Group 5

Section I

1. User story 1:

Alex, an administrator, can add and remove technician and user accounts from the website. Firstly, Alex would have to register into his admin account. He has access to see each registered technician and user account and is able to have all the previous user privileges of a technician and user account. An example of removing a technician’s account would be when the technician has been resigned/fired from the his/her job. There should be a prompt to ensure that removing the account was not by accident. After removing, there will be a prompt to confirm of the removal. Alex would then log off from the site after completing his tasks.

* 1. **User:** Administrator
  2. **System:**
     1. The system should register the administrator as a separate user with separate privileges. After logging in the administrator should see a different screen than other users see.
     2. In addition to administrative privileges, the system shall give the administrator technician privileges as well.
     3. The system should prompt the administrator for confirmation after he/she makes an administrative decision that will affect other users using the database (i.e., unregistering a user or changing a user’s status).
  3. **Functional:** Add and remove technician and user accounts.

1. User story 2:

Issac, a technician, can update the status of a certain computer part (fixed, not fixed, etc.) from the website. Firstly, Issac would have to log into his account. Upon successfully logging into his account, he would be able to change the status of customer’s computer parts to inform customer of their hardware status. An example of when Issac may visit the website is to change the status of the computer item from processing to completed. Issac would then be prompt with the confirmation and would log out with another prompt confirming that he has successfully logged out.

* 1. **User:** Technician
  2. **System:**
     1. The system should register the technician as a separate user with separate privileges. After logging in the administrator should see a different screen than other users see.
     2. Thesystem shall update the database periodically so that users can see the developments that the technician has made in fixing their hardware or software. The technician is also allowed to update hardware to be serviced on the system (The customer is not allowed to do this).
     3. The system shall provide the technician a means of posting comments that the user will be able to see as well as updating the status of the project from “in progress” to “finished”.
  3. **Functional:** Update the status of computer parts and add computer hardware that needs servicing.

1. User story 3:

Blaine, a user/regular customer, is allowed to view the status of all computer parts being serviced and is able to add more computer parts that need servicing to the website. Firstly, Blaine would have to register into his user account. Upon successful login, he would be able to go to his personal online account to view the status of all computer parts being serviced and would be able to add more computer parts if needed. An example of when Blaine may visit the website would be to track the status of a computer part and view if the part has been repaired. Upon seeing the status, Blaine would log off from the site with a prompt confirming his account has been logged out.

* 1. **User:** Customer
  2. **System:**
     1. The system shall register the customer as a separate user with separate privileges. After logging in the customer should see a different screen than others see.
     2. The system shall allow the customers to submit software to be serviced but not hardware.
     3. The system shall allow the customers to see the status of the hardware/software that is theirs and not the status of other customer’s hardware/software.
  3. **UseCaseDiagramTechHelpDeskSystem.pngFunctional:** View the status of the computer parts and add more computer software that need servicing.

**General Functional Requirements:**

1. The system shall provide a way for customers to securely pay for the service.
2. The system shall remove a user’s account if it is inactive for 2 years (730 days).
3. The system should provide all users a means of accessing their password if they have a valid email/username to provide the system.

**General Non-Functional Requirements:**

1. The development team is using a waterfall development method. This constraint could add extra time and money to the project depending on whether our customer requires changes in the software after the implementation process.
2. Traffic on the server will only limit the functionality of the system if 1,000+ users are trying to access the server
3. A user has to be registered in order to use the system. Guests are not allowed.

Section II

1. Will employees that adopt this system need retraining in order to make use of it?
   * There will likely be an adjustment period where users of the system would need to either be formally trained, or have some time to experiment around with the interface to get a feel for the workflow. The interface will be designed to make this process easy and quick.
2. What significant benefits will users experience by adopting this system?
   * Users of the system (both employees and customers) should find that ease of communication with the opposite party (techs with customers, customers with techs) is increased. Users should also find that due to simplified process of reporting and replying to issues with ticket postings, response time should be decreased. The increase in process simplicity and speed will benefit all parties involved.
3. What associated costs will come with the deployment and adoption of this system?
   * There will be a time cost, as employees must go through a period of training and/or experimentation with the product to fully understand its capabilities and workflow. The time cost will also come in the form of administration of the server hosting the products, and maintenance of the product (adding new users, applying updates, etc). The monetary cost will come in the form of paying for a server to host the application, although this can be mostly eliminated by deploying a virtual appliance inside existing infrastructure (HyperV, bhyve, etc) as well as any licensing fees that the company must pay to maintain support for the product.

Security and Legality

1. How will passwords be stored?
   * Hashing paired with bcrypt, or sole bcrypt
2. How will user permissions be limited?
   * User roles will be implemented, which allow users to view/edit only data that is allowed by their role (implement with mysql views and hierarchical roles object system)
3. How will the server hosting the database be secured?
   * SSH will be administered via pubkey authentication only, with yubikey for 2fa. Fail2Ban will be enabled and configured to reduce server load from unauthorized login attempts. The MySQL database will be configured locally on the server, and will not be exposed via a web interface to the internet (such as PHPMyAdmin or similar), and the database server will not run any extraneous servers. It will be running a GRSec patched kernel on Debian 8.
4. What personal information will be stored on users who are registered with the system?
   * A user’s phone number, street address, or email address may be stored depending on their needs for ways to be contacted, and where to ship their serviced system back to (if it needs to be shipped).

Alex Jenness - completed required section 1b (use-case diagram)

Isaac Boyd - completed required section 1b (list of requirements)

Blaine Herro - invited group to Google docs, Powerpoint; completed required section 2 (security/legal questions, feasibility analysis)

Joey Xiong - completed required section 1a (user stories)