Scrap Ticketsystem

Konzept:

Das Ticketsystem ist für Softwareentwickler konzipiert und soll Ihnen das Arbeiten in kleinen Teams erleichtern. Es soll als eine Art Verwaltungstool für Softwareprojekte dienen. Es soll zum einfacheren Koordinieren des aktuellen Projektstands und von noch zu erledigenden Aufgaben hergenommen werden.

Ist Zustand:

Aufgaben werden mündlich verteilt, hier muss man dann selbst entscheiden ob die neue Aufgabe eine höhere Priorität als die alte Aufgabe hat. Es gibt keine Möglichkeit herauszufinden wie weit eine Aufgabe eines Kollegen ist. Zudem wissen die Kollegen nicht an was bereits gearbeitet wird.

Soll Zustand:

Kern der Software soll das Verwalten von Tickets sein, welche die aktuellen Arbeiten zusammen mit einem Status, einer Priorität für den Bearbeiter und einer Kategorie versehen. Diese Verknüpften Datensätze sollen jederzeit erweitert werden können um im entsprechenden Softwareprojekt beispielsweise eine neue Kategorie wie "Deployment" hinzufügen zu können.

Der Zugang zum Ticketsystem soll pro Projekt verwaltet werden. Neue User können sich für das Ticketsystem registrieren und dann Anmelden. Als Standard sollen Sie nach dem Login nur Informationen erhalten, die Ihnen explizit zur Verfügung gestellt wurde. Eine Benutzergruppe mit erweiterten Rechten muss pro User explizit von einem Admin vergeben werden.

Der Bearbeiter eines Tickets soll jederzeit von einem Admin verändert werden können. Das bei aus fall eines Bearbeiters ein Ticket nicht bis zur Rückkehr liegen bleibt. So gibt es auch die Möglichkeit einen Kommentar in einem Ticket zu erstellen diese helfen zum einen einem Selbst um nach einer Pause das Ticket schnellst möglich wiederaufnehmen zu können und zum anderen helfen die Kommentare den Kollegen falls man für eine längere Zeit ausfallen sollte.

Es sollen Benutzergruppen definiert und implementiert werden um die Rechte verschiedener Benutzer zu verwalten. Im Standard vorgesehene Benutzergruppen sind "Normal", "Redakteur" und "Admin".

Ein Entwickler mit der Benutzergruppe "Redakteur" soll nur Tickets sehen, die Ihm zugewiesen sind.

Ein Admin sieht alle Tickets des Projekts, soll aber auch eine eigene Ansicht für Tickets haben, die nur Ihm zugewiesen sind. In einem Projekt sollen User mit der Benutzergruppe Admin auch als Ansprechpartner für Fragen anderer Entwickler zum Projekt agieren und können im Rahmen eines Projekts die Rolle des Projektmanagers übernehmen.

Tickets die erstellt, aber noch keinem Bearbeiter zugewiesen wurden, sollen in der Ansicht eines Admins einzeln hervorgehoben erscheinen.

In einem Kanban Board sollen alle Tickets des aktuellen Projekts angezeigt werden. Die Übersicht im Kanban Board soll durch eine visuelle Zuordnung zum aktuellen Status eines Tickets umgesetzt werden. Im Kanban Board sollen die Zuordnungen zu einem Status per Drag-And-Drop bearbeitet werden können. Vordefinierte Status sind "Backlog", "Ready for Development", "In Progress", "Waiting for …" und "Finished".

Ein häufig auftretendes Problem bei Softwareprojekten ist die Priorisierung einzelner Arbeiten. Um den Entwicklern einen besseren Überblick im Projekt zu verschaffen soll einem Ticket immer eine Priorität zugewiesen werden. Diese Priorität kann je nach Bedarf und Dringlichkeit der Anfrage angepasst werden. Hierbei sollte vorab definiert werden welchen Dringlichkeitsgrad bestimmte Prozesse haben um eine einheitliche Priorisierung im Projekt zu verfolgen. Vordefinierte Prioritäten sind: "Blocker", "High Priority", "Normal" und "Low Priority".

Voraussichtlich verwendete Software:

Zur Entwicklung des Scrap-Ticketsystems wird das Flask-Framework verwendet, welches während der Entwicklung als Server agieren soll. Als Datenbank ist für die Entwicklung ein MariaDB Docker Container vorgesehen, mit dem unabhängig voneinander Daten zum Testen eingegeben werden können und die Entwicklung komplett lokal stattfinden kann.

Die Versionisierung im Projekt und die Arbeiten einzelner Entwickler sollen mit einem Git-Repository koordiniert werden. Entwickelt wird im Development Branch. Der Master Branch soll für das Deployment verwendet werden.

Die Ansichten für Benutzer des Scrap-Ticketsystems soll aus den Daten generiert werden die von einer REST-API übergeben werden. Das Frontend baut auf dem Bootstrap Framework auf und soll, was Design und Benutzeroberfläche betrifft, dem Standard Design von Bootstrap ähneln.

Datenbankentwurf:

Hierbei ist das Design so zu wählen, dass alle geforderten Informationen in der Datenbank Platz finden, und diese der dritten Normalform entsprechen. Hierzu zählen folgende Regeln:

* Einzelne Attribute sind nicht weiter zerlegbar.
* Nicht-Schlüssel-Attribute werden nicht allein von einem Teilschlüssel bestimmt.
* Kein Nicht-Schlüssel-Attribut ist transitiv abhängig vom Primärschlüssel.

Zudem wird bereits beim Entwurf beschlossen, welchen Datentypen die verschiedenen Felder haben sollen. Anhand der zu verarbeitenden Daten und unter Einhaltung der dritten Normalform wurde ein Datenbankschema erstellt.

Besonderheiten:

Ein Eintrag in die Tabelle Benutzer wird durch eine n:m Beziehung mit den Tabelle User\_Group verknüpft, da ein User auch in mehreren Gruppen gleichzeitig sein kann wird dies nicht direkt in den Datensatz des Benutzers geschrieben sondern in die User\_In\_Group ausgelagert.

Wichtige Bestandteile der Software sind:

* Anlegen von Tickets
* Bearbeiten von Angelegten Tickets
* Kanban Board als Übersicht
* Anlegen und Bearbeiten von Prioritäten
* Anlegen und Bearbeiten von Status
* Anlegen und Bearbeiten von Kategorien

Benutzte Software:

* PyCharm Programmierung in Python
* Visual Studio Code
* Google Chrome zur Darstellung des Projekts

Implementierung:

ER-Diagramm:

Datenbankschema: