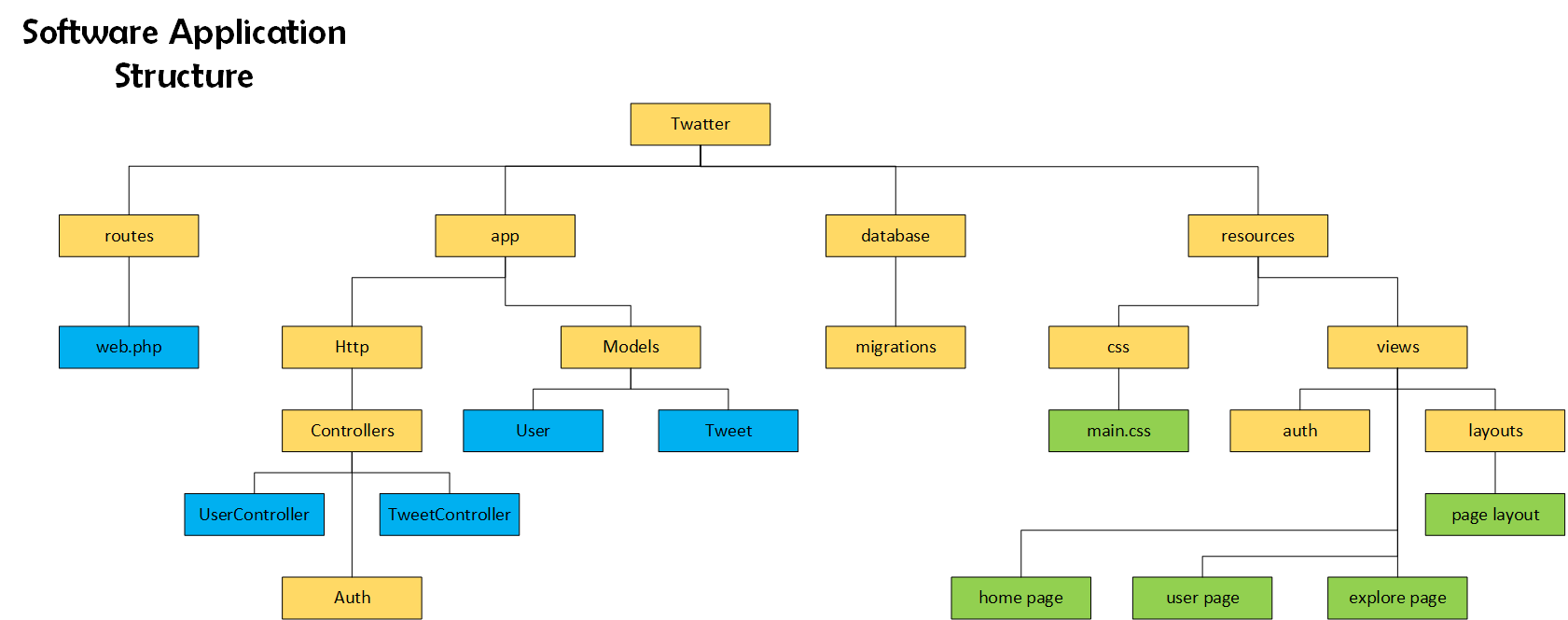
The web application server will run on a Dell Precision workstation laptop connected to a stable internet connection.

**3. System Architecture**

**3.1 Architectural Design**

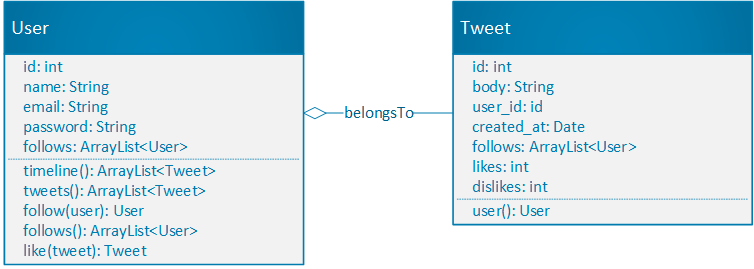
Architecture of the software is based on Laravel, PHP-based framework for creating MVC web applications. The models in our applications will be User, model that describes users, and Tweet, model that describes tweets (posts). The views (browser pages) will be controlled through controllers that directly access models. Controller access and distribution is done through the routes file (web.php). Models will be accessed through and saved in database tables, which will be created through migrations files using php artisan, a command tool provided by the Laravel framework. In addition, the authentication API will also be provided and integrated through Laravel during the creation of the application.

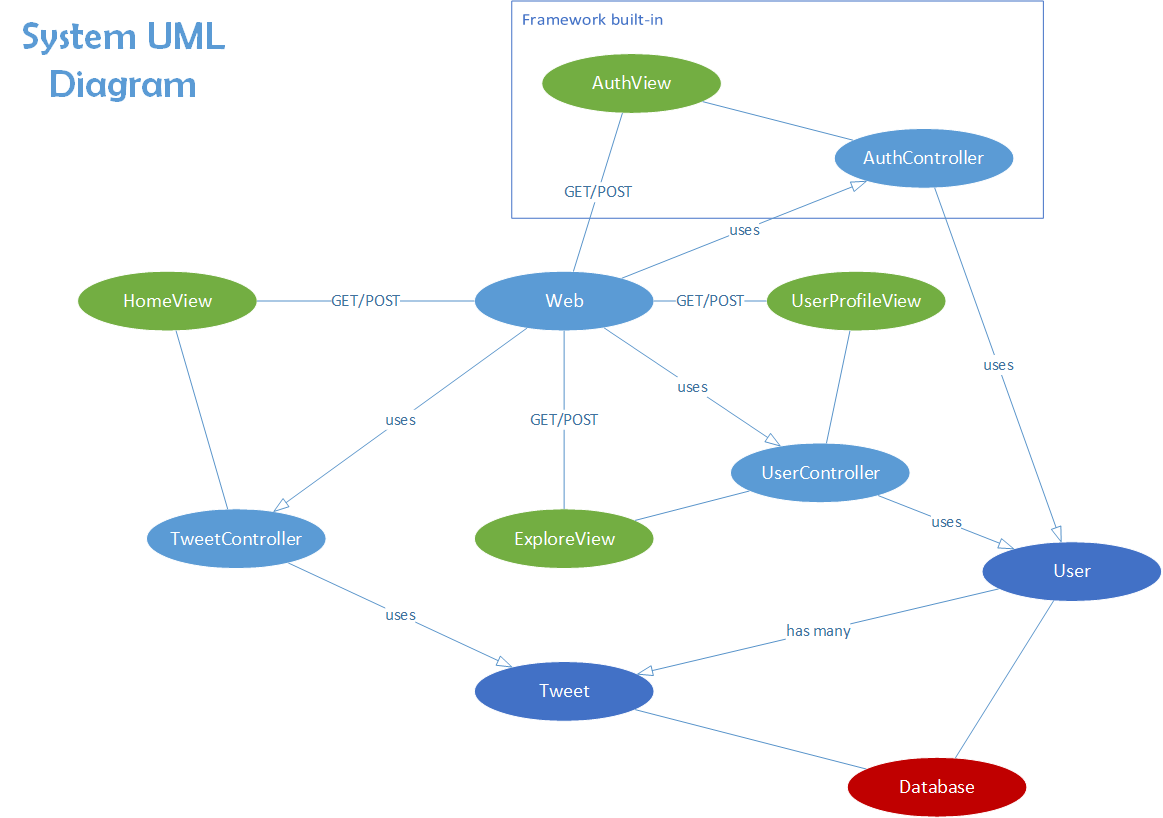
Here is a brief structure of the software file/package structure:



**3.2 Decomposition Description**

Below are the object diagram of the models and an overall system components UML diagram:





**3.3 Design Rationale**

Model-View-Controller software architecture pattern is the best possible pattern for this type of applications, where user interacts with interface, by viewing, creating, or updating different objects. Essentially, it will be our backend software, specifically, models and controllers that will manage the website.

In addition, the architecture is standard for the projects built using Laravel framework. While Laravel is simpler to use PHP framework compared to others, it is also very suitable for creating php web MVC applications that are of medium or smaller scale; large scale social network applications, like Facebook, should not be written in Laravel. Overall, it is not the most efficient or secure, but it is also not worst, and it is not hard to integrate, which makes it the best fit for this project.

**4. Data Design**

As it was said previously, Twatter is an MVC pattern web application, and therefore, the core of the application would be its models, User and Tweet.

Model User describes a user who created his account through authentication. It contains id of the user, name, email, and password. In addition, user have his own many tweets and followings (other users whom the user follows). User can follow another user, return his tweets, return his timeline (tweets of the user and of his followings), and return his followings (users whom he follows).

Model Tweet describes a tweet (post) that was created by a user. It contains the id of the tweet, its title, text body, date of creation, and id of the user to whom it belongs. User and Tweet are connected with a “has many” relationship. Tweet can return the user it belongs to.

Both models will have database tables (users\_table and tweets\_table), which will be created or changed through migration files, which is an easier way of database management, provided by the framework. The type of database for this project will be MySQL.

**6. Human Interface Design**

**6.1 Overview of UI**

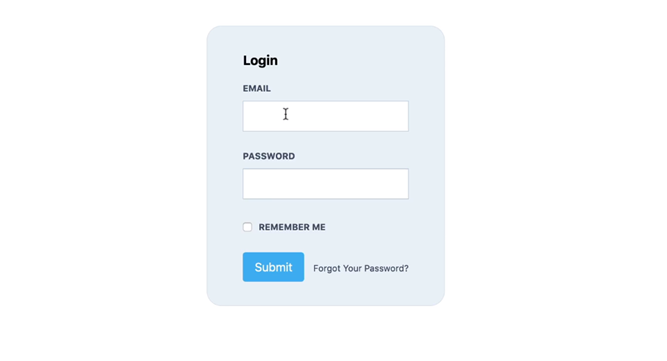
The UI of this application will consist mainly of 4 parts/pages: authentication page, home page, profile page, and explore page.

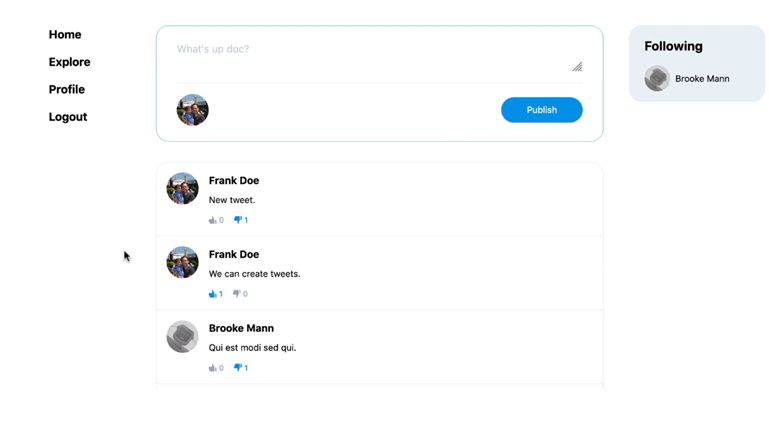
Authentication page is for the user to register or login to his account. Home page is for displaying the user’s timeline, which is a list of his and his followed users’ recent tweets. In addition, the home page will consist of the list of the user’s followings and a navigation bar to navigate to other pages. The user / profile page is a page that describes a certain person and shows that person’s tweets. Finally, the explore page is a page where user can explore other users.

Home, user, and explore page will have similar design, and therefore, will use the same css stylesheets and layouts.

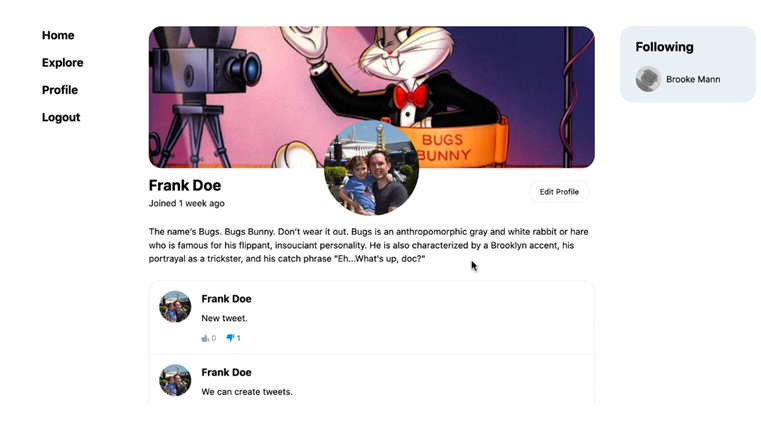
**6.2 Screen Images**

1) Authentication page:

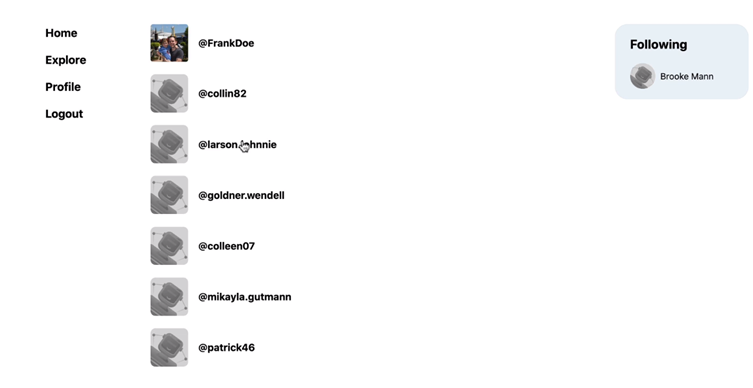


2) Home page (Twatter feed) 

3) Profile page



4) Explore page



**6.3 Screen Objects and Actions**

The authentication page has two states – register and login. Register page will display username, password, and name fields that the user can fill. Then, user can click the “register” button, which will create and store the new user and redirect the user to the home page. Login page will have only two fields, username and password, and the “login” button that will log in in the user, if the credentials are correct, and redirect to the home page.

The home page is for displaying the user’s timeline, which is a list of his and his followed users’ recent tweets. Each tweet in the timeline will have a text body, clickable name of the user who posted it that redirects to that user’s page, and user’s profile picture. Also, there will be a clickable like/dislike counter. On top of the timeline will be a form for creating and posting a new tweet on behalf of the user who will be logged in. In addition, the home page will consist of the list of the user’s followings and a navigation bar to navigate to other pages. Each link in the navigation panel will be clickable and will redirect the user to that page. On the right side, there will be a panel with clickable names of the users whom our user follows.

The user page will consist of the user description: user’s name, profile picture, short bio, and user’s tweets. If the user page belongs to the user then the user will be able to click on the “edit profile” button and edit his info, including picture. If the user page is for another user, then there will be a button “follow” that our user will be able to click and become that user’s follower. In addition, there will be navigation bars on the sides, same ones as in the home page.

The explore page has a clickable list of all users. It will also have the same navigation bars as the home page, and might even have a search option for the users later on.

**7. Requirements Matrix**

2.1.1 – Auth API, provided by Laravel

2.1.2 – Auth API, provided by Laravel

2.1.3 – Auth API, provided by Laravel

2.1.4 – user page, user controller

2.1.5 – explore page, user controller

2.1.6 – other user’s page, user controller

2.1.7 – right side navigation panel (home page), user controller

2.1.8 – timeline (home page), tweet controller

2.1.9 – timeline (home page), tweet controller

2.1.10 – any tweet on timeline, tweet controller

2.2.1 – web routes file, controllers, views

2.2.2 – tweet controller, home page, tweet model

2.2.3 – tweet controller, home/user pages, tweet model

2.2.4 – user controller, navigation panel, user page, user model

2.2.5 – user controller, explore page, user model

2.2.6 – user and tweet models, database migrations and entries, Laravel provided database control API