

CityTown Form System

Website description

Group #3:

Alexus Depeza, Kyle Desrosiers, Daniel Gjuraj, Jonathon Nosowitz

**Table of contents**

**Domain analysis…………………………………………………………………………………….2**

**Requirements document……………………………………………………………………….….4**

**Use cases…………………………………………………………………………………………….6**

**Modeling……………………………………………………………………………………………...8**

**Testing strategy.……………………………………………………………………………………22**

**Test results.………………………………………………………………………………………….23**

**Ethical considerations.…………………………………………………………………………....30**

**Project plan ………………………………………………………………………………………....31**

**Algorithm(s)………………………………………………………………………………………….39**

**Cyber security fulfillment……………...……………………………………………………..…...40**

**Screenshots of GUIs/User Guide………………………………………………………………...42**

**Planning for Version 2.0…………………………………………………………………………...60**

**Progress reports…………………………………………………………………………….……....61**

**Domain Analysis:**

One of the key problems facing the public sector (federal, state, local governments) not just in the United States but abroad is constituent service. A key component of that is the fact that the governmental sector tends to be behind the private when it comes to issues involving the use of new technology, social media, etc… which should be used to enhance service. In recognition of this problem, we are working to develop a system to improve the current state of a local town website. The goal is to create a more user-friendly and cost effective system, which streamlines services. The ultimate goal is to begin development of smarter cities. Currently used systems lack efficiency and security; and requires the applicant to print certain forms, manually fill them out and then either scan them onto their computer to email them or mail them to the town clerk’s office. The new system will eliminate this time consuming process since we are implementing a new interactive burglar alarm form, that was chosen to showcase features from all aspects of the system, updates to the front and back-end dashboard, and a new user interface to allow submission of applications and payments online to the appropriate departments. Since the user enters sensitive information into the system, we are going to improve the overall security to ensure that data maintains its integrity. We are going to enable the departments to review a form sequentially so that it will be the only one able to review the form at a time. Once the department submits its review, the next appropriate department receives the application. This process repeats until all the departments submit their evaluation. The applicant is then notified as to whether their form was approved or denied by a notification in the dashboard and an email. During the review process, both front and back-end users will be able to check the status of the application. The front-end user will be able to submit any complaints or concerns to be handled by the appropriate department(s). Implementing these features will allow the town to streamline the application process and shorten the timeline necessary to review the applications.

**GLOSSARY:**

Constituent - a member of a community or organization that has the power to appoint or elect

System - intercommunicating components of software

Smart City - an urban area that uses different technological methods to collect data in order to manage assets or resources efficiently

Front-end - the part of a program that the consumer interacts with (e.g. user interface)

Back-end - the part of a program that the supplier interacts with to perform daily functions and maintenance

Dashboard - a part of the user interface that organizes and presents the information in a coherent manner

Sensitive information - a user’s private information (e.g. credit card number)

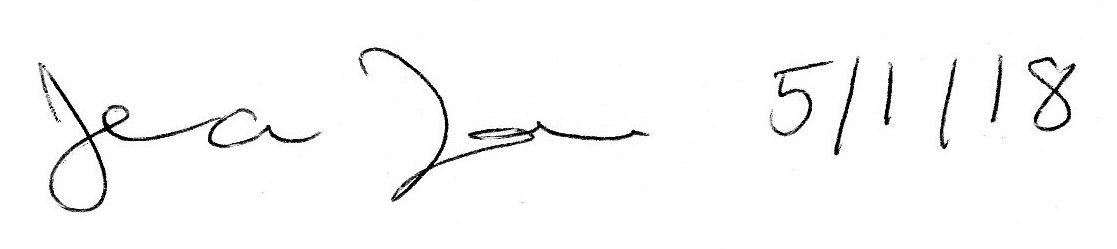
Complaint - the user’s feedback on a system function that has made them discontent

Concern - the user’s feedback on a system function that could be improved

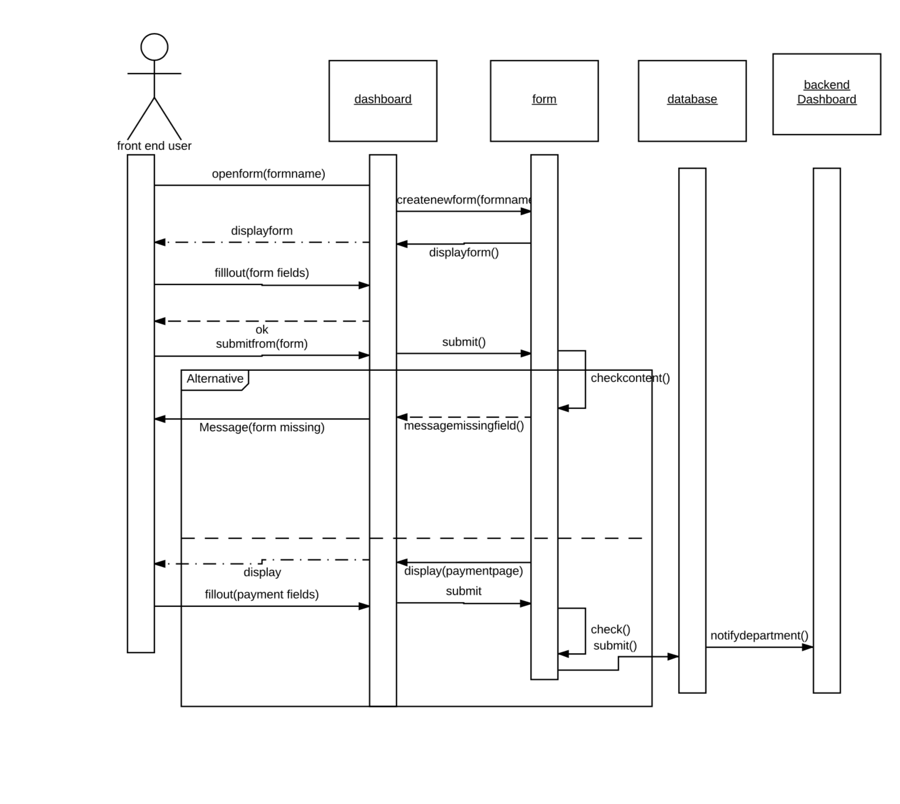
User interface - the part of a program that the user has direct interactions with

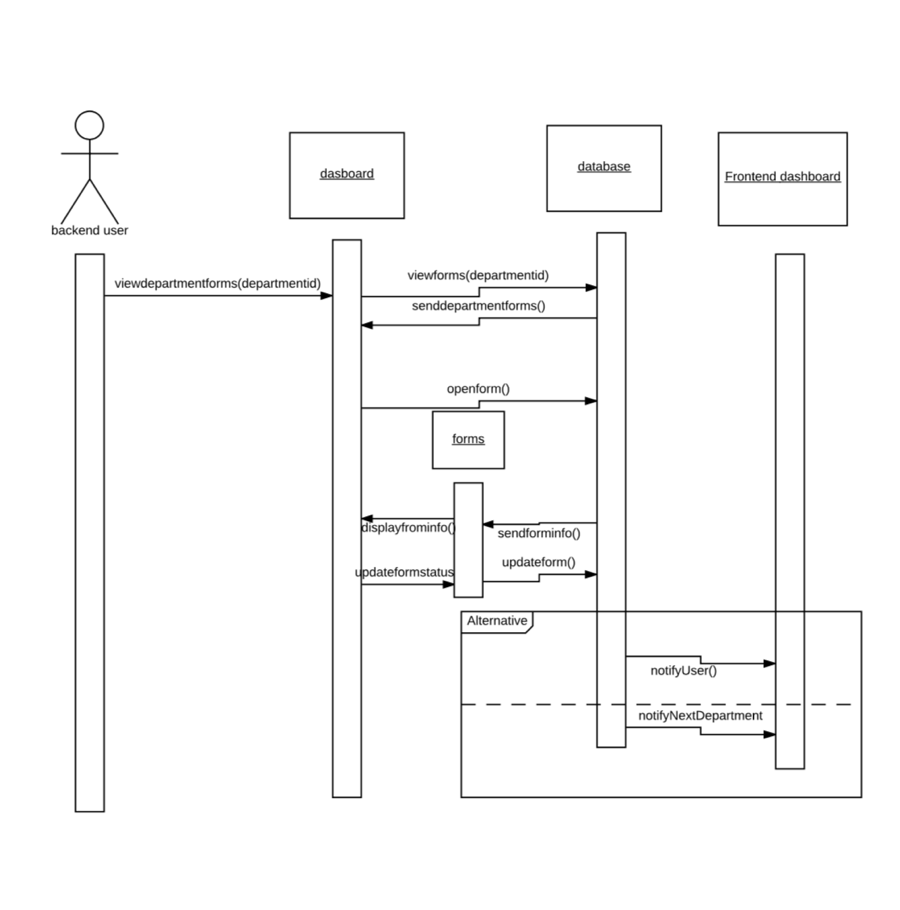
**USER REQUIREMENTS:**

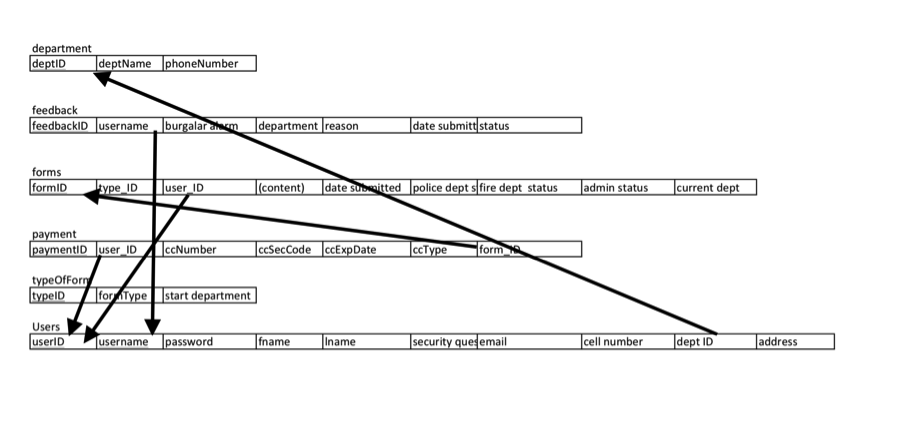
|  |  |  |
| --- | --- | --- |
| **User Requirement** | **System Requirement** | **Rationale** |
| 1. Login page (front-end) | The user must create a username and password to fully use the system and to be able to submit a form or concern | Allow the user to manage his account information, forms and concerns |
| 2. Interactive burglar alarm form | Allow user to enter information interactively; the system should check for valid input. | Make application easier and more convenient for user |
| 3. Secured sensitive information | The user’s private information will be encrypted | Maintain the integrity of the user’s information |
| 4. User’s digital signature | The user will be able to digitally sign the form as a form of verification and confirmation | Verification and confirmation of user submitting a form |
| 5. Concern/complaint page | The system should notify the appropriate department when a user submits a concern | Improve the system and user experience |
| 6. Front-end dashboard | The user can check the status of forms and concerns | Improve user involvement in review process and ensure process is efficient |
| 7. Front end search function | The system should sort through the applications submitted to find the one that matches the search criteria | Enable user to find a form easily to edit information quickly |
| 8. Login page (back-end) | The user must create a username and password to access the information available to his department | Allowing user to manage applications and complaints but only those related to his department |
| 9. The forms and concerns will be processed by the appropriate department(s) | The dashboard will show a notification when there is a new submission and allow the department to notify the user when they addressed the concern | Ensure user experience is best possible; ensure system is working properly |
| 10. Back-end dashboard | The system will display application and functional notifications | Ensure jobs are identified as they are submitted to keep review process as continual as possible |
| 11. Back-end search function | The system should sort through the applications submitted to find the one that matches the search criteria for that department | Allow user to review a form or complaint a second time or contact applicant in case of discrepancy |
| 12. Back-end digital signature | The back end digital signature verifies the department’s decision on the form | Verification and confirmation of the department’s review decision |
| 13. Back-end notifications | The dashboard will highlight the notification of another job | Ensure it is known when a job is submitted so that it is handled in an efficient manner |
| 14. Super user with access to all files and forms while each department has limited access | The system will prevent a department from accessing data that it should not have access to; Super user can override departmental decisions | Access control maintains a level security and ensures that the integrity of the data is maintained |

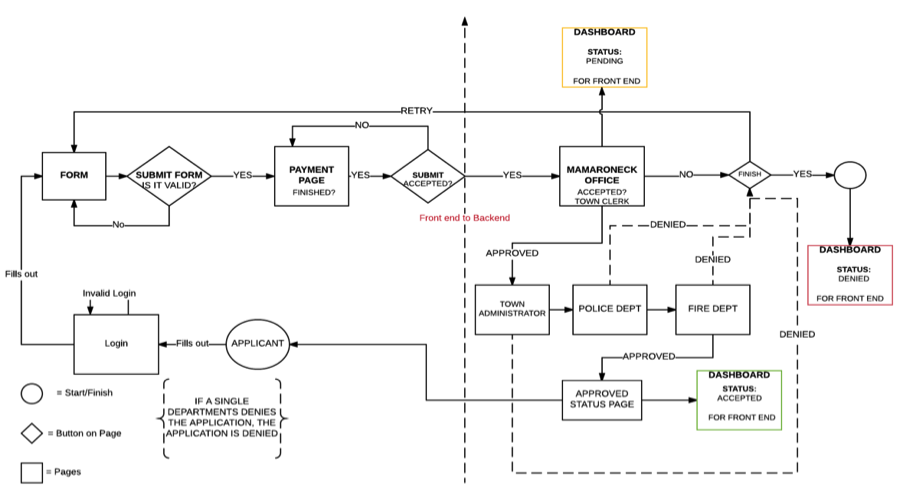
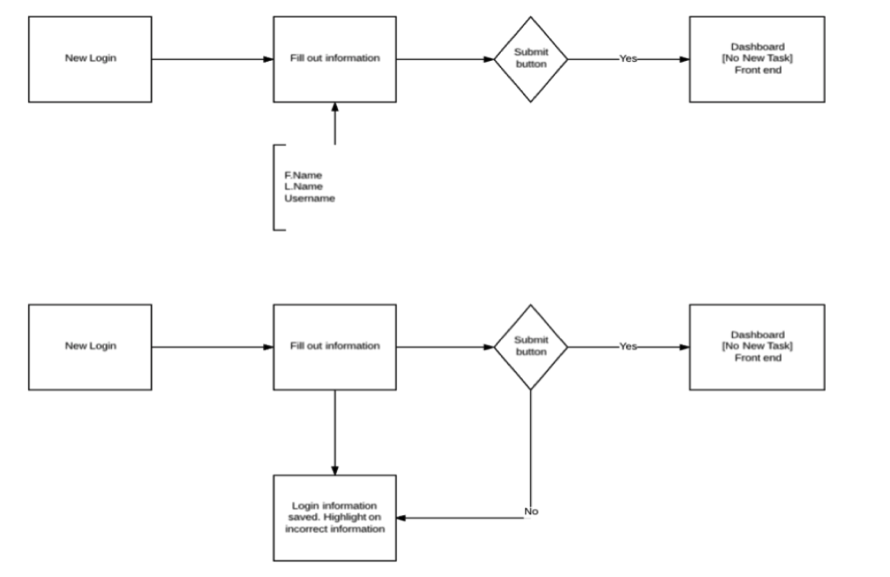
****

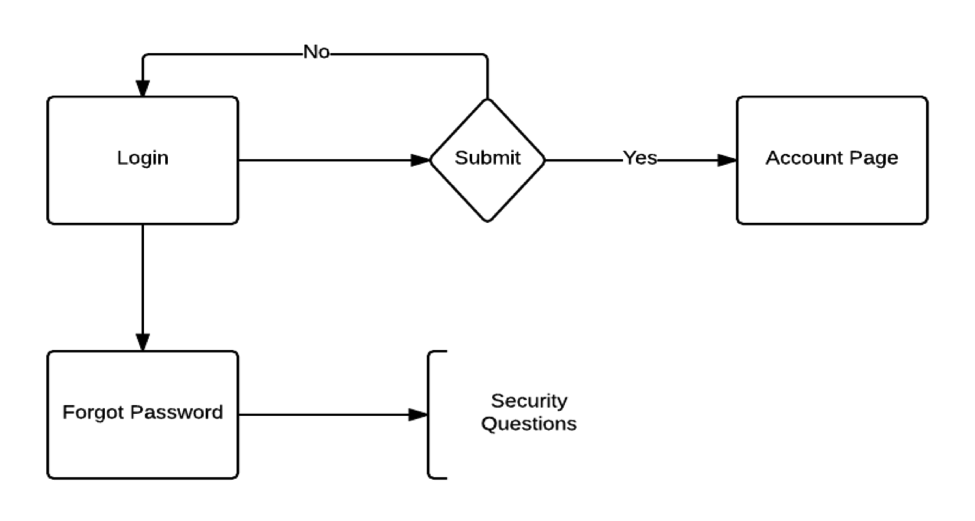
**Use Cases**

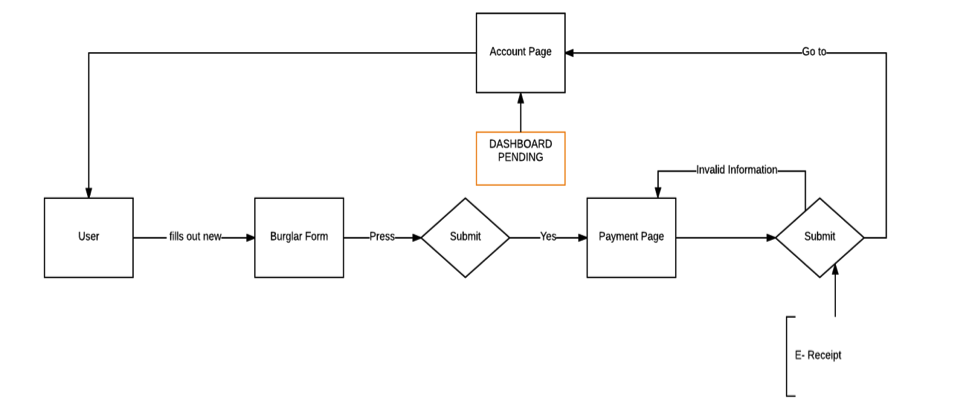
****

****

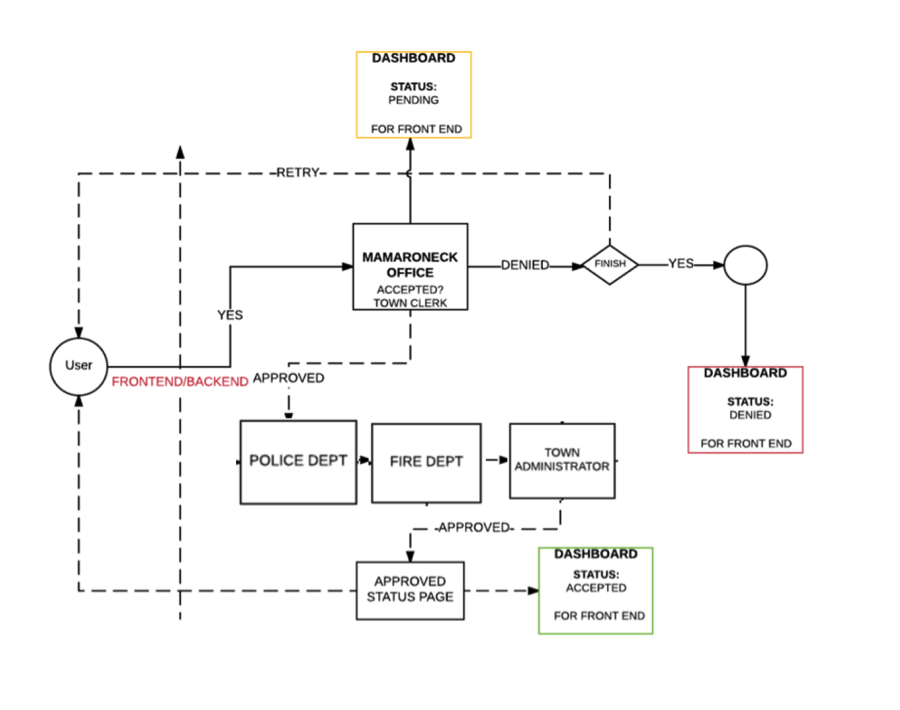
**Modeling of different aspects of the system  
  
  
ERD diagram  
  
**

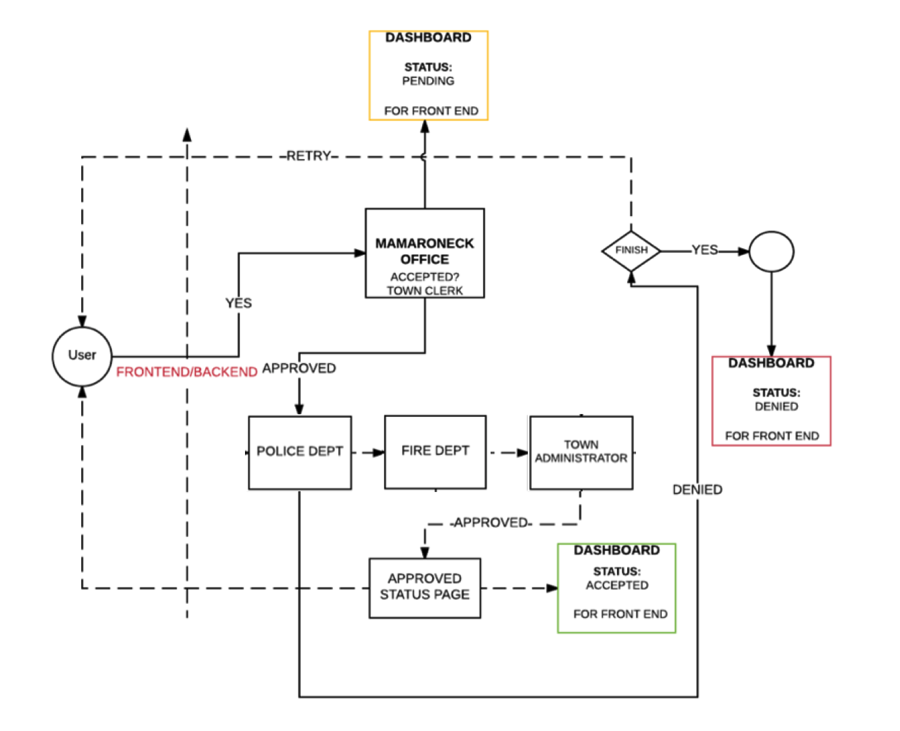
**WORKFLOW DIAGRAMS:  
Overall Workflow Diagram:  
  
   
  
New User Login:   
  
**

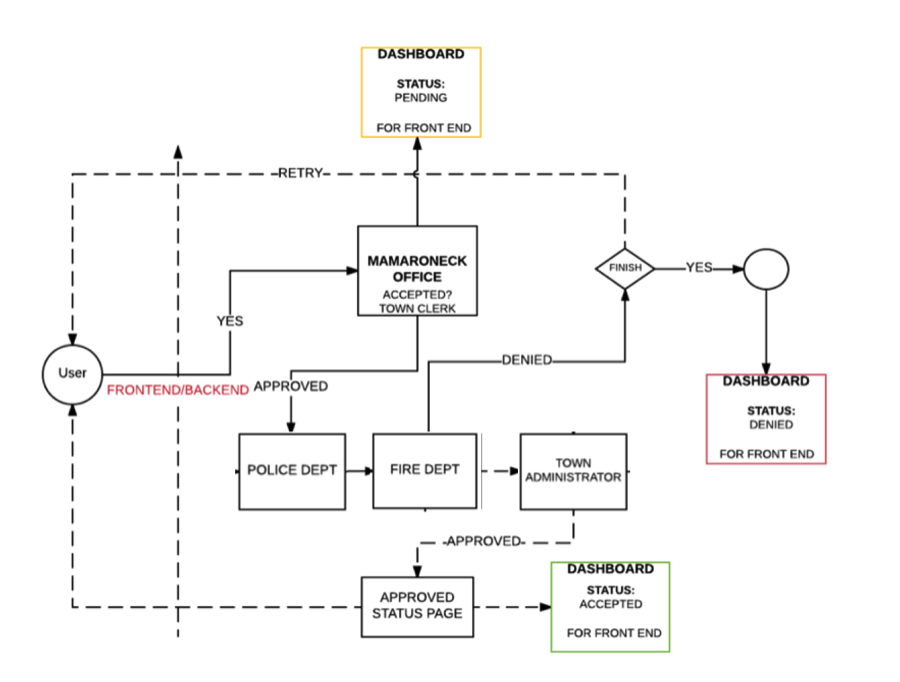
**Returning User Login:  
   
  
  
New Form (front-end):  
   
  
  
  
  
  
  
  
  
New Form Invalid Form Information (front-end):**

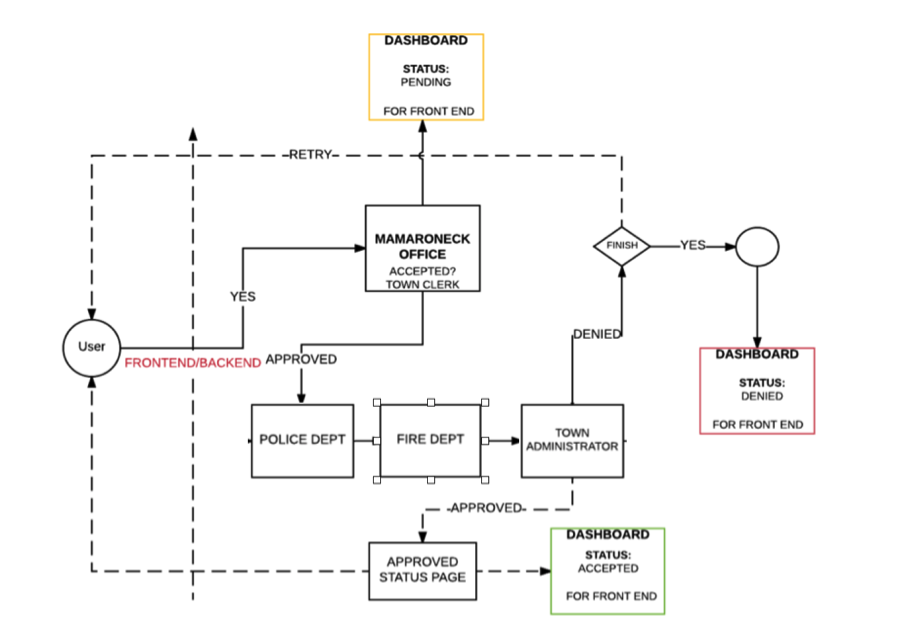
****

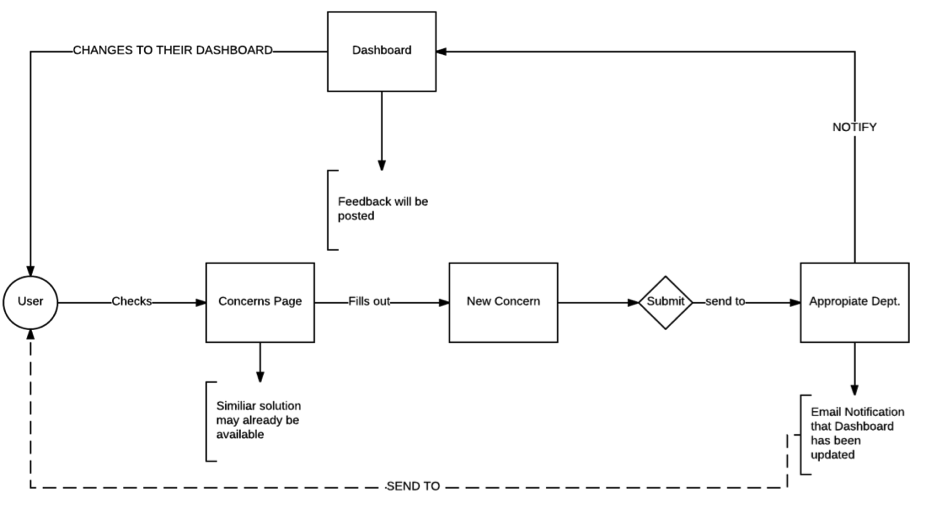
**New Form Invalid Payment Information: **

**Back-End Process Denied at Town Clerk:  
   
  
**

**Back-End Process Denied at Police department:  
   
  
**

**Back-End Process Denied at Fire Department:  
   
  
**

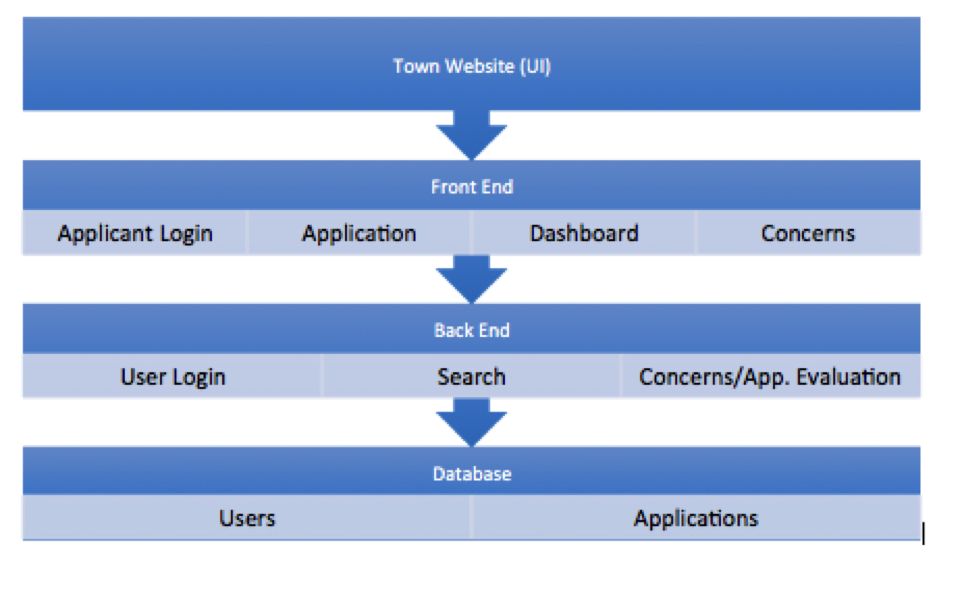
**Back-End Process Denied at Town administrator:  
 **

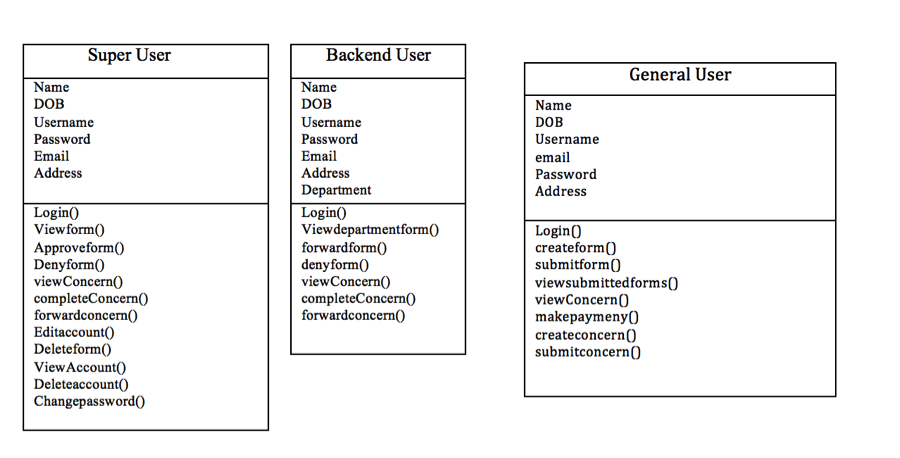
**Concerns Workflow:   
**

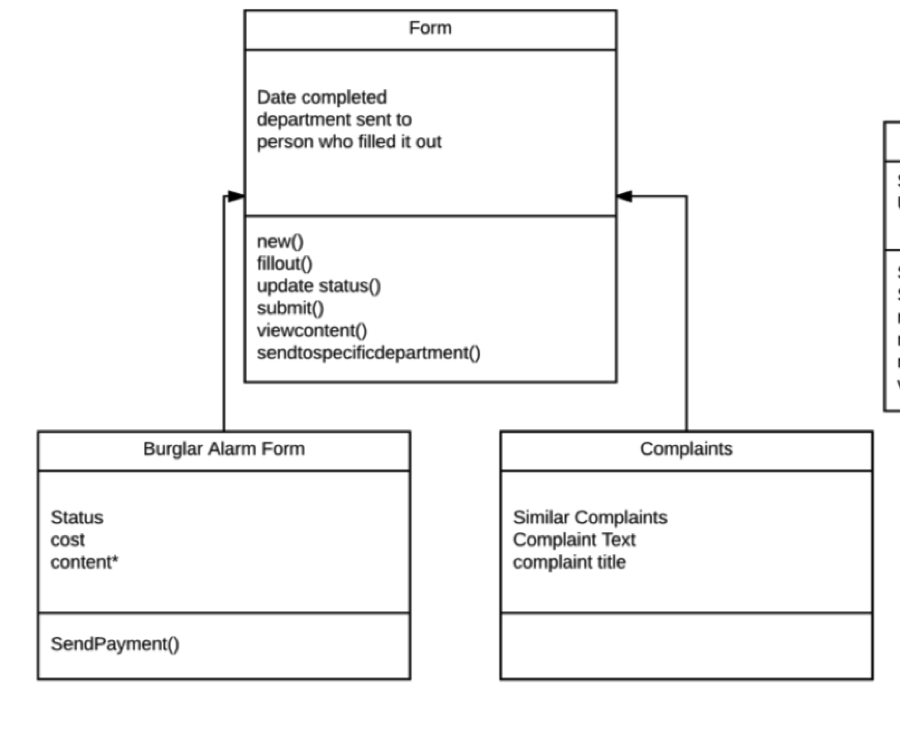
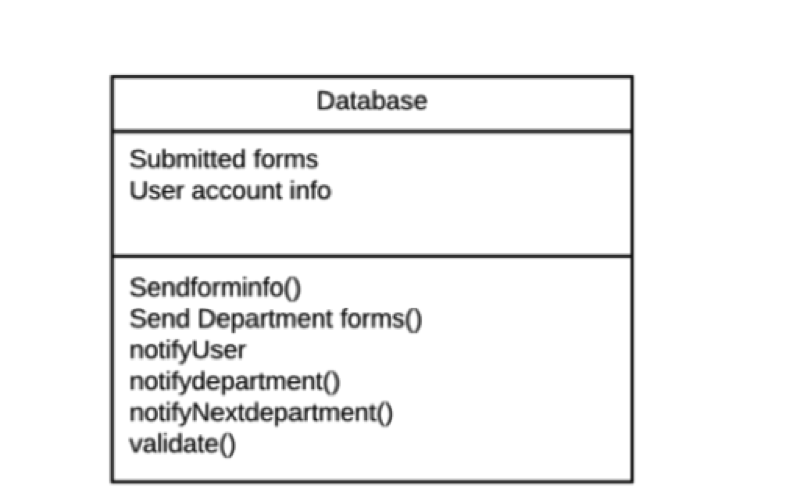
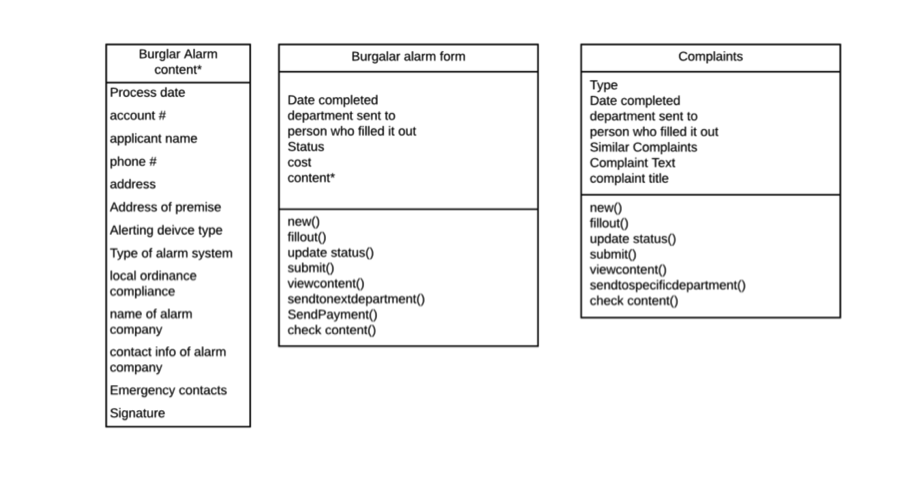
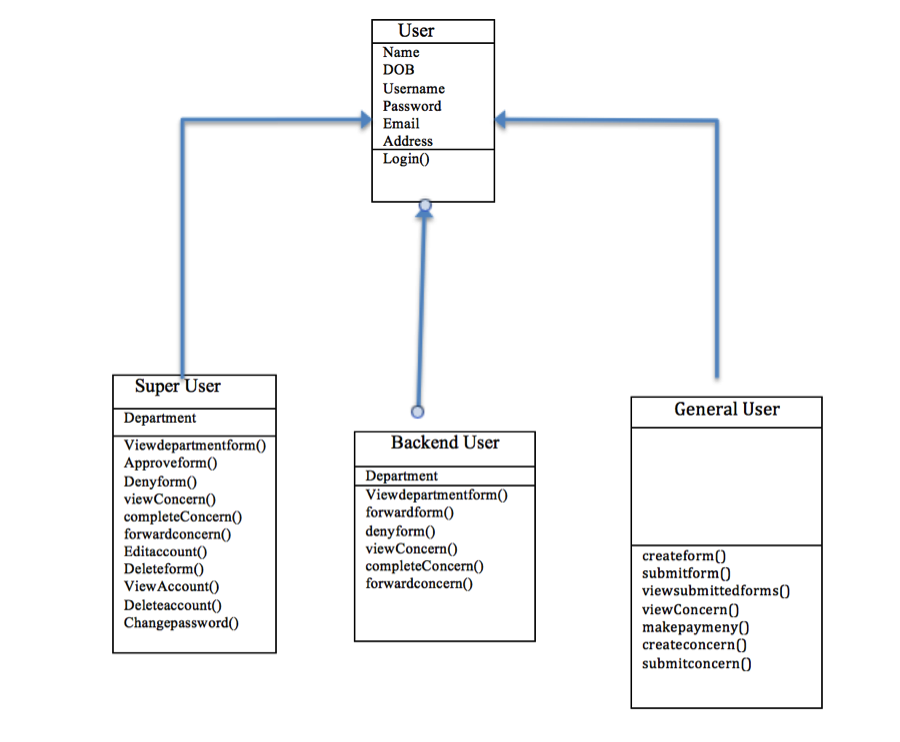
**Design Considerations:**

Since the new system is intended to be implemented as a web page, the most significant design consideration is going to have a clean flow to the layout. The system is going to need to be easy for people to navigate through, allowing experience to be simple. The system needs to make it clear to the user where they need to go to find what they are looking for. Another area of focus is reducing the amount of clutter on each page. The system should only have the necessary information and buttons for the process of each page. The goal is to make the system as simple and understandable as possible. Another focus is on functionality of both frontend and backend of the system. For this the backend of the system should not do anything with a user’s account unless the user submits a form or concern to the backend. For the frontend user, they should be able to see the progress of the backend with submissions. In order to successfully complete these tasks, we used ajax, bootstrap, jQuery and the w3 library.

**Architectural Organization:**Layered Architecture fits the system best because it allows the system to be modified or updated as long as the main interface is maintained. Due to the localization of the architecture, certain parts of the system can be modified or fixed without having to change the entire system. This is important for the town of Mamaroneck system because as the system develops and more layers are created, more and more functions will be available to the users.

**Architectural Design: **

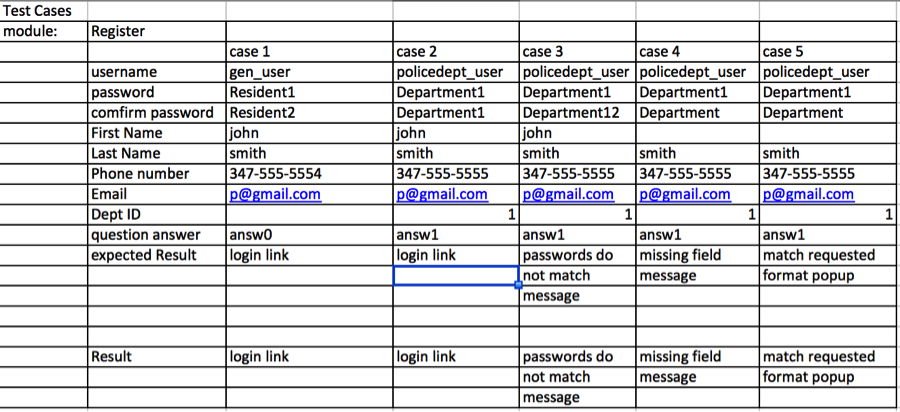
**Class Diagrams  
   
**

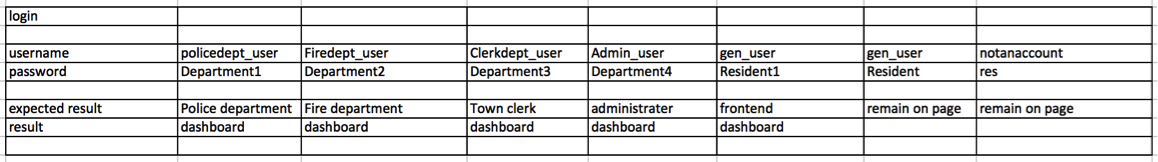
****

**Testing strategy:** The strategy to testing this website was to create a form and concern and follow it to its approval. This process begins by creating an account for each backend department, which are the Fire department, Police department, Town Clerk, and Town Administrator, and a general user account. We first sign into the general user account. Signing in opens a dashboard that displays all forms that the logged in user has submitted, which should be nothing at this point. To create a form, we will click create form, fill it in, then fill in the proceeding payment page. Next we will create a new concern by clicking the new concern tab, and fill in the content of it, and for the sake of this test, enter town clerk as the directed office. Once we go back to the dashboard, we will see a new form and a new concern appear on it, along with its status. Now we must log out of the general user account and sign into the town clerk account. Signing in with the Town clerk’s username and password will bring us to the town clerk dashboard, which contains both the form submitted, and concern submitted. Checking off forward, and picking police department for both the concern and the form will make the form disappear from the dashboard after you hit submit. The next step is to sign into the police department, and see if the form has appeared on the police department dashboard. If it has we will repeat this with the fire department, then sign into the town administrator’s account, and check off approve on the form and complete on the concern that appears in the dashboard. We will then sign out, and sign into the general user account to see if the form appears as approved on its dashboard. All these steps should be repeated, except, by checking the status of the form before its final approval to make sure the current department that is viewing it is shown, denying it at a different point to make sure a denial is shown.

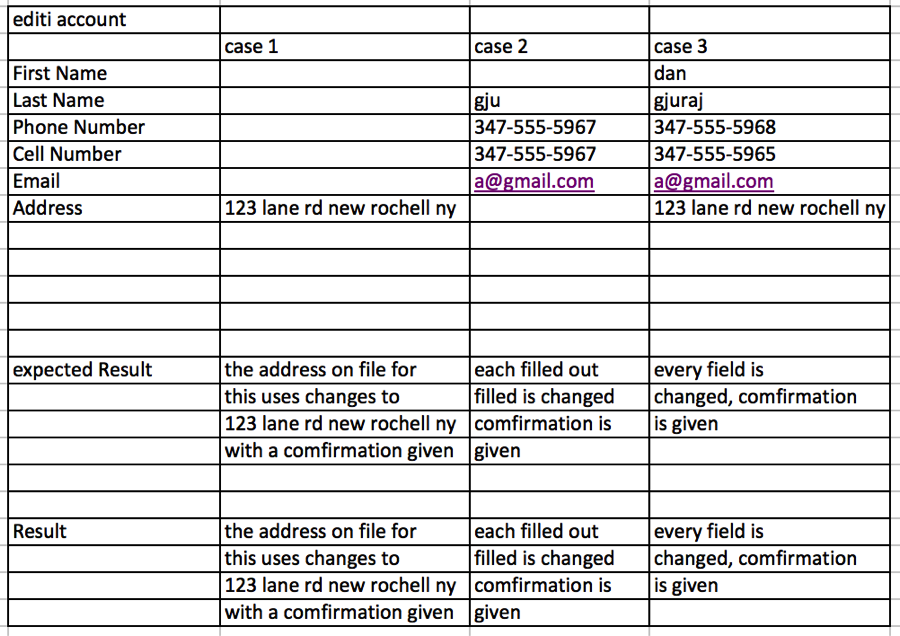
We also conducted testing with the client. We displayed the front end capabilities in the first meeting and the back end features in the second meeting. The client was happy with the features of the front end, but had some suggestions for aesthetics as well as the information that was displayed in the dashboard such as the status and type of form. The client also wanted to ensure that we were going to have the super user displaying all information somewhere on the page. We made the suggested changes and highlighted these changes at the second meeting along with the back end functionalities. Again, the client was happy with the overall appearance, but wanted us to fix the way the user was able to view a submission such as a form or feedback entry. We made the appropriate changes and the client was satisfied with the product we had presented.

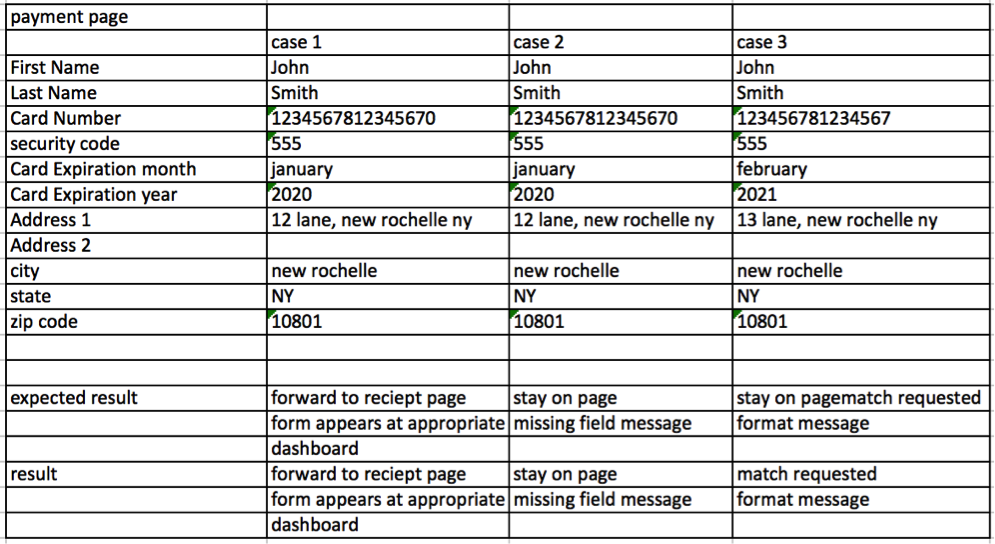
Additionally, we met with the security professor to ensure that our security was up to standards. We showed her each individual feature listed in the security requirements document on page 42 and she was content with the way we secured and tested any potential vulnerabilities.

**Test Cases  
  
Registering   
 **

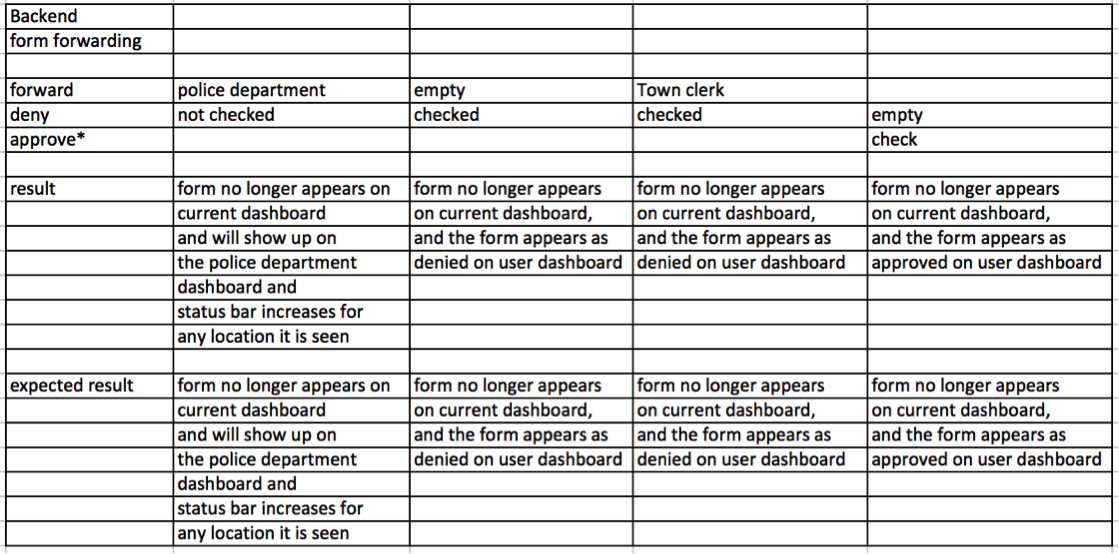
**Login  
  
 **

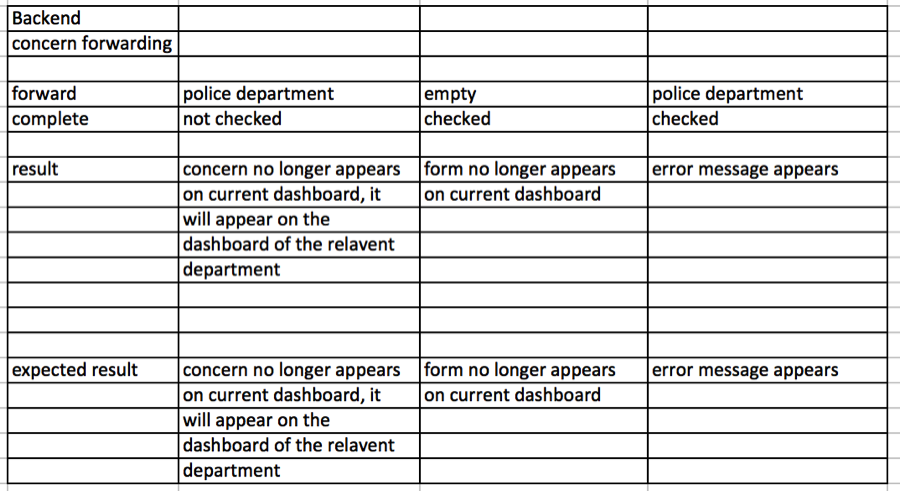
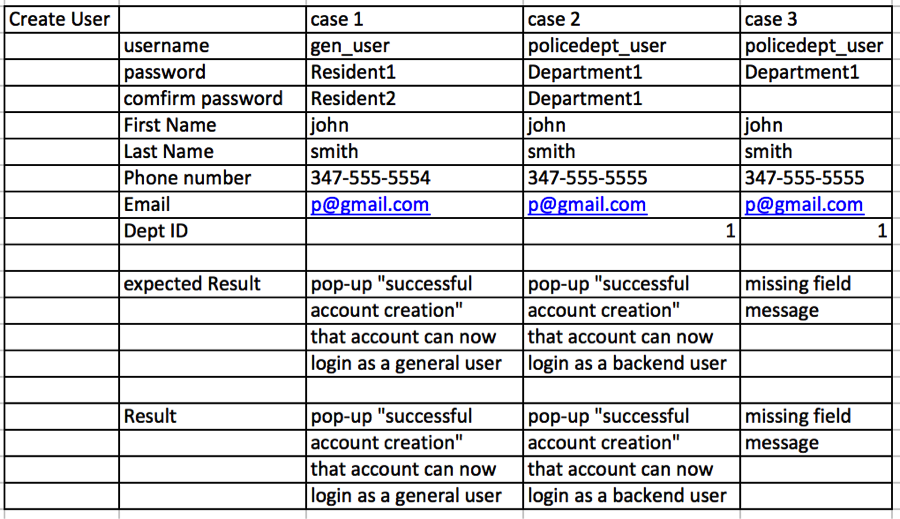
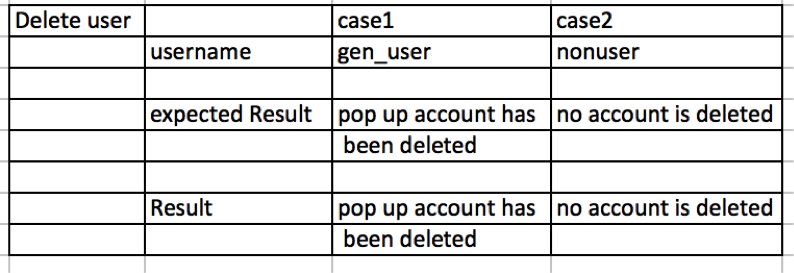
**Frontend: Form page**

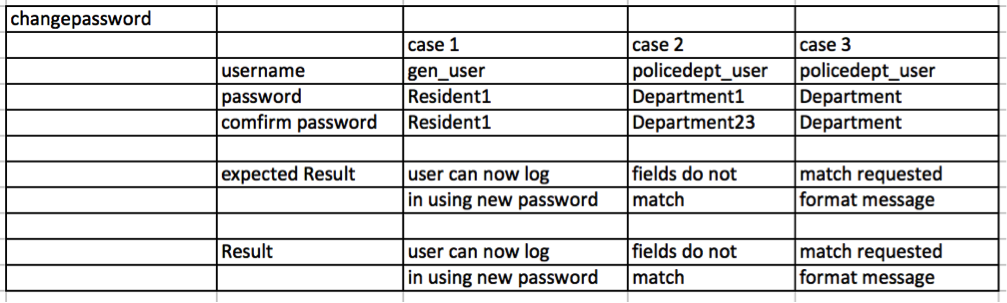
****

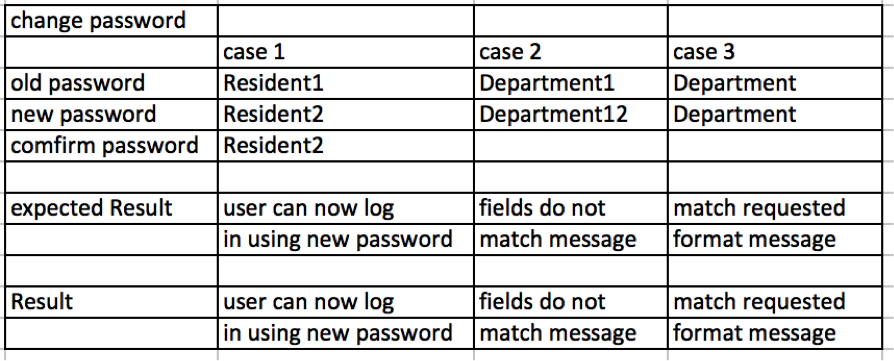
**Frontend: payment page  
  
 **

**Frontend: concerns page  
 **

**Backend: forward form  
 \*approve only appears for administrator account**

**Backend: concerns  
   
  
  
  
Town Administrator: create user  
  
   
Town Administrator: delete user  
 **

**Town Administrator: change password   
 **

**Account settings: change password  
  
 **

**Ethical concerns** The major ethical concern when putting together the website was security of critical user data. When making the website, there was a ethical responsibility to make sure the personal data of the users is sufficiently secured as it was being entered, and making sure it remained secure once it was in the database. The most pieces of information to keep secure were passwords and credit card information to make sure only they can access their own accounts and their card information remains private. We did this by hashing passwords when registering and logging in by using passwords verification that compare to the hashed password. We also sanitize all input variables in order to remove vulnerable characters, which could be used in malicious attacks, further details of our security procedure is in the cyber security section of this document.

**Project Plan**

**Estimation:**

Length of time ~ 4 months

Number of people - 4

Resources - 4 computers with wifi, text editor, Sequel Pro, MAMP

**Project Timeline:**

January

Create database

Begin developing front-end

Begin developing interactive form

February

Continue developing front-end

Begin adding security

Begin creating back-end

Begin testing

March

Connect front-end to back-end

Create super user account and functionality

Continue adding security

Continue testing

April

Finish adding security

Finish group testing/fixing any bugs

Begin client testing

**Risk Analysis:**

Code Malfunctioning - Restructure code

- Change functionality

- Seek help

Requirements Change - Update client on amount of change possible in project term

|  |  |  |  |
| --- | --- | --- | --- |
| TASK | Effort (person days) | Duration (days) | Dependencies |
| T1 - Front end interface | 1 | 14 |  |
| T2 - Front end dashboard | 2 | 14 | T1 |
| T3 - Front end security | 2 | 14 | T1, T2 |
| T4 - implementing form to UI | 1 | 7 |  |
| T5 - Back end interface | 1 | 7 | T1 |
| T6 - Back end dashboard | 1 | 7 | T5 |
| T7 - Back end security | 2 | 14 | T5, T6 |
| T8 - User database | 1 | 7 | T1, T5 |
| T9 - Secure user | 2 | 14 | T8 |
| T10 - Connecting front end to back end | 4 | 14 | T1-T8 |
| T11 - testing | 5 | 28 | T1-T10 |

Weeks

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13+ |
| T1 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| T2 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| T3 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| T4 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| T5 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| T6 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| T7 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| T8 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| T9 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| T10 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| T11 |  |  |  |  |  |  |  |  |  |  |  |  |  |

Weeks

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13+ |
| Alexus | T1 | T1 | T5 | T6 | T3 | T3 | T7 | T7 | T9 | T9 | T10 | T10 | T11 |
| Kyle | T1 | T1 | T5 | T6 | T3 | T3 | T7 | T7 | T9 | T9 | T10 | T10 | T11 |
| Daniel | T4 | T1 | T2 | T2 | T3 | T3 | T7 | T7 | T9 | T9 | T10 | T10 | T11 |
| Jonathon | T8 | T1 | T5 | T2 | T3 | T3 | T7 | T7 | T9 | T9 | T10 | T10 | T11 |

**Test Plan:**

Dataset

Login information

Username

Password

Create a new account information

First name

Last name

Email

Phone/cell number

Username

Password

Address

Security question

Department ID

Form Information

Name

Phone number

Address & zip code

Address of alarmed premize & zip code

Alerting device(s)

Type of alarm system

Local ordinance compliance

Alarm service agency

Name of alarm company

Address & zip code

Phone number

Contact person  
 Emergency contact

Electronic signature

Concern/Complaint/Question

Department

Type of feedback

Content

Payment Information

Credit card number

Expiration date

Security code

First name

Last name

Billing address & zip code

City

State

Phone number

Email address

Town Employee Information

Date form processed

Account number

Unit Testing

These specific cases are the most complex and are most likely to cause an error in the case of an update

Form Submission

This too contains an extra complexity because when a user creates a new form, this will create a new form in the database as well as notify the appropriate department. The form must follow the department order shown in the workflow diagrams earlier in this document to ensure that the right department is notified first and is not sent to all departments.

Form Update

A backend user should be able to update a form’s status. They should be able to either deny or forward a form. To deny the form would stop the form’s progression and forwarding a form to a department twice will be prevented by checks. These processes would set off a trigger which would either only update the database or update the database and send an automated notification to either the next department that there is a new job available for review or the user that submitted the form informing them of the form’s approval or denial. This would require checking the database and invoxes of the appropriate back and front end users to ensure that this works.

View Form (Departmental and General User)

For the front end user, viewing the forms involves displaying a list of all forms the user submitted. This involves querying a list of those forms so that when the view link is clicked, the form is displayed in a pleasant way that is also printable if necessary. The back end user version of this allows the user query a list of forms that relates only to that user’s department so that when they click on one of the forms, they are able to see all the necessary information in a printable way to deny or forward the form.

These processes are also applied to any concerns, comments, or questions that general users submit.

System Wise Testing:

Login/Registration

Input: username and password, email, department ID, security question

Check: that all fields have input and ensure that input is valid by comparing information to that of user database

Output: successful login redirects the page to the user’s account page; unsuccessful login returns to login page; new user redirects to registration page and user is able to create account

Form

Input: form information (i.e. name, address, alarm company name)

Check: to ensure that all required fields have input and information is “correct” (no numbers in name field)

Output: The user is able to submit form and move on to make the proper payment and receives appropriate notifications

Security

Input: user’s sensitive information

The system prevents attacks on user’s sensitive information

Output: unchanged data or notification that data was attempted to be hacked or was hacked and the user is able to perform usual system functions

Submit button

Input: any form, login, payment information

Check: that the button redirects the user to the next appropriate page

Output: confirmation (i.e. e-receipt, pop-up, notification), page redirect

Payment

Input: payment information

Check: all fields have input and are verified

Output: e-receipt and confirmation alert and the user is returned to his dashboard and is able to perform system functions

Submit Form

Input: completed form with “correct” information (no numbers in name field)

Check: that the form is sent to the back end with a notification to the appropriate department

Output: confirmation email and notification; the user is able to return to the account page and perform system operations

Front end

Input: any submitted forms, incomplete (saved) forms, denied forms, feedback/concerns

Check: the dashboard has updated record of forms and feedback

Output: status of the forms and the feedback, notifications when status is updated; user is able to submit other forms and payments

Back end

Input: forms or feedback from front end

Check: that department is able to receive forms/feedback pertaining to that department

Output: notifications of new pending jobs; the user is able to view and process that submission as well as other submissions for that department

Front end/Back end change

Input: pending forms (status)

Check: the status of the pending forms

Output: the user is notified of the status change in the dashboard; the user is able to view the dashboard notification and perform system operations

Acceptance Testing:

Login

The user will only be able to login if they provide the correct username and password

New users will be able to register if they provide a unique username and password

A user that forgot his username/password will be able to recover them by providing the appropriate information

An error message if any unsuccessful attempts occur

Fill out form

The form will only accept valid data to be submitted; if data is invalid, an error message will be raised

The user is able to re-submit a form if it is denied due to errors

Submit

When the submit button is pressed, the form/information is sent to the back end (appropriate department) and a notification is raised in the dashboard; error if unsuccessful

Dashboard

Each time a form is submitted, or the status is updated, the user receives a notification in the dashboard or an error message if unsuccessful update occurs

Payment

The user receives an e-receipt each time a payment is successfully made; an error message when payment is unsuccessfully made

Security

If a user’s sensitive information is hacked or attempted to be stolen, the user receives a notification; otherwise data is properly and correctly encrypted then decrypted by system

**Algorithm(s) For Major Computing**

Registration

Check that all fields have input

Verify data (i.e. password meets criteria stated on registration page)

If data is valid, submit information to database, create notification of successful registration and bring user to login page

If data is not valid, raise error

Login

Check that all fields have input

Verify input with data in database

If data is valid, send user to appropriate account page

If data is not valid, raise error message

Submit Form/Payment

Check that all fields have input

Verify data (i.e. no numbers in name field)

If form data is valid, submit form to appropriate department, alert submission confirmation and send user to payment page

If form data is not valid, raise error

If payment data is valid, create confirmation (e-receipt) and send user back to account page

If payment data is not valid, raise error

Submit Feedback

Check that all fields have input

If all fields filled out, submit feedback to appropriate department, raise submission confirmation alert, send user back to account page, update list of feedback submitted

If not all fields are filled out, raise error message

Update Form/Feedback Status

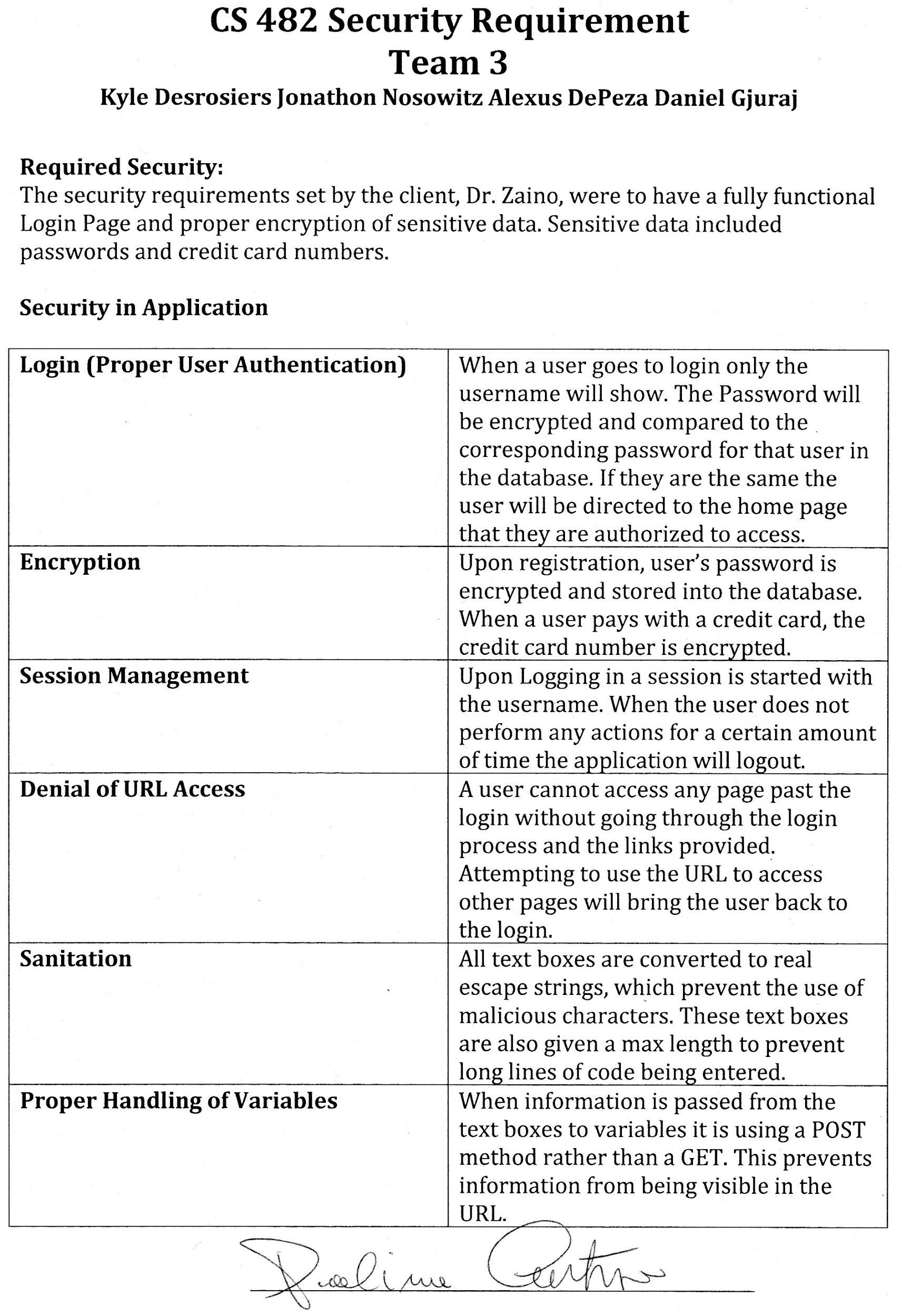
Check that update fields are set

If fields are set, update information in database, send submission to new department as well as notification to user that his submission was updated and department to inform that there is a new job available

If fields are not all set, raise error alert

**Cybersecurity Fulfillment:**

To ensure that a user’s personal information is secure there are multiple security algorithms implemented in the system. When registering, the sensitive information such as the user’s desired password is hidden on the screen and encrypted in the database. Additionally, for the password to be accepted, it must meet certain criteria to ensure that the user creates a strong and secure password. The user also answers a security question so that if he forgets his username or password, he will be able to recover it. When the user is logging in, the password is again hidden on screen and both the username and password are kept out of the address bar to prevent any intrusions. Both the username and password fields as well as all other input text fields are sanitized to prevent SQL injection. Once the user is logged in, a session starts. This logs the user out after a certain amount of time if there is no activity. When the user is making a payment, all credit card information will be encrypted before it is entered into the database to prevent credit card theft. The user is also able to change their password should he feel that he needs to create a stronger password. Any user must use the links in the site to navigate the system; he is not able to type the link to a particular page in the system into the address bar to move to that page. To further protect the front-end user’s information, each department employee is only able to see the information that pertains to his department.

****

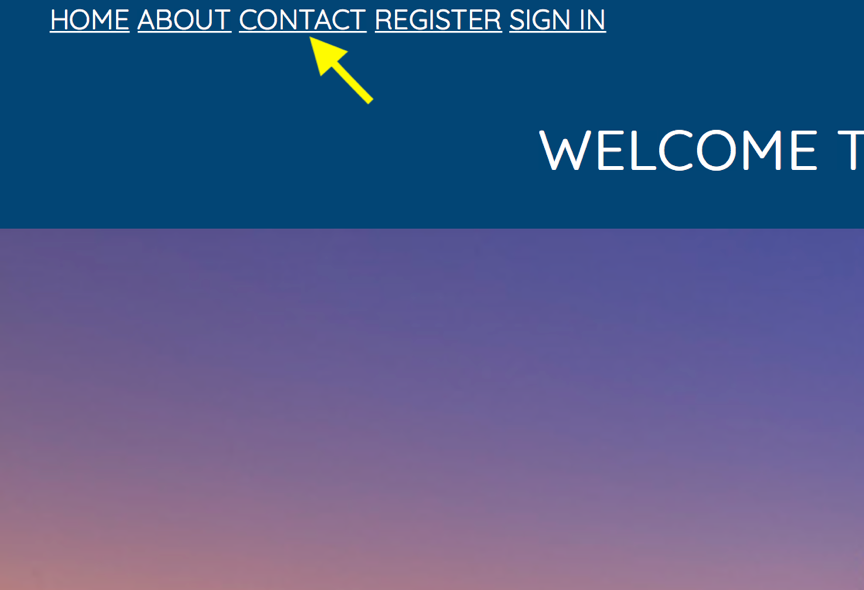
**User’s Guide**

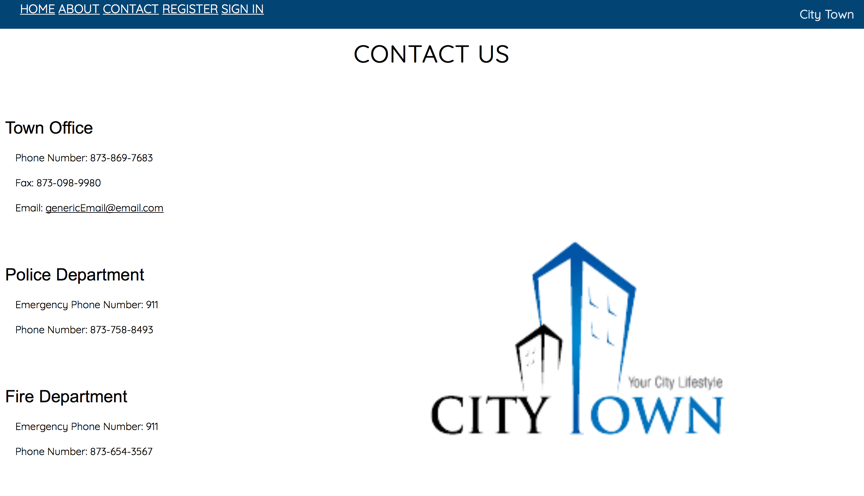
**Home**

****

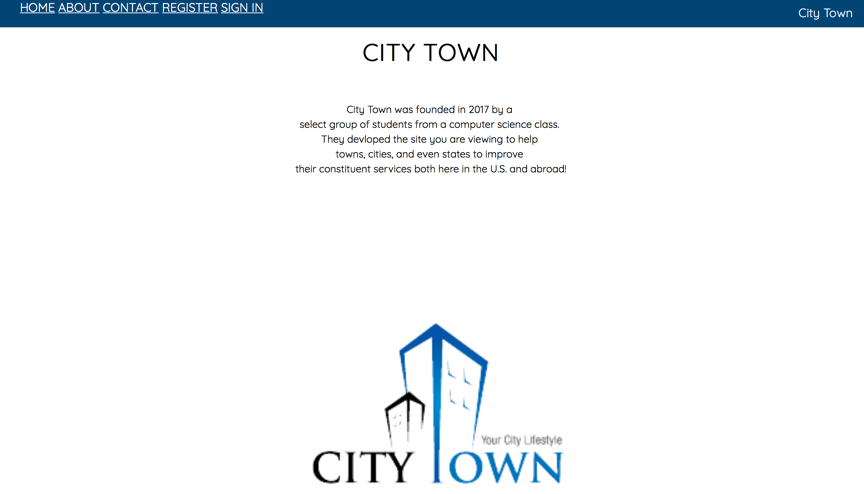
User determines on home screen whether they want to read the about, get contact information, sign in or, make an account by clicking the appropriate link.

If you click on **Contact**, information for contacting any of our town options will appear as shown below.



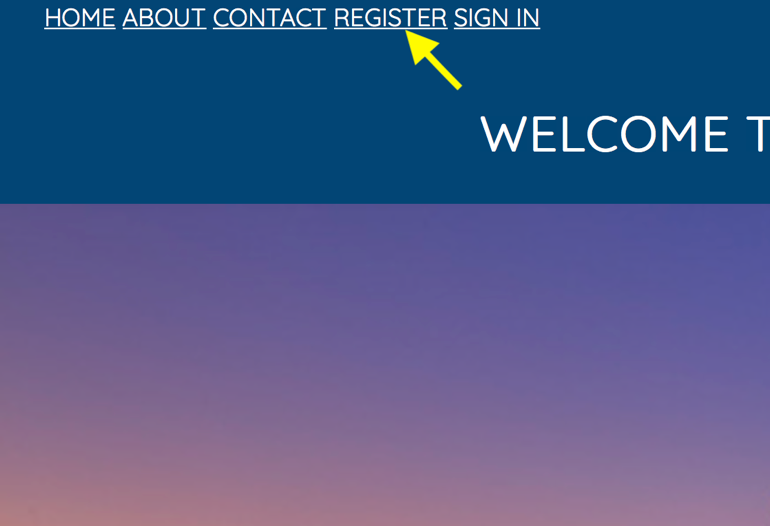
****

If you click on **About**, you will get a general description of the purpose of this website.

****

**Creating an Account**

To create an account, click on **Register** on the home page.



**Register**

1. User determines if they are General User or Employee

a. Employee: Someone who works for the town

b. General User: Any other user

If User clicks **General User**:

1. Choose a username

2. Choose a password

3. Enter first name

4. Enter last name

5. Enter primary phone number

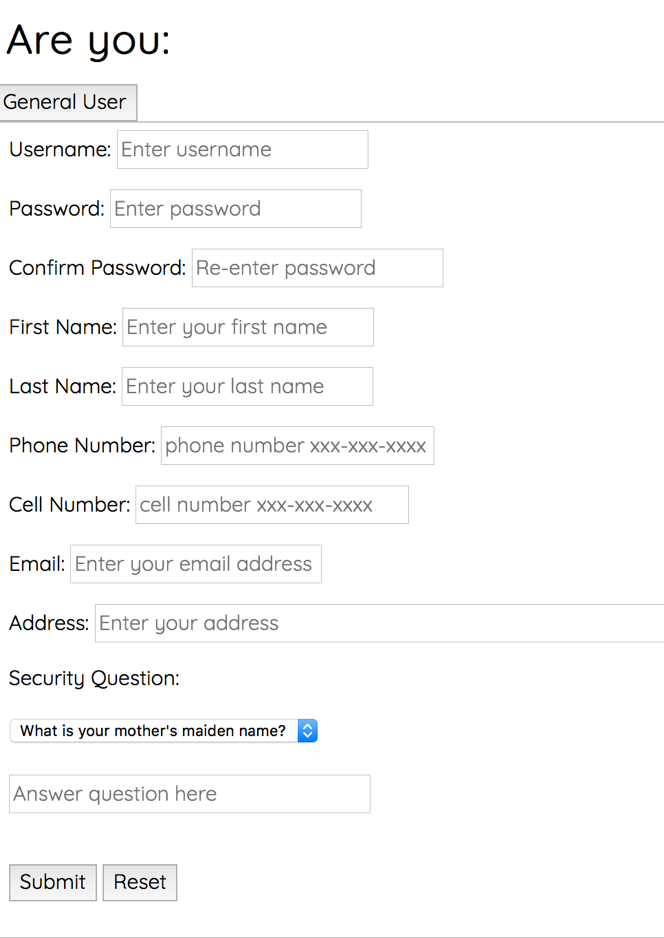
6. Enter Cell Phone (optional)

7. Enter email

8. Enter Address

9. Fill out security question

10.Press submit- General User will be prompted to sign in with their new account

****

**If User clicks Employee:**

1. Choose a username

2. Choose a password

3. Enter first name

4. Enter last name

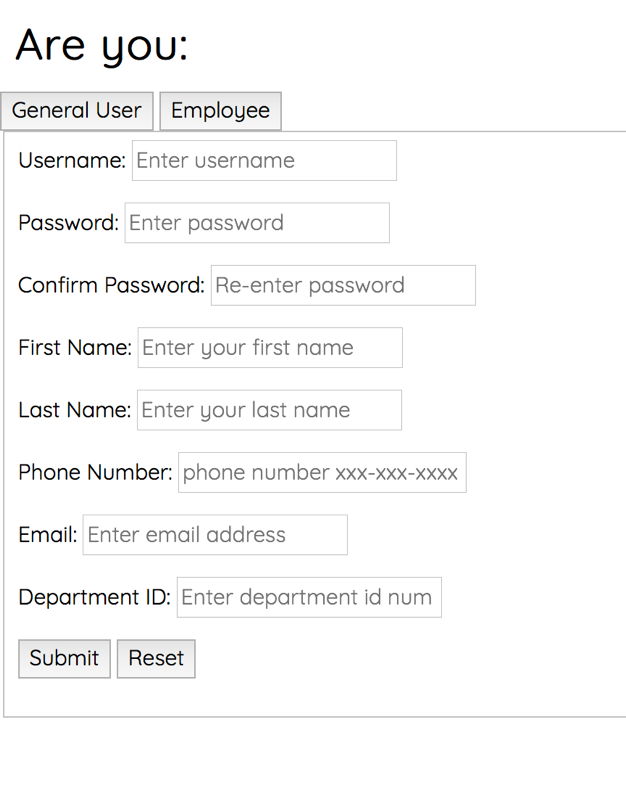
5. Enter primary phone number

6. Enter email

7. Select Department

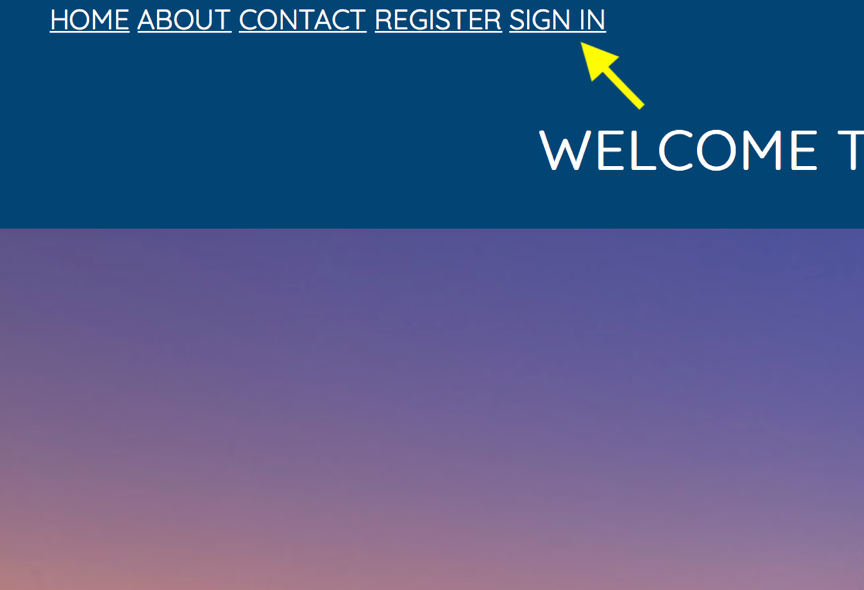
a. Fire, Police, Town Clerk's Office, Super User

8. Press Submit - Employee will be prompted to sign in with their new account

****

**Login:**

Click on Sign In on the home page or after you register

****

**Sign-In Page**

1. Enter Username and Password

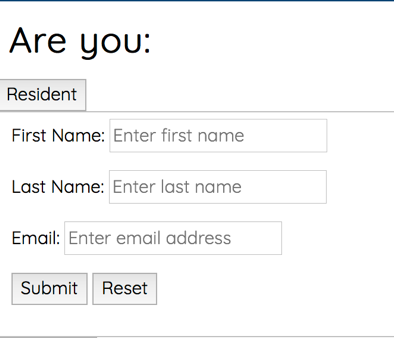
2. Press Submit- General user is taken to account home page

If General User forgets username

1. Enter first name, last name, and email.

2. Check email for following instructions

3. Once username is retrieved sign in as normal.



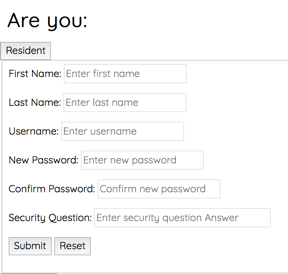
If General User forgets password

1. Enter first name, last name, and username.

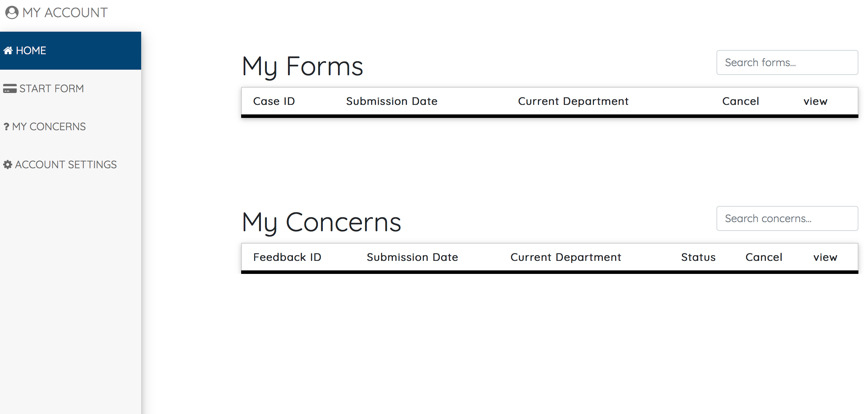
2. Choose a new password

3. Enter the answer to your security question

4. Press submit and sign in as normal



**General User Account Page**

****

**·** Search: Allows user to look for any form or concern within their respective table



· Dashboard- Allows user to see all activity such as pending forms and pending

concerns

· Start Form- Allows user to fill out an application to be submitted for review

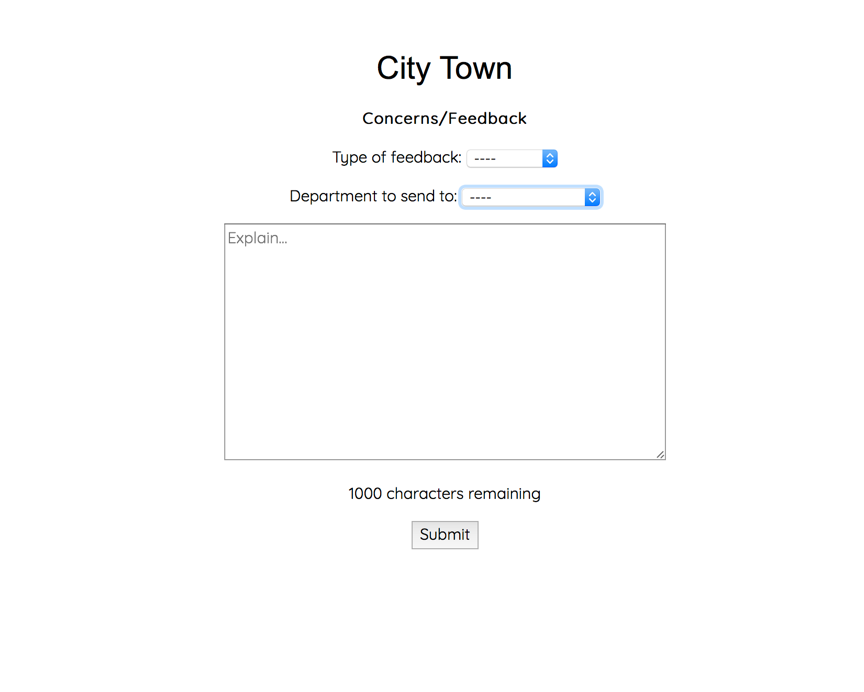
· My Concerns – Allows user to submit a concern, comment or question

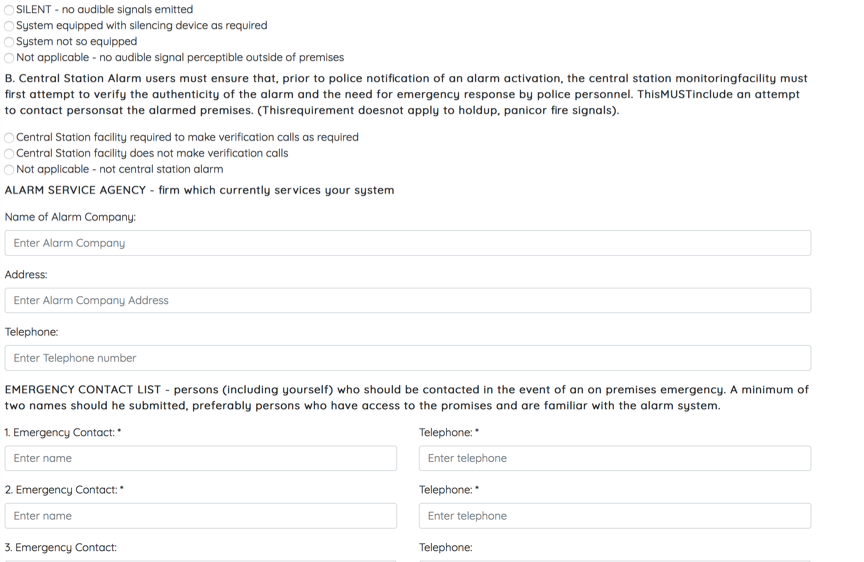
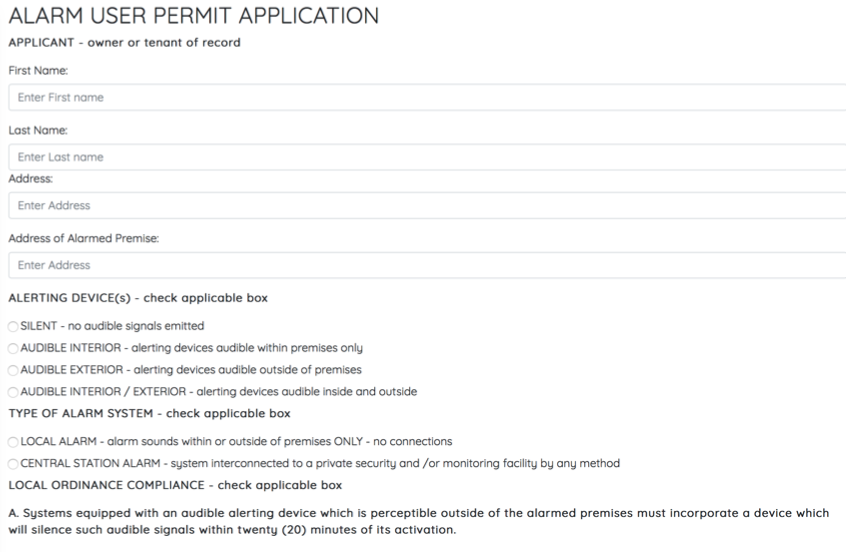
· Account settings – Editing account information, you will have the options of changing your personal details on your account or changing your password. If you click change your password, you must enter your old password on the box specified and enter your new password on the next text box along with confirming it in the third box. If you click on edit personal information, just enter into the box that you want to change your information and hit submit.

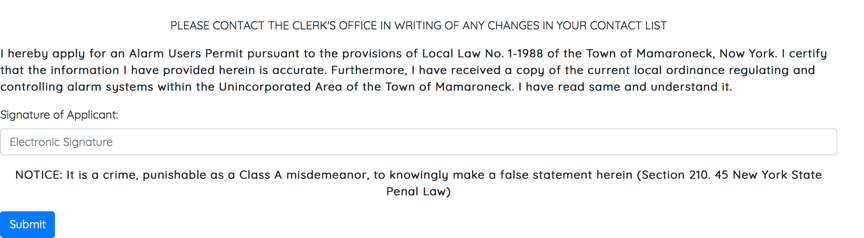
**Concerns Page**

* User chooses the type of feedback
  + Questions: Any question in relation to your account, town website,
  + Comments: Any thoughts in relation to the town website or account
  + Concerns: Any issues you may face within the website.

* User chooses the department: If user knows where their feedback needs to go they may select which department it get sent directly to.
* They then use the space available to submit their feedback.
* Once submitted a popup notification should appear letting you know it was submitted.

****

**Creating a form:** Fill out all the information required on the formr****

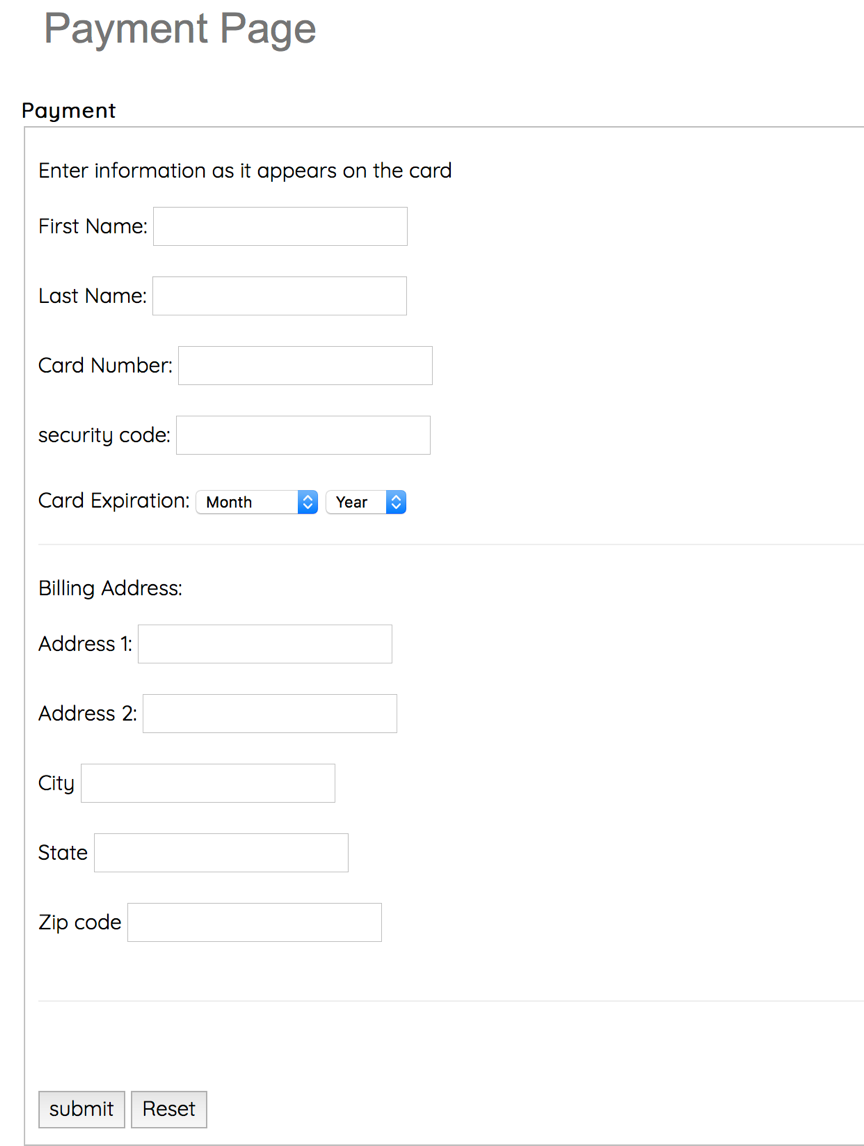
****

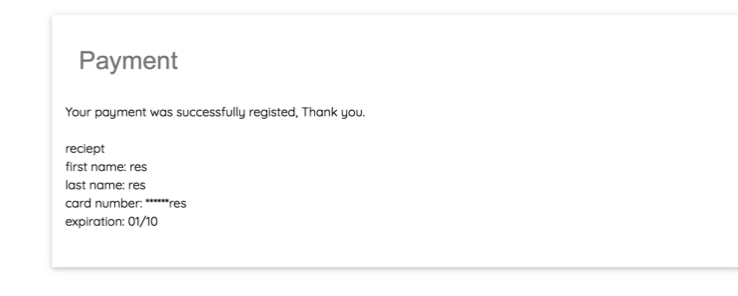
**Once you hit submit you will be directed to a payment page.**

**Payment Page**

User will then submit the payment for their completed form application by entering

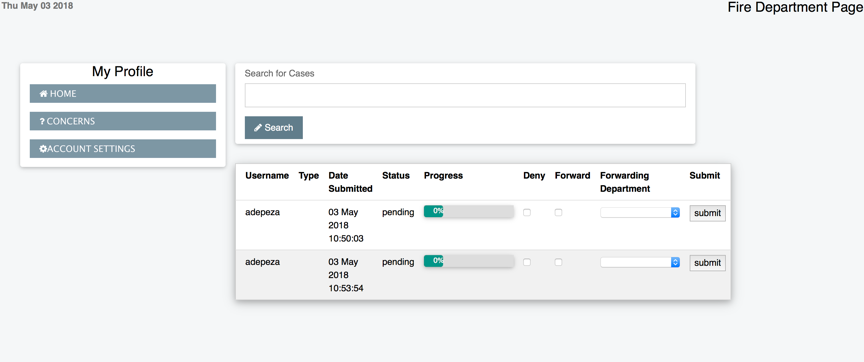
* First name
* Last name
* Card Information
* Billing Information

****

****

**This page will indicate whether your payment went through or not**

**Backend User Account Page**

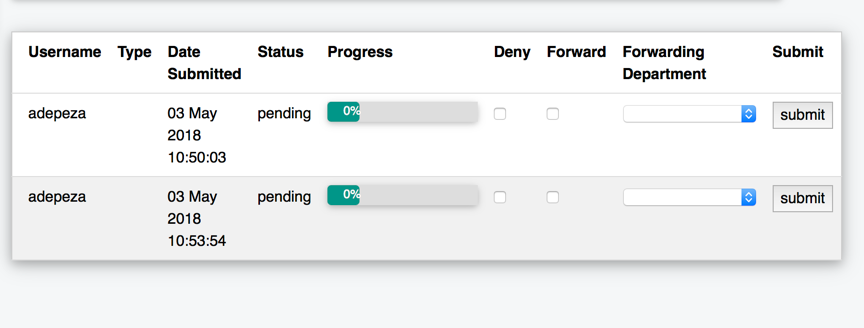
****

* Search: Allows user to look for any case or concern within their respective table



* Dashboard- Allows user to see all activity such as open cases that still need to be reviewed. This part will have the necessary information pertaining to the department.
* My Concerns – Allows user to submit a concern, comment or question
* Account settings – Editing account information, you will have the options of changing your personal details on your account or changing your password. If you click change your password, you must enter your old password on the box specified and enter your new password on the next text box along with confirming it in the third box. If you click on edit personal information, just enter into the box that you want to change your information and hit submit.

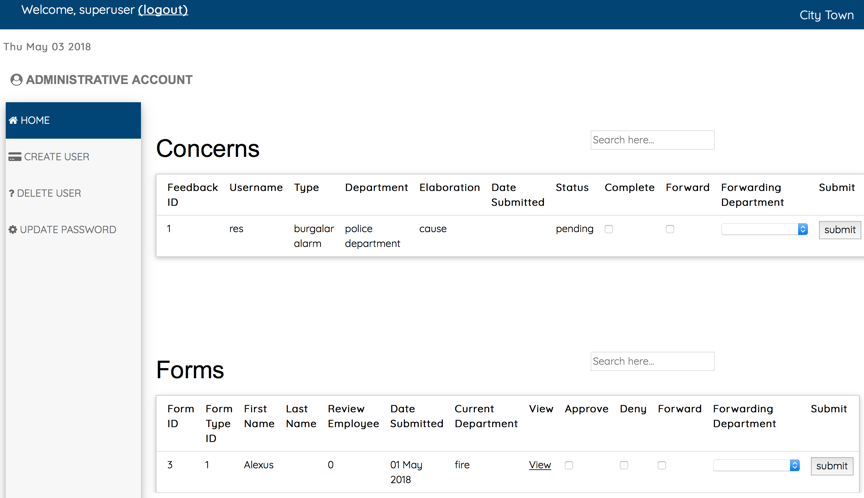
**Backend User Dashboard**

****

**The employee of this department is allowed to view the cases in their dashboard and forward or deny them.**

* Username: Shows the unique username of the applicant
* Type:
* Date Submitted: Shows the date and time their form was submitted by the applicant
* Status: Shows the current status of the form
* Progress: Shows how much the form is complete from review. The form has to go through all the appropriate departments before it reached back to the applicant on their approval of denial status
* Deny: Allows employee to deny applicant if they don’t meet department standards and send form back to applicant.
* Forward: Allows you to go to send the case to the next department
* Forwarding Department: Allows you to choose which department to send to next
* Submit: The final step for approving (to send to next department) or denying (to send back to applicant)

**Super User**

****

**Super user will have the ability to view every form and concern from every department and applicant.**

* Create User: Allows the Super User to add a new employee
* Delete User: Allows the Super User to delete a user
* Update Password: Allows Superuser to change any password

**Concerns:**

* Feedback Id: Shows the unique case ID
* Username: The username of either the employee or the applicant
* Type: Shows the type of for
* Department:
* Elaboration: Further details concerning the submission
* Date Submitted: The date the Concern was submitted
* Status: Shows if the form still need to be worked on
* Complete/Forward

**Forms:**

* Form ID: Shows the unique form ID
* Form Type ID: Each form has a unique form ID
* First name: Shows applicants first name
* Last name: shows applicants last name
* Review Employee: Current employee reviewing the form
* Date Submitted: The date the form was submitted by the applicant
* Current Department: This shows which department is currently reviewing the form
* View: This allows you to review the form that was submitted by the applicant to see if it meets standards
* Approve/Deny: Allows you to either accept or reject the form that was submitted
* Forwarding Department: Allows you to send it back to another department if necessary

**Planning for Version 2.0**

**INTEGRATION**

For the future of this project we are hoping the new version can be integrated with the already existing town website. This way this system can be used for any form on available. The dashboard pages may be able to integrate with the existing employee dashboard.

**USER- FRONTEND AND BACKEND**

* User should be able to have a PDF document of their filled out forms so they are able to print and store their files.
* User will be able to receive updated information about their account or submitted forms via email.
* Concerns submission should be determined in backend which department it should go to.
* Payment submission should be attached to appropriate credit card software
* Adding a payment history page for General User so they are able to keep track of all their purchases.
* Adding saved information so when a returning user fills out another form the majority of their information is pre-saved.
* Email notifications if the form needs to be filled out annually to General User when their form is close to expiration.

Progress Report 1 **date: \_\_\_1/22/18\_\_\_\_\_\_\_\_\_\_**

**Student Names:\_\_Kyle Desrosiers, Alexus Depeza, Daniel Gjuraj, Jonathon Nosowitz\_\_\_\_\_\_\_\_\_\_**

**Reporting Period: from \_\_\_\_1/10/18\_\_\_\_\_\_\_\_\_\_\_\_ to \_\_\_\_\_\_\_\_1/26/18\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |
| --- |
| **Summary:**  The user database, home, about, contact, and register pages are set up and all but the login page function properly; the html and php need to be edited. The register page was tested to ensure that no duplicates will be allowed in the database. The passwords in the register page were encrypted. Front end user page was created with dashboard and user account options. Backend User homepage was created with options for viewing account information and forms. Payment history page was created and connected to front end user page. The timeline for the project has changed. Issues with php have pushed back the expected development time; however, the implementation of security has reduced the overall set back. There have not been any changes to the architectural design. The library class being used is no longer FPDF, it is now bootstrap for programming purposes. |

|  |
| --- |
| **Activity Report:** |
| **Planned activities achieved** |
| *Summarize the following:*   * *List of modules developed during the reported period*   + user database, home, contact, register, about, frontend and backend pages * *List of test cases, results, and errors*   + When user fills out registration information and presses submit information goes to database   + Make sure password was hashed in database   + Error: No notification for duplications * *List of system functionalities tested*   + registration   + database   + security for created pages and database * *Major problems encounters and actions taken*   + registration php conditions not functioning properly -> restructured registration pages   + aesthetic problems -> sought help * *Progress on documentation*   + requirements document   + use cases   + modelling of different aspects of the system   + testing strategy |
| **Responsibilities and contributions of each team member (replace member # with the name)** |
| Alexus Depeza: Created front end user page  Kyle Desrosiers: added security to database and registration pages, began testing  Daniel Gjuraj: created back end user pages  Jonathon Nosowitz: created user database and home, contact, about html and register html and php pages |
| **Planned activities unfinished** |
| Alexus Depeza: New Form page, connecting front end to back end  Kyle Desrosiers: security for unfinished pages, connecting front end to back end  Daniel Gjuraj: connecting front end to back  Jonathon Nosowitz: login pages, connecting front end to back end |
| **Planned activities for the next period** |
| Alexus Depeza: Forms page  Kyle Desrosiers: security for login, negative testing  Daniel Gjuraj: begin testing, code verification  Jonathon Nosowitz: login pages, concerns page |

Progress Report 2 **date: \_1/7/2018\_\_\_\_\_\_\_\_\_\_\_**

**Student Names:\_\_Kyle Desrosiers, Alexus Depeza, Daniel Gjuraj, Jonathon Nosowitz\_\_\_\_\_\_\_\_\_\_**

**Reporting Period: from \_\_\_\_\_1/27/2018\_\_\_\_\_\_\_\_\_\_ to \_\_\_\_\_\_\_\_2/9/2018\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |
| --- |
| **Summary:**  The login pages html and php were created and completed. This includes forgot username/password options. The sessions functionality was added to the login pages. This ensures that the user will be logged out after a certain amount of time of inactivity for security purposes. The login pages were tested using negative testing to ensure that the user was logged in with only the proper information. However, the login pages still need to be sanitized. Logout functionality was also added. The layouts for the front and back ends were created (html) and the php was started, but not completed. The form page was started using Bootstrap. Thus far, the completed pages behave as intended. Not being able to send an email for username/password recovery was the only major problem encountered for this reporting period. In order to deal with it, the method of recovery was changed. |

|  |
| --- |
| **Activity Report:** |
| **Planned activities achieved** |
| *Summarize the following:*   * *List of modules developed during the reported period*   + login pages including forgot username/password and sessions functionality, logout functionality, and the forms page * *List of test cases, results, and errors*   + When user fills out login information and presses submit information is verified with information in the database     - transfers user to appropriate dashboard   + user fills out information for forgot username/password and information is verified with information in database     - transfers user to login page   + errors: incorrect information   + errors: Information for forms page not storing in database * *List of system functionalities tested*   + Security for login and logout   + Tested login to see it compared to database * *Major problems encounters and actions taken*   + not being able to send email for username/password recovery -> adjust notification method * *Progress on documentation*   + Cyber security fulfillment |
| **Responsibilities and contributions of each team member (replace member # with the name)** |
| Alexus DePeza: Created form page and redid it using Bootstrap  Kyle Desrosiers: Set up sessions for login, employee and resident pages. Added logout functionality. Negative tested the login page  Daniel Gjuraj: Continue testing pages and php for backend home page  Jonathon Nosowitz: created and finished login pages including forgot username/password and created html for concerns page and started php |
| **Planned activities unfinished** |
| Alexus DePeza: Forms page needs to be revised and connected to other pages  Kyle Desrosiers: Finished all tasks  Daniel Gjuraj: Finished all tasks  Jonathon Nosowitz: concerns page php |
| **Planned activities for the next period** |
| Alexus DePeza: Create Super user page, and pages for different departments  Kyle Desrosiers: Sanitization of all text fields, add requirements to password  Daniel Gjuraj: Work on creating payments and account settings pages  Jonathon Nosowitz: finish concerns page php, start connecting front and back ends |

Progress Report 3 **date: \_\_\_2/23/18\_\_\_\_\_\_\_\_\_\_**

**Student Names:\_\_Kyle Desrosiers, Alexus Depeza, Daniel Gjuraj, Jonathon Nosowitz\_\_\_\_\_\_\_\_\_\_**

**Reporting Period: from \_\_\_\_2/12/18\_\_\_\_\_\_\_\_\_\_\_\_ to \_\_\_\_\_\_\_\_2/23/18\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |
| --- |
| **Summary:**  The concerns page is completed (html, php) and the backend was started to be connected to the front end. The account settings page was completed for general information that can be used for frontend and backend as well as a page to change password which is linked to the account. Payment page has been made, but it’s php has not yet been completed. Functioning concerns page was created. Not all text fields were sanitized of special characters that could be used for SQL injection. The form page has been completed but needs to be connected to database and sent to different department pages. |

|  |
| --- |
| **Activity Report:** |
| **Planned activities achieved** |
| *Summarize the following:*   * *List of modules developed during the reported period*   + payment page and account settings pages with change password functionality * *List of test cases, results, and errors*   + The concerns page updates the database when a user submits a comment, concern or question     - error -> previous entry is overwritten before backend user is able to process   + payment information is entered and stored in database     - error -> information is not stored in database   + account settings changes/updates user information as well as password in the database     - error -> information is not updated * *List of system functionalities tested*   + security for login and logout   + SQL injection testing for login/logout * *Major problems encounters and actions taken*   + Login malfunction -> changing necessary code to ensure proper functionality * *Progress on documentation*   + started screenshots of GUI |
| **Responsibilities and contributions of each team member (replace member # with the name)** |
| Alexus Depeza: Completed Form page and created pages for each department  Kyle Desrosiers: Sanitization of all text fields  Daniel Gjuraj: Complete account settings page and payment page  Jonathon Nosowitz: Concerns page and started to connect front and back ends |
| **Planned activities unfinished** |
| Alexus Depeza: Super User page needs to be finished  Kyle Desrosiers: New pages have not been sanitized  Daniel Gjuraj: php for payment page and an added page to allow editing of most profile information  Jonathon Nosowitz: connecting front and back ends |
| **Planned activities for the next period** |
| Alexus Depeza: Sending form page to different departments pages  Kyle Desrosiers: Finish sanitation, and password restrictions  Daniel Gjuraj: test account settings page, complete php for payment page and change information page and a general test for all pages.  Jonathon Nosowitz: Connecting front and back ends |

Progress Report 4 **date: \_\_\_3/9/18\_\_\_\_\_\_\_\_\_\_**

**Student Names:\_\_Kyle Desrosiers, Alexus Depeza, Daniel Gjuraj, Jonathon Nosowitz\_\_\_\_\_\_\_\_\_\_**

**Reporting Period: from \_\_\_\_2/24/18\_\_\_\_\_\_\_ to \_\_\_\_\_\_\_\_3/9/18\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |
| --- |
| **Summary:**  The front end was connected to the backend. The layout for the superuser home page was created along with links to pages that will be used to view all forms, organized by department and edit the accounts of residents or town employees once they are functional. Department pages were updated with appropriate functionalities. Form page having trouble connecting to backend.Finished sanitation of remaining pages from previous iteration. Added password restrictions (ex. Capital letter, special character etc) to the register and change password pages. |

|  |
| --- |
| **Activity Report:** |
| **Planned activities achieved** |
| *Summarize the following:*   * *List of modules developed during the reported period*   + notifications for when the front-end user submits concern, question or comment   + Super user home page along with layout for its related pages * *List of test cases, results, and errors*   + Form page to database     - error -> information not storing in database   + filling all information on form page     - error -> shouldn’t allow you to go to payment page unless all information is filled in * *List of system functionalities tested*   + notifications when new concern is submitted   + form page updating information in database   + department pages functionality   + sanitation and password security * *Major problems encounters and actions taken*   + form page not connecting to database -> reviewed php code to find where error was occurring * *Progress on documentation*   + continued taking screenshots and updated test cases |
| **Responsibilities and contributions of each team member (replace member # with the name)** |
| Alexus Depeza: Worked on connecting form page, updated department pages, made client suggested improvements  Kyle Desrosiers: Finished sanitation of pages, password restrictions  Daniel Gjuraj: Worked on payments and change password pages and created super-user pages  Jonathon Nosowitz: front-end connected to back-end (notifications) |
| **Planned activities unfinished** |
| Alexus Depeza: Connecting Form page and rest of pages to database  Kyle Desrosiers: None  Daniel Gjuraj: php for payments page is still not in working order.  Jonathon Nosowitz: none, all planned activities were completed |
| **Planned activities for the next period** |
| Alexus Depeza: Connecting Form page and rest of pages to database  Kyle Desrosiers: encryption for sensitive information of payment page  Daniel Gjuraj: php for payment page, change password page, design and php of edit personal information page, and work on php for super-user form page  Jonathon Nosowitz: concerns/questions/comments forwarding, updating concerns page so that user can pick where to send information, updating login system to remove resident/employee option |

Progress Report 5 **date: \_\_\_3/23/18\_\_\_**

**Student Names:\_\_Kyle Desrosiers, Alexus Depeza, Daniel Gjuraj, Jonathon Nosowitz\_\_\_\_\_\_\_\_\_\_**

**Reporting Period: from \_\_\_3/10/18\_\_\_\_\_\_ to \_\_\_\_\_\_\_3/23/18\_\_\_\_\_\_**

|  |
| --- |
| **Summary:**  Finished edit account page. Payment submissions are updated to the database. Change password page is functional. The login and registration pages were restructured to remove employee/resident options. This required the structure of the database to be modified. The sessions were updated to maintain user information throughout the system as long as the user remains logged in. The associated pages were also updated to make sure that sessions worked properly. Form and department pages are connected database. Added encryption to the sensitive credit card information on the payment page. Edited registration pages to give visual cues for users when they create a password so they can see what they are missing out of the restrictions. |

|  |
| --- |
| **Activity Report:** |
| **Planned activities achieved** |
| *Summarize the following:*   * *List of modules developed during the reported period*   + Edit account → change password   + submit payments to database to be processed   + visual cues to ensure password is sufficient * *List of test cases, results, and errors*   + Login Page→ Redirected to to appropriate page. Validates user input   + Form Page → User input gets sent to database.   + Change Password Page → Password gets updated in database   + Payment Page → Information is sent to database and encrypted * *List of system functionalities tested*   + restructured login, edit account page → change password, payment submissions after filling out form, session identification information on each page(i.e. username), form submission sends information to database, login sends user to appropriate page and password cues inform user when password is insufficient * *Major problems encounters and actions taken*   + no major problems were encountered this period * *Progress on documentation*   + Ethical considerations, and started user’s guide and planning for version 2.0 |
| **Responsibilities and contributions of each team member (replace member # with the name)** |
| Alexus Depeza: Connected form page and department pages to database  Kyle Desrosiers: Encryption of credit card information and edited register pages  Daniel Gjuraj: Worked on php for edit account information page. finished payment page php so that information is sent to the database. Finished change password page.  Jonathon Nosowitz: restructured login/ registration pages, updated database and pages for sessions |
| **Planned activities unfinished** |
| Alexus Depeza: none  Kyle Desrosiers: none  Daniel Gjuraj: none  Jonathon Nosowitz: form/comments/questions/concerns forwarding |
| **Planned activities for the next period** |
| Alexus Depeza: Input validation for form and dept pages.  Kyle Desrosiers: Work on super user page  Daniel Gjuraj: Create a view payments page, work on super user pages.  Jonathon Nosowitz: form/comments/questions/concerns forwarding |

Progress Report 6 **date: \_\_4/6/18\_\_\_\_\_\_\_\_\_\_\_**

**Student Names:\_\_Kyle Desrosiers, Alexus Depeza, Daniel Gjuraj, Jonathon Nosowitz\_\_\_\_\_\_\_\_\_\_**

**Reporting Period: from \_\_\_\_\_3/24/18\_\_\_\_\_\_\_\_\_\_\_ to \_\_\_\_\_\_\_\_4/6/18\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |
| --- |
| **Summary:**  The form and concern forwarding functionality was completed. The database was modified to support this. The super user pages were created which include create user, update users password, and delete user capabilities. Also includes a list of all users that have accounts. Debugged the system by making sure every action went to the correct page. Made view payment page so that once submitted a payment is submitted it can be seen in the database. Form page now also redirects the user directly to payments when done. Created a page for town users to view forms submitted to them. Connected form/employee backend pages to database and implemented it to the rest of the web pages. |

|  |
| --- |
| **Activity Report:** |
| **Planned activities achieved** |
| *Summarize the following:*   * *List of modules developed during the reported period*   + form/comments/questions/concerns forwarding   + case history   + view payments page   + super user pages * *List of test cases, results, and errors*   view payment page → lists past payments the user as made from database  form page → redirects right to payment page once form is submitted  case history→displays current and old cases, contains option to view and approve new ones   * *List of system functionalities tested*   form submission notification, form page redirect, case history display, form approval and  denial, super user create user, update password and deletes accounts, forwarding concerns and  forms created to the required departments .   * *Major problems encounters and actions taken*   none. |
| **Responsibilities and contributions of each team member (replace member # with the name)** |
| Alexus Depeza: form php/ frontend/ backend php pages  Kyle Desrosiers: Created super user/Debugging of system/testing  Daniel Gjuraj: view payment pages, cases and case history pages  Jonathon Nosowitz: form/comments/questions/concerns forwarding |
| **Planned activities unfinished** |
| Alexus Depeza: none  Kyle Desrosiers: Finish debugging/testing  Daniel Gjuraj: none  Jonathon Nosowitz: none |
| **Planned activities for the next period** |
| Alexus Depeza: Fixing dashboard to meet clients expectations  Kyle Desrosiers: Work on debugging/testing and sanitize new pages  Daniel Gjuraj: adapting code to new dashboard design  Jonathon Nosowitz: none |