

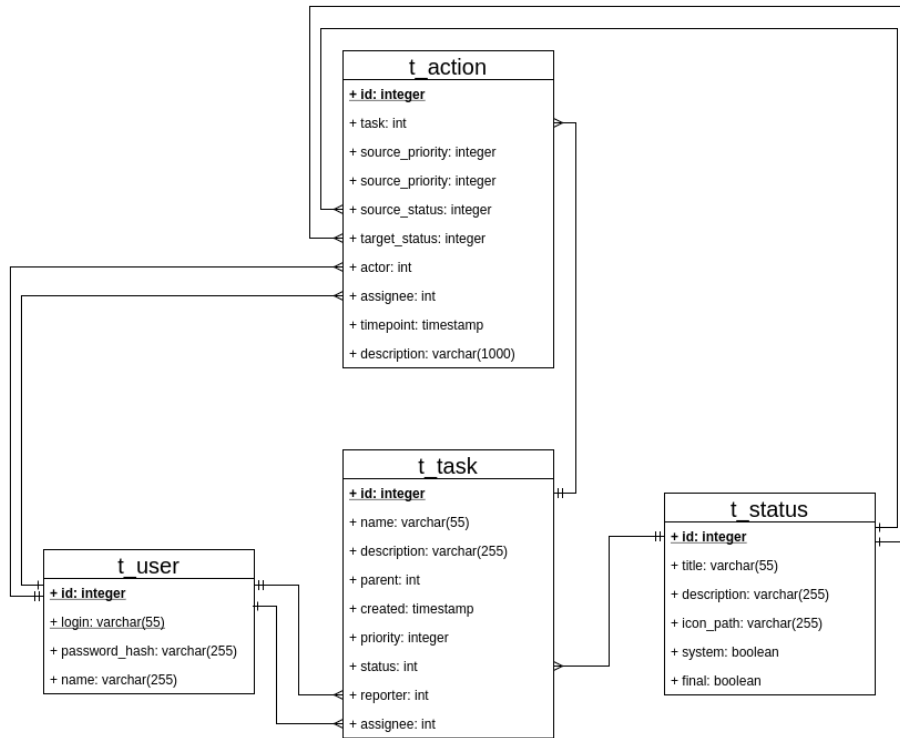
# Temporal Issue Tracking System

## User Manual

## Contents

<b>1</b>	<b>Logical model</b>	<b>1</b>
<b>2</b>	<b>Database</b>	<b>1</b>
<b>3</b>	<b>PHP</b>	<b>1</b>
<b>4</b>	<b>CSS</b>	<b>2</b>
<b>5</b>	<b>JavaScript</b>	<b>2</b>

## 1 Logical model



## 2 Database

Application utilizes MySQL database management system. All the database queries are executed as prepared statements in order to provide security. Credentials are stored in the form of username and password hash pair.

## 3 PHP

PHP code is split into several files. Header is implemented in `header.php`, database connection utilities are in `db.php`, rest of the files implement different pages of the application and are name accordingly.

Header is injected into each page and contains context actions specific for each page.

Passwords are hashed using the salt, combined from the password itself and additional characters, see `signup.php` or `login.php`.

## 4 CSS

CSS styles are stored in separate files, **style.css** being common for the entire application one and **header.css** is common for the part of the application, an authorized user can see. **footer.css** is a common footer for the list pages. The rest of the files correspond each to its own page and are named accordingly.

Pseudoelements are used to style placeholders.

Most elements' dimensions are tied to a font size so that application can be easily scaled.

## 5 JavaScript

JavaScript is utilized for pre-validation of the form fields, but application is fully functional without it.

Script files are separated from the main code, each script corresponds to a page it's named after.