Cheyenne Pierpont

Bailey Forbes

Brandon Maurer

RS-9

03/23/2025

Write a programmer manual:

1. Vision statement:

Our project will be designed to help Cave Country Canoes improve the management of emergency distress situations and the handling of map-based data on their operations, among the critical tasks that are currently being done by them. With the emergence of such a modern system, the management of processes will be made more efficient, integrating the following: Introduction and functions

2. Introduction:

The system will be a web-based platform made with the help of Python's Flask framework, and the back-end data will be managed through a MySQL database. Primary features of the system are trip management functionalities, including logging customer trips, recording points of interest (POI) and geofences and sending emergency alerts. Also, to ease the management of the resources and response teams, sending processing of the same will occur, thus ensuring quick encapsulation and response during times of emergency. Data security and continuous updates are improved by the automatic triggers for logging changes within the accounts, customers, turns, and emergency incidents. This layout is designing a trustworthy, responsive system from both customer and in-house side.

By using the services of Heroku, the application will be online and accessible to both the staff and users. This solution is a scalable one, which means it can be used to manage both day-to-day and emergency operations.

3. Component Overview (one for each)

- Each procedural component should include one section devoted to the detailed description of the component
    - Introduction/Purpose of this Component/Entity
    - Our project will be designed to help Cave Country Canoes improve the management of emergency distress situations and the handling of map-based data on their operations, among the critical tasks that are currently being done by them. With the emergence of such a modern system, the management of processes will be made more efficient, integrating the following:
        - Emergency incident tracking
        - Customer feedback management
        - Route and trip condition logging
        - Resource allocation for emergency response teams
- Input for this Component/Entity
- The Input for this app will be from the customer when they are having an emergency situation on the river. The customer will come to a login screen and they will login to their account. After logging into the app they will come to the Cave Country Canoes map on the next page.On this map there will be a button saying “Contact Representative from Cave Country Canoes with location for emergency”. The user will send their location from the map to CCC. Essentially this will be sent to who is working that

night and the information can be sent to the right Emergency services and get help to them faster.

- Output for this Component/Entity

- The output will be that a representative at Cave Country Canoes will receive a customer's message with their location in an emergency situation. With this they will forward the information to the correct Emergency services and send people out in a boat to get to them as quickly and safely as they can.
    - Component/Entity Process to Convert Input to Output
- Once the customer has downloaded the app and opened it from the app store which could be from Apple App Store, Google Store, or App store, etc. They will log into the app that will be hosted on Flask and Heroku. Once they input their data for their location it will be stored in the database that will be stored and held for a certain amount of time on MySql. MySql will then send the information through the app back to the employee at the time who is responsible for these locations and for forwarding this information. This information should and will be sent to the appropriate Emergency services depending on the situation.
    - Design constraints and performance requirements of this Component/Entity
- A performance requirement for this app will be the customer having their phone handy on the river.

    - if they do not have their phone on the river they of course do not have access to the usefulness of the app.
- Another performance requirement for the app to work as well is being able to connect to the internet or having a service connection along the river.
    - This is a performance requirement as well as a design constraint. If the paddler does not have service at the time of the emergency. They may not be able to contact Cave Country Canoes through the app. At that point they would need to call 911.
    - Another Design Constraint / Risk for the app is server migration: which will happen every 3-4 years when new hardware is procured, data migration would be handled through bulk data export/import using tools like mysqldump for MySQL and reconfiguring the Heroku deployment.

4. Tool overview (one for each)

- Are the open-source license terms compatible with my business requirements?
    - SQLite - Public domain; no licensing restrictions
    - MySQL - GNU General Public License (GPL)
    - Flask - BSD 3-Clause; can be used commercially
    - Heroku - Not open-source; it’s a proprietary Platform-as-a-Service
    - Python - Python Software Foundation License; allows commercial use
    - HTML/CSS - Standards by W3C; open and free to use

- What is the strength of the community?
    - SQLite - Widely used with substantial community support
    - MySQL - Better SQL database standard for production and can be used with Heroku
    - Flask - Strong community with numerous extensions and resources
    - Heroku - Active user base with community forums and support channels
    - Python - One of the largest and most active programming communities
    - HTML/CSS - Fundamental web technologies with extensive communities
- How well is the product adopted by users?
    - SQLite - Embedded in countless applications and devices
    - Flask - Popular for lightweight web applications; used by Pinterest and LinkedIn for example
    - Heroku - Usually preferred by startups and enterprises for rapid development
    - Python - Widely adopted across various industries
    - HTML/CSS - Standard technologies for web development
- Can I get a warranty or commercial support if I need it?
    - SQLite - Commercial support available through third parties
    - Flask - Primarily community-supported; some third-party services offer support
    - Heroku - Offers commercial support plans
    - Python - Community-supported; commercial support available from vendors
    - HTML/CSS - No formal support; extensive community resources
- What quality assurance processes exist?
    - SQLite - has a variety of quality assurance processes:
        - Data Quality Checks: these checks monitor the health of tables by ensuring data is accurate and complete.

    - Performance Monitoring: this process ensures that MySQL is executing queries as expected. MySQL has a server status variable called Question that counts all statements sent by client applications.
    - Container testing: MySQL uses InSpec to test its docker images. InSpec testing is now part of the automated release process for MySQL Server, MySQL Cluster, and MySQL Router docker images.
    - SQL interleaving: this process tests specific features within the server, as well as a larger part of the server. It can be used for regression testing, stress testing, and feature testing.
- Flask - features are comparable to pythons test cases
- Heroku - Maintains internal QA processes for platform reliability
- Python - has several ways to use python for quality assurance:
    - Using Python testing tools: python has built-in testing tools, and you can use frameworks like PytestL
    - Writing tests early: writing tests early in development can help ensure software quality.
    - using style guides: Style guides like PEP8 can help bring consistency to your code.
    - using linter: linters can help identify problem areas and inconsistencies
    - using QA automation frameworks: QA automation frameworks create rules for writing test cases and executing a testing strategy.
    - using tools like Selenium, Robot Framework, and Appium: these tools can be used for test automation.
- HTML - has several ways to use HTML for quality assurance:

        - Performance Testing: Ensures that a product can handle the expected load and user traffic. It can also help identify bottlenecks and scalability issues.
        - Security Testing: Helps identify and address software vulnerabilities
        - Functional testing: Validates the functionality of web applications and ensures they meet their functional requirements.
        - Compatibility testing: Ensures that a product works well with different devices, browsers, operating systems, networks, and other software components.
        - Accessibility testing: Ensures that a website is accessible to all users, including those with disabilities.
        - Integration testing: tests the interaction between multiple components. Integration tests can be performed throughout the website redesign process.
        - Content testing: tests a website to discover mistakes on errors that may not have been noted during web design or development.
    - CSS - features are comparable to HTML cases.
- How good is the documentation?
    - SQLite - Extensive official documentation
    - Flask - Detailed documentation with tutorials
    - Heroku - Offers extensive documentation and deployment guides
    - Python - Comprehensive official documentation
    - HTML/CSS - W3C provides thorough specifications; numerous tutorials available

- How easily can the system be customized to my exact requirements?
    - SQLite - Configurable for various use cases
    - Flask - Highly customizable due to modular design
    - Heroku - Customization within the platform's constraints; supports various buildpacks and add-ons
    - Python - Highly customizable due to modular design
    - HTML/CSS - Fully customizable for web design

    i. Flask is a web based service which reduces the load of the front end trifecta of languages particularly JavaScript and migrates it to Python creating a unified full stack experience. Since the project relies less on JavaScript, CSS, and HTML for the front one can easily add standard Python to do system based work responsible for cryptography, mathematics, etc.

- How is this project governed and how easily can I influence the road map?
    - SQLite: Proprietary; influenced by companies strategic decisions
    - Flask: Proprietary; influenced by companies strategic decisions
    - Heroku - Proprietary; roadmap influenced by company's strategic decisions

    ii. We are aiming to have this system to be web hosted so users can easily access. Heroku or similar would help achieve this. Flask would primarily govern the general application environment and structure for how it works.

    - Python: Proprietary; influenced by companies strategic decisions
    - HTML: Proprietary; influenced by companies strategic decisions
    - CSS.: Proprietary; influenced by companies strategic decisions
- Will the product scale to my enterprise's requirements?
    - SQLite - Best for low to medium traffic
    - MySQL -Usable to high traffic

- Flask - Scalable with appropriate architecture
- Heroku - Designed for scalability; supports scaling applications easily
- Python - Scalable with appropriate architecture
- HTML/CSS - Scalability depends on implementation

iii. We plan on using MySQL for the final design however in the worst case scenario we would migrate the database structure to SQLite. We have two database templates for the project.

1. The client does not tend to get many incidents for which this product would be used. The record number of incidents cave country canoes dealt with in a day is normally 1-2 incidents that are not life threatening. For a life threatening incident there were 30 calls last season (Season normally spans from April to late August) where Emergency Services were called in from the paddlers not knowing where they were or how far away they were from the ending ramp. Some of these calls were life threatening, some of them were not. This is the highest number of cases so far Cave Country Canoes is wanting to make sure the paddlers feel more confident and can reach out to CCC to give them their location and CCC can notify Emergency Services if they are needed and send the location of the paddlers to them. The system also includes the ability to dump old log data a month after it is initially logged. This will help increase the long-term usage of the product

iv. Since Heroku is a cloud based platform it is easily manageable and easily accessible by the users for the service.

- Are there regular security patches?
  - SQLite: there have been security patches

    - CVE-2022-35737: A high severity vulnerability was patched that could allow attackers to crash or control programs that use SQLite. This vulnerability was introduced in 2000 and was exploitable on 64- Bit systems.
    - USN-6566-1: this update fixed several vulnerabilities in SQlite, including one that could cause SQLite to crash and result in a denial of service.
    - USN-6566-2 this update provided the fix for CVE-2023-7104 for ubuntu 18.04 LTS.
- Flask: Yes there are regular security patches
    - Flask Libraries: often release security updates and patches to address vulnerabilities. You can use tools like pip or conda to ensure you're using the latest security updates.
    - Flask-WTF: this popular extension integrates with flask forms and protects against CSRF on your forms by default.
    - Flask updates its system every one to two quarters. The most recent version - 3.1.0, was released on November 13th, 2024.
    - Flask-Security: this extension allows you to quickly add common security mechanism to your flask application, including
        - authentication
        - user registration
        - role and permission management
        - account activation
        - password management
        - two-factor authentication

        - WebAuthn Support
        - JSON/Ajax Support
    - Flask-Talisman; this extension helps protect against common web application security issues by setting HTTP headers
    - SQL injection Protection: this tool provides protection against SQL injection attacks, which can compromise your applications database.
- Heroku - Manages platform security; users responsible for application-level security
- Python: Yes there are regular security patches:
    - Python release cycle: Python versions have a life cycle with different stages. In the “security” stage, only security fixes are accepted, and no new binaries are released.
    - Python 3.8; the final patch release for Python 3.8 was Python 3.8.20, released on September 6, 2024. After that, there will be no more release for Python 3.8
    - Ubunutu; Canonical releases patches for supported versions of Ubuntu to address security flaws in Python. However, older versions of Ubuntu, like 16.04, and 18.04, no longer receive official security support.
    - Automated tools: tools like Dependabot and pip-tools can automate the process of updating dependencies and ensuring they are secure.
    - Security Scanners: Static application Security Testing and Dynamic Application Security Testing can help identify vulnerabilities in Python.
    - Manual testing: developers and pentesters should perform manual testing before releasing a new version.
- HTML: Yes there a 3 security updates for HTML:

    - Microsoft: has released security updates for HTML help, including 890175, which blocks the use of the HTML ActiveX control in remote content.
    - Valve: Has patched HTML injection vulnerabilities in its game Counter-Strike 2. The Vulnerability was in the games PAnorama user interface, which was built using HTML,CSS, and JavaScript.
    - HTML Security: Generally refers to preventing attacks on web apps and websites, such as HTML injection, cross-site scripting(XSS), clickjacking, and tabnabbing. It also refers to protecting HTML content and code from misuse and piracy.
- CSS: Yes there are 4 security updates for CSS
    - Use a content security policy (CSP): A CSP should be set on all responses to all requests, and delivered to the browser in the Content-Security-Policy response header.
    - Use the OWASP CSS Cheat Sheet: this cheat sheet provides recommendations and requirements for securing CSS, including keeping CSS isolated, removing identifying information, and obfuscating class names.
    - Regularly patch HTML Sanitization Libraries: Browsers are constantly changing, and bypasses are regularly discovered.
    - Update your WordPress Theme: regularly updating your WordPress theme can help keep your site secure and compatible. However, updating can be tricky, if you've added custom CSS.

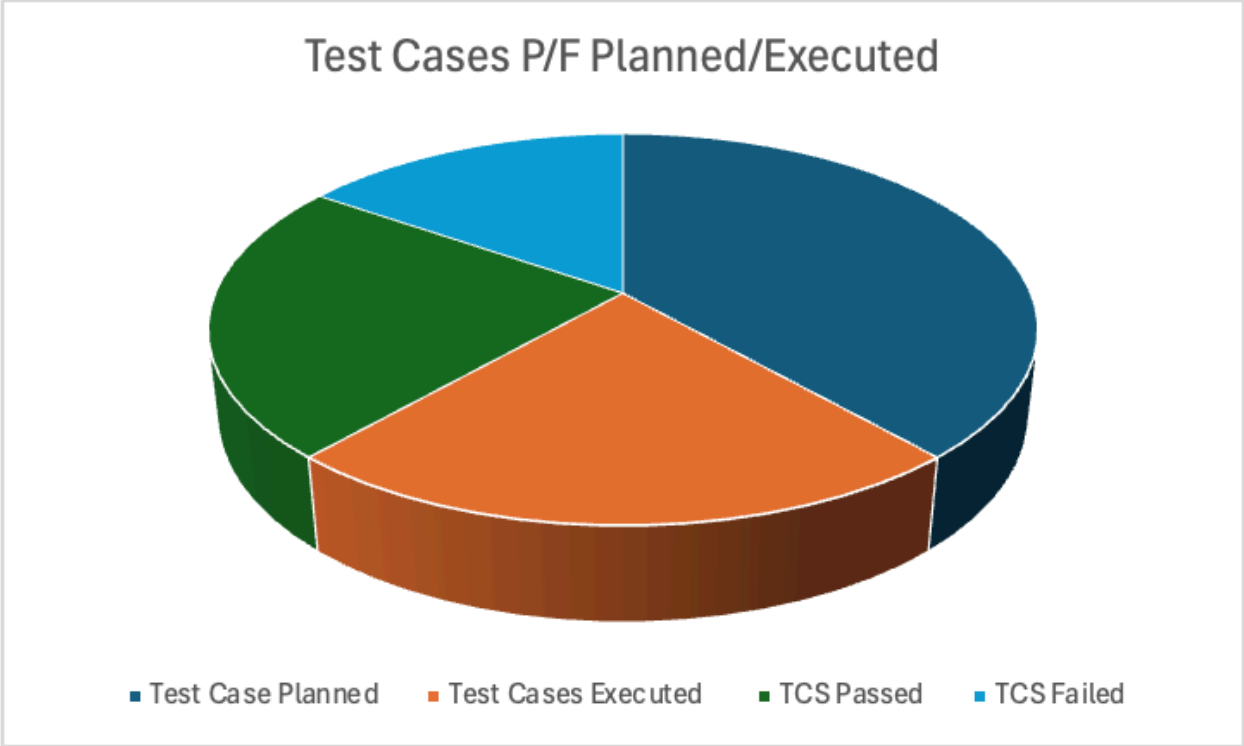
5. Project Repository: [CheyennePierpont6459]

a. Software

- Flask
- MySQL workbench
- HTML

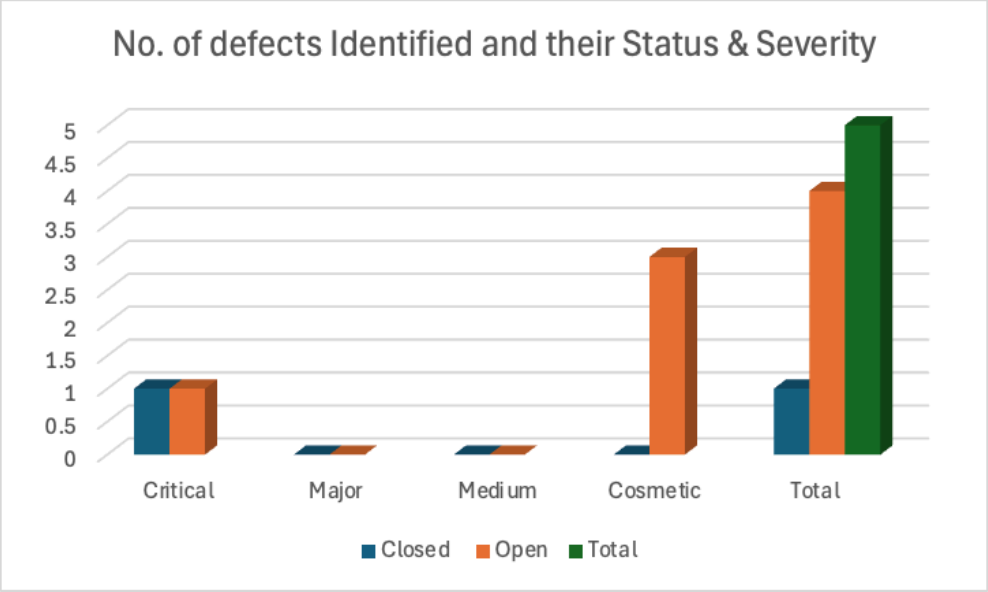
b. Test Cases

Test cases planned	**Test cases executed**	**TCs Pass**	**Tcs Failed**
5	3	3	2



f) No of defects identified and their Status & Severity

	Critical	Major	Medium	Cosmetic	Total
Closed	1	0	0	0	1
Open	1	0	0	3	4
					5

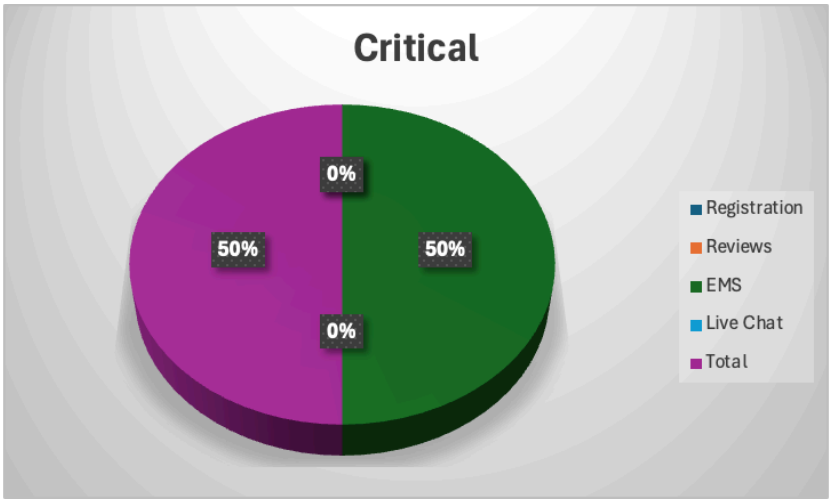


Summary Report

g) Defects distribution – module wise

	Registration	Reviews	EMS	Live Chat	Total
Critical	0	0	1	0	**1**
Major	0	0	1	0	**1**
Medium	0	0	1	0	**1**
Cosmetic	1	0	0	0	**1**
Total-->	**1**	**0**	**3**	**0**	**4**

****EMS issue was fixed and updated through github this is one issue that fits under all three categories**



6. Installation for new install

   - Refer to installation on github repo: CheyennePierpont6459

7. Installation for new platform (preserves data from previous)

   - Refer to installation on github repo: CheyennePierpont6459

8. Further development statement (if I had another year to do this I would …)

   - If we had another year from what we have seen from different projects throughout the semester. There is / a few different tools we could have used to code up the front end. As well as the back end but that comes with experience. This is / was a brand new project. I feel like we did a good job. - Cheyenne Pierpont.