PROJECT 02: WEB-BASED ISSUE TRACKING SYSTEM REPORT
CSC 530: Software Engineering Methodology
Henry Posada, Frankie Colasurdo, Akmal Muminov, Kyle Pagnutti

OVERVIEW

Find our GitHub repository here

Find our YouTube demonstration here

Our project utilizes HTML and CSS to create a simple web-based interface in order to portray our tracking system. Leveraging JavaScript (JS), our homepage prompts users with a sign in feature that allows existing users to log in as well as new users to sign up.

Once a user is authenticated they will have access to our system. Our web interface connects with a SQL database we're hosting, which has tables for users as well as tickets. From here, there are a number of functional options developed using Python, JS, and PHP.

FEATURES

- A user can create an account
- An authenticated / registered user can sign in
- A user can record a new incident which will have a unique identifier
- A user can view particular incidents from a list
- A user can edit the fields of an incident
- A user can update the status of an incident
- A full history of incidents is maintained
- Incidents can be queried through a number of attributes (such as a relevant keyword, its unique identifier, or its status)

Our code also contains various other methods, such as to check if a particular user exists as well as configuring users in the database.

BACKLOG

To supplement our [JIRA page](), the following generally details our workflow of tasks and how we organized our project's development using sprints and Agile methodology.

Sprint 1: November 17 - 24

- Initialize the SQL database
- Populate SQL tables with test data
- Basic front-end development started
- Plan and organize project goals

Sprint 2: November 24 - December 1

- Develop Python script with functions for querying SQL (see connectionToSql.py)
    - As an authorized user I want to query tickets by ID
    - As an authorized user I want to query tickets by a keyword
    - As an authorized user I want to query tickets by status
    - As an authorized user I want to update the status of a ticket
    - As an authorized user I want to check if a user exists
    - As an authorized user I want to add a new ticket
- Set up log in / sign up homepage (see homepage.html, admins.html)
    - As an authorized user I want to log in
    - As an unauthorized user I want to sign up
- Set up pages and code for creating, modifying, and viewing tickets (see query.html, users.html)

Sprint 3: December 1 - 8

- Create Python formatting method to clean up the ticket array (see format.py)
- Finish user functionality; including log in, sign up, etc.
- Finish ticketing functionality; viewing, creating, etc.
- Touch-up CSS for final presentation
- Polishing / fine-tuning the system and documentation for release

Screencap of our JIRA page:

Projects / Ticketing system

Backlog

🔍 [search] 🐱 AM HP 👥 | Epic ⌄

▾ **TS Sprint 3** 1 Dec – 8 Dec   0 0 0 [Complete sprint] •••
Polish final project
15 issues

🟩	TS-10	The user should be able to update or close the existing tickets	IN PROGRESS	👤
🟩	TS-6	As an authorized user I want to update the ticket	IN PROGRESS	👤
🟩	TS-7	Create a web site with ticketing functionality	IN PROGRESS	👤
🟩	TS-2	As an authorized user I want to create a new ticket	IN PROGRESS	👤
🟩	TS-5	As an authorized user I want to close the ticket	IN PROGRESS	👤
🟩	TS-11	The user should be able to add a new ticket	IN PROGRESS	👤
🟩	TS-4	As an authorized user I want to view the list of existing tickets	IN PROGRESS	👤
🟩	TS-9	As the user logs in, he should be able to see the list of existing tickets and view individual tickets	IN PROGRESS	👤
🟩	~~TS-14~~	Create Python methods for querying SQL database	DONE	🐱
🟩	~~TS-3~~	As an unauthorized user I want to login	DONE	AM
🟩	~~TS-8~~	Create a sign-up page for an unauthorized user	DONE	AM
🟩	~~TS-12~~	Create a log-in page for unauthorized user	DONE	AM
🟩	~~TS-16~~	Basic front-end developed	DONE	AM
🟩	~~TS-17~~	Basic database foundation initialized	DONE	HP
🟩	~~TS-18~~	Plan out and organize project goals	DONE	🐱

\+ Create issue   💡 Quickstart