# Batalog

Project Team: Hadi Soufi, Thien Le, Kevin Keomalaythong, David Massey, Rahim Iqbal

## TABLE OF CONTENTS

## 1. **Project Definition** – *Thien L., Hadi S.*

Bat scientists at UNCG record bat calls from all over the state. Doing research on these calls can require years of manual work. Batalog seeks to do that automatically in a web interface using deep learning techniques to catalog bat calls. We aim to catalog bat calls according to their purpose, specifically whether they are echolocation calls or abnormal calls. The calls are stored in Zero Crossing format; the data will have to be cleaned up as it contains a significant amount of noise. Once the data is cleaned, the bat calls will be clustered according to their shapes and then catalogued for future scientific research. We would like to display the bat calls on a website and allow the user to interact with it. As a stretch goal, we will also be able to predict the nature of the calls based on metadata such as the time, location, and season that the calls were recorded in.

## 2. Project Requirements

### 2.1. Functional

- At least 90% accuracy (*Hadi*)
  - A convolutional neural network will be used to classify the type of bat call that is uploaded. The CNN should achieve a minimum of 90% accuracy during the training and testing phase prior to uploading any files.
  - The model must be evaluated to remedy overfitting. This can be done by checking training accuracy versus testing accuracy.
- Interactive user interface (*Thien*)
  - Display the data in a meaningful way: Allow the user to be able to see the sorted bat data in a collection of album. The user will be able to favorite and mark the data.
  - Allow the user to upload zero-crossing files for analysis: This function will give a front end design for the user to upload the file. While the zero-crossing files being processed in the background the user will get greet with a loading screen.
  - Allow the user to login and create an account

### 2.2. Usability

- User interface (*Hadi*)

        - Web-based
    - Performance (*Thien*)
        - The speed the website can process user data and output in a meaningful way. However this will be a low priority because we want to focus on the accuracy first.

### 2.3. System

- Hardware
    - Any Laptop and Desktop PC capable of running modern Operating System such as Mac OS X, Windows 10
    - Stable internet connection
- Software
    - Any modern browser (e.g. Chrome, Firefox, Opera, Safari). (*Hadi*)
- Database
    - The system shall allow uploaded images to be sent into the database, and facilitate communication between database and web interface
    - The database shall store PNG images in binary-large-object format and have a storage capacity of at least 50 GB

### 2.4. Security

- Username/password
    - Users will need to provide a username and password in order to login. More communication is needed with the Biology Dept if additional specifications are required.(*Hadi*)

## **3.** Project Specification

### 3.1. Area

- Machine Learning (*Hadi*)
- Web Development (*Thien*)
- Database Management (*Rahim*)

### 3.2. Libraries

- TensorFlow + Keras API (*Hadi*) - Python machine learning library/interface set that supports deep learning, i.e. facilitates building (convolutional) neural networks.

- Numpy (*David*) - Python library that introduces n-dimensional arrays including matrices, and facilitates linear algebra operations such as matrix multiplication.
- Matplotlib (*Kevin*) - Python 2D plotting library that allows visualization and statistical analysis of data.
- Dash (*Rahim*) - Dash is a framework based on python. It is used to build an analytical web application. Primarily it builds a dashboard using python.
- Pillow 5 (*Thien*) - A powerful image library for python web development that will help with organizing the image libraries.
- Django 2 (*Rahim*) - Django is a free and open source framework based on python. Django has less freedom but it is more efficient and secure than other frameworks such as asp.net or visual studio.

### 3.3. Framework

- Django 2 (*Thien, Rahim*)
- SQLite3 (*Thien*) - for database and the interaction between the website and the database storage.
- Python 3 (*Thien, Rahim*)

### 3.4. Development Environment

- Jupyter Notebook (*Hadi*) - Application used to write and execute python code and visualize output.
- Atom (*Thien*) - Text editor with built-in terminal that helps execute the codes.
- Sublime Text (*Rahim*) - Text editor that helps to read/write python and django code.

### 3.5. Platforms

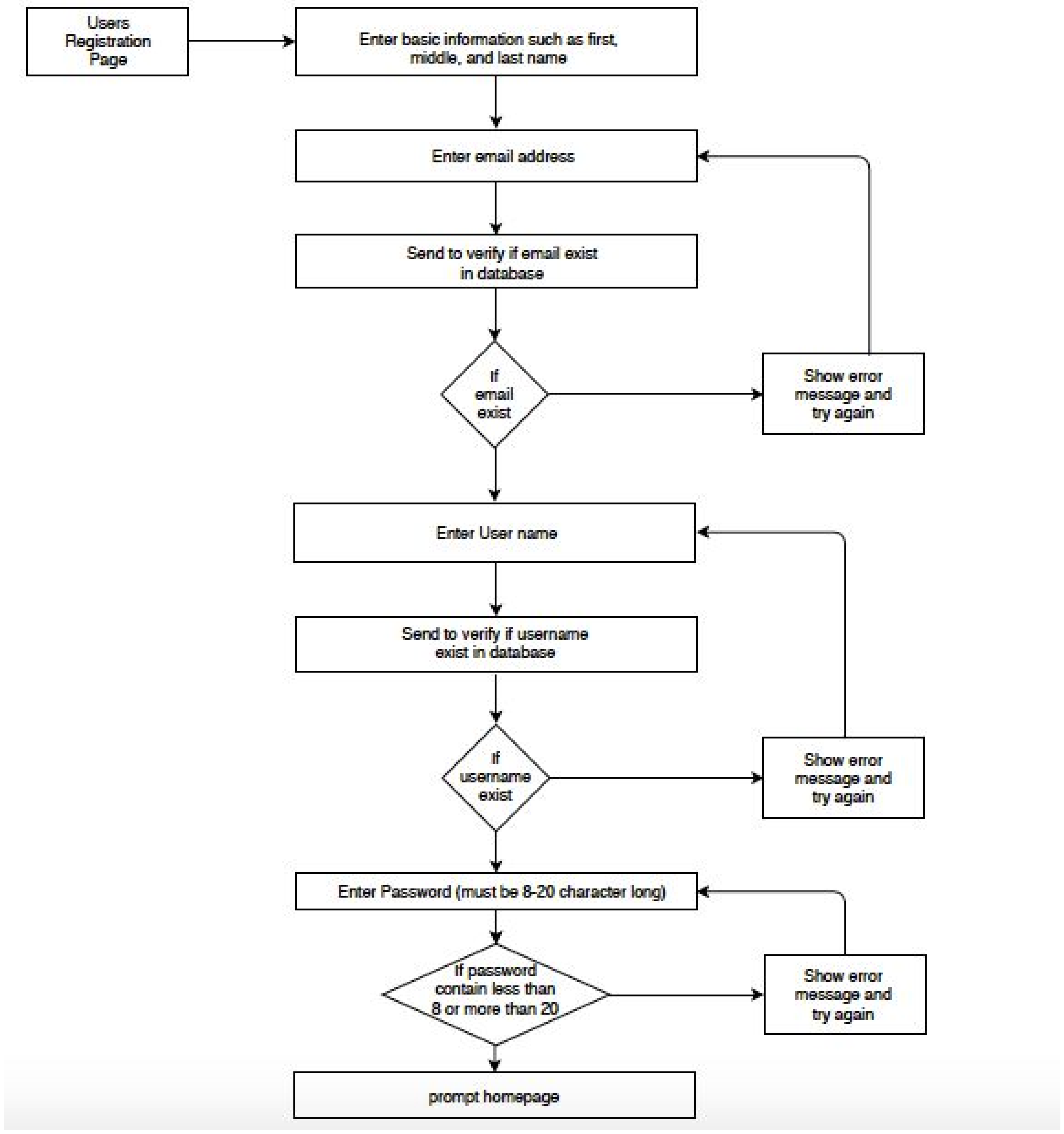
- Web (*David*)

### 3.6. Genre

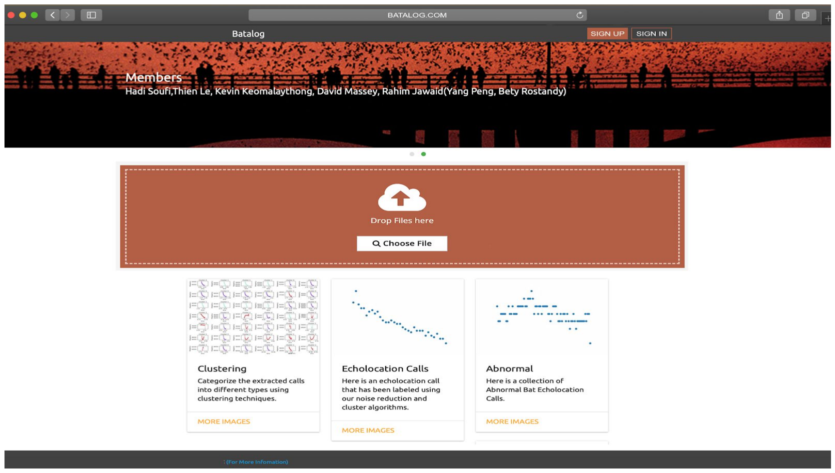
- Research tool (*Rahim*)

## 4. System – Design Perspective – *Group responsibility*

- Subsystems
    - Front-end
        - User registration page (Rahim)


Users Registration Page
Enter basic information such as first, middle, and last name
Enter email address
Send to verify if email exist in database
If email exist
Show error message and try again
Enter User name
Send to verify if username exist in database
If username exist
Show error message and try again
Enter Password (must be 8-20 character long)
If password contain less than 8 or more than 20
Show error message and try again
prompt homepage

- Zero-crossing file analysis (*Thien*)



BATALOG.COM

Batalog

SIGN UP SIGN IN

Members

Hadi Soufi,Thien Le, Kevin Keomalaythong, David Massey, Rahim Jawaid(Yang Peng, Bety Rostandy)

Drop Files here

Choose File

Clustering

Categorize the extracted calls into different types using clustering techniques.

MORE IMAGES

Echolocation Calls

Here is an echolocation call that has been labeled using our noise reduction and cluster algorithms.

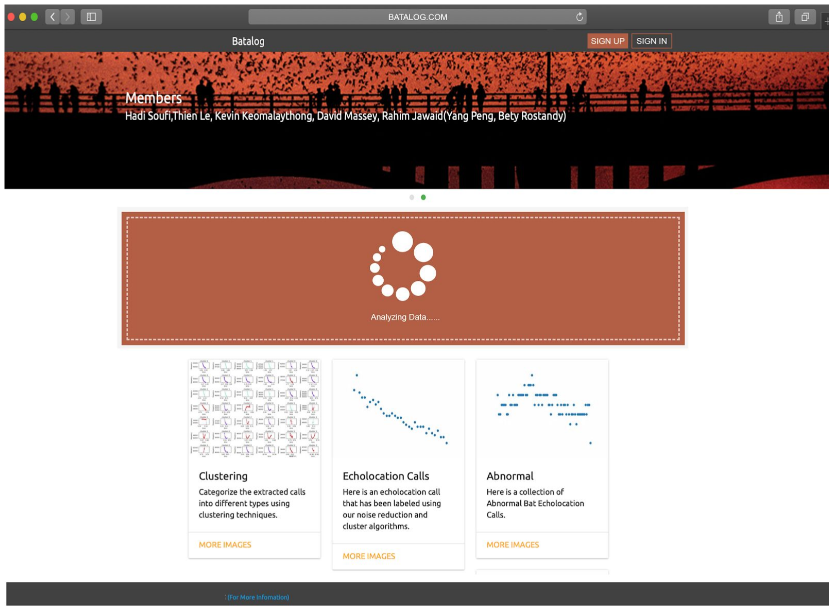
MORE IMAGES

Abnormal

Here is a collection of Abnormal Bat Echolocation Calls.

MORE IMAGES

(For More Information)



BATALOG.COM

Batalog

SIGN UP SIGN IN

Members

Hadi Soufi,Thien Le, Kevin Keomalaythong, David Massey, Rahim Jawaid(Yang Peng, Bety Rostandy)

Analyzing Data......

Clustering

Categorize the extracted calls into different types using clustering techniques.

MORE IMAGES

Echolocation Calls

Here is an echolocation call that has been labeled using our noise reduction and cluster algorithms.

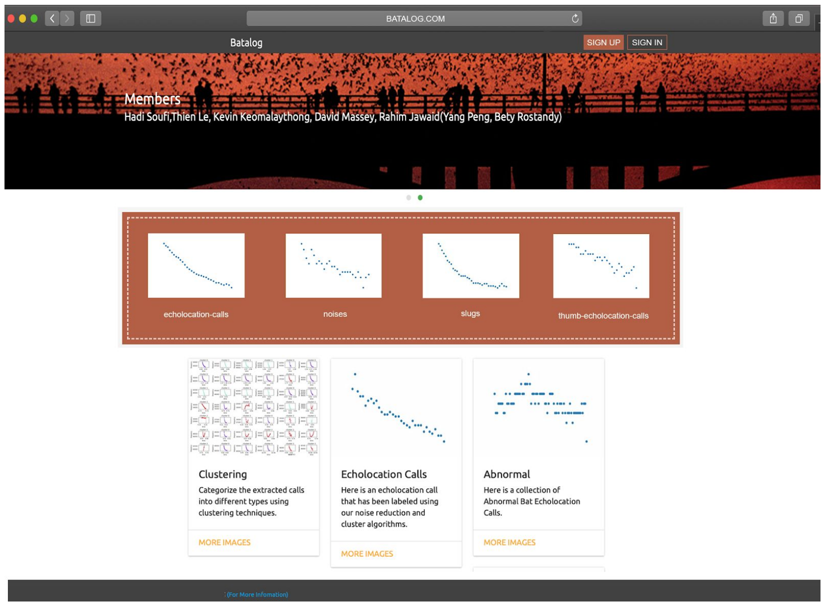
MORE IMAGES

Abnormal

Here is a collection of Abnormal Bat Echolocation Calls.

MORE IMAGES

(For More Information)



BATALOG.COM

Batalog

SIGN UP SIGN IN

Members

Hadi Soufi,Thien Le, Kevin Keomalaythong, David Massey, Rahim Jawaid(Yang Peng, Bety Rostandy)

echolocation-calls

noises

slugs

thumb-echolocation-calls

Clustering

Categorize the extracted calls into different types using clustering techniques.

MORE IMAGES

Echolocation Calls

Here is an echolocation call that has been labeled using our noise reduction and cluster algorithms.

MORE IMAGES

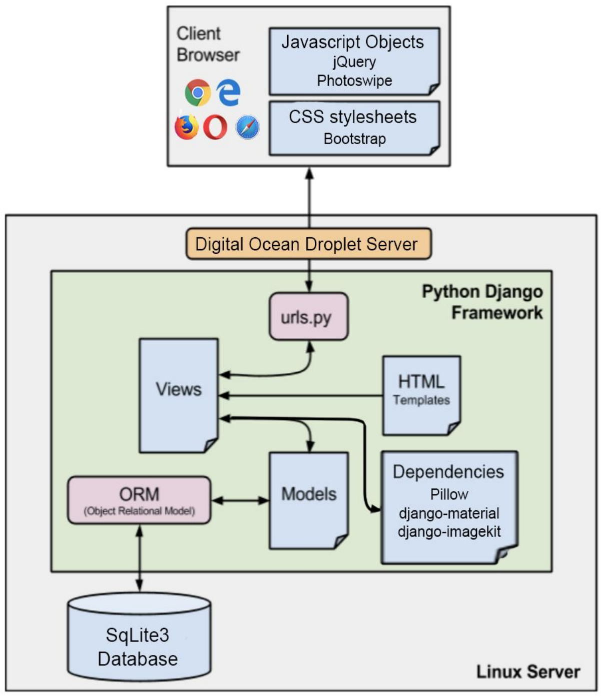
Abnormal

Here is a collection of Abnormal Bat Echolocation Calls.

MORE IMAGES

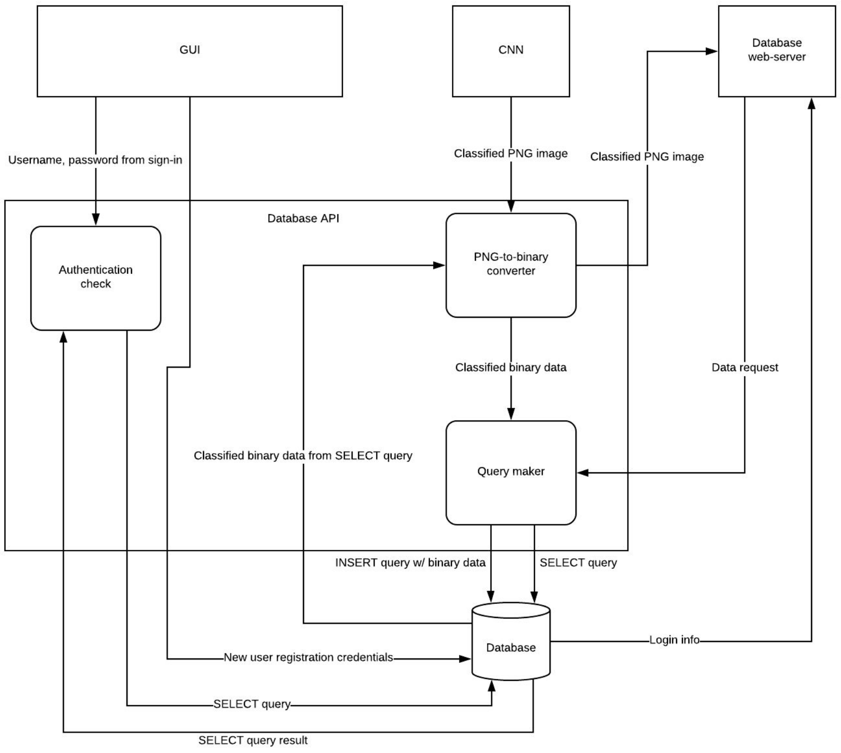
(For More Information)

- Back-end
  - Connection from web to Database (*Thien*)

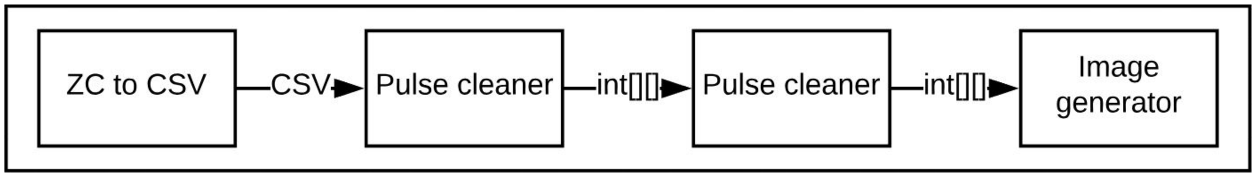


The main connection will be established from the digital ocean droplet server to the client browser.

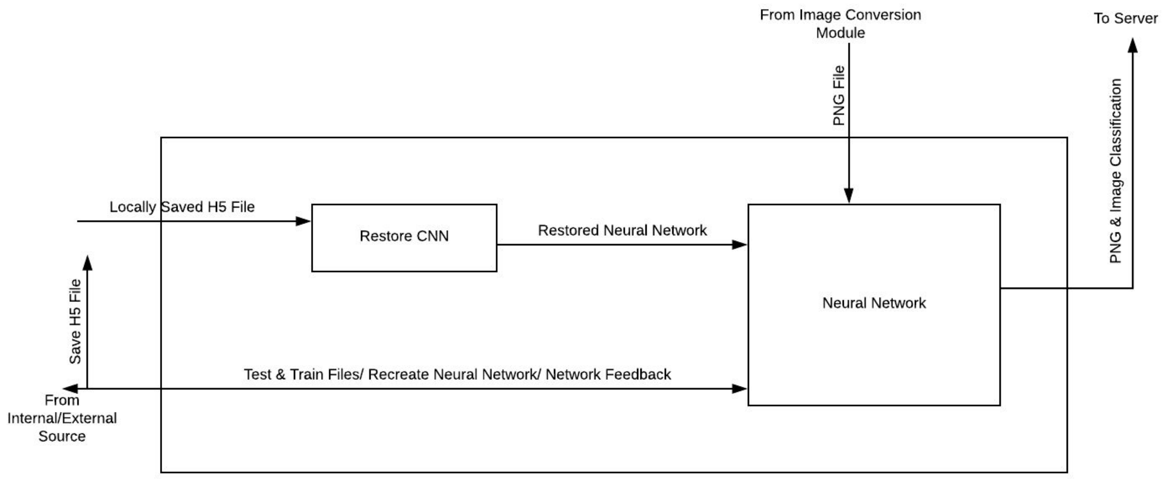
  - Database API (*Kevin*)



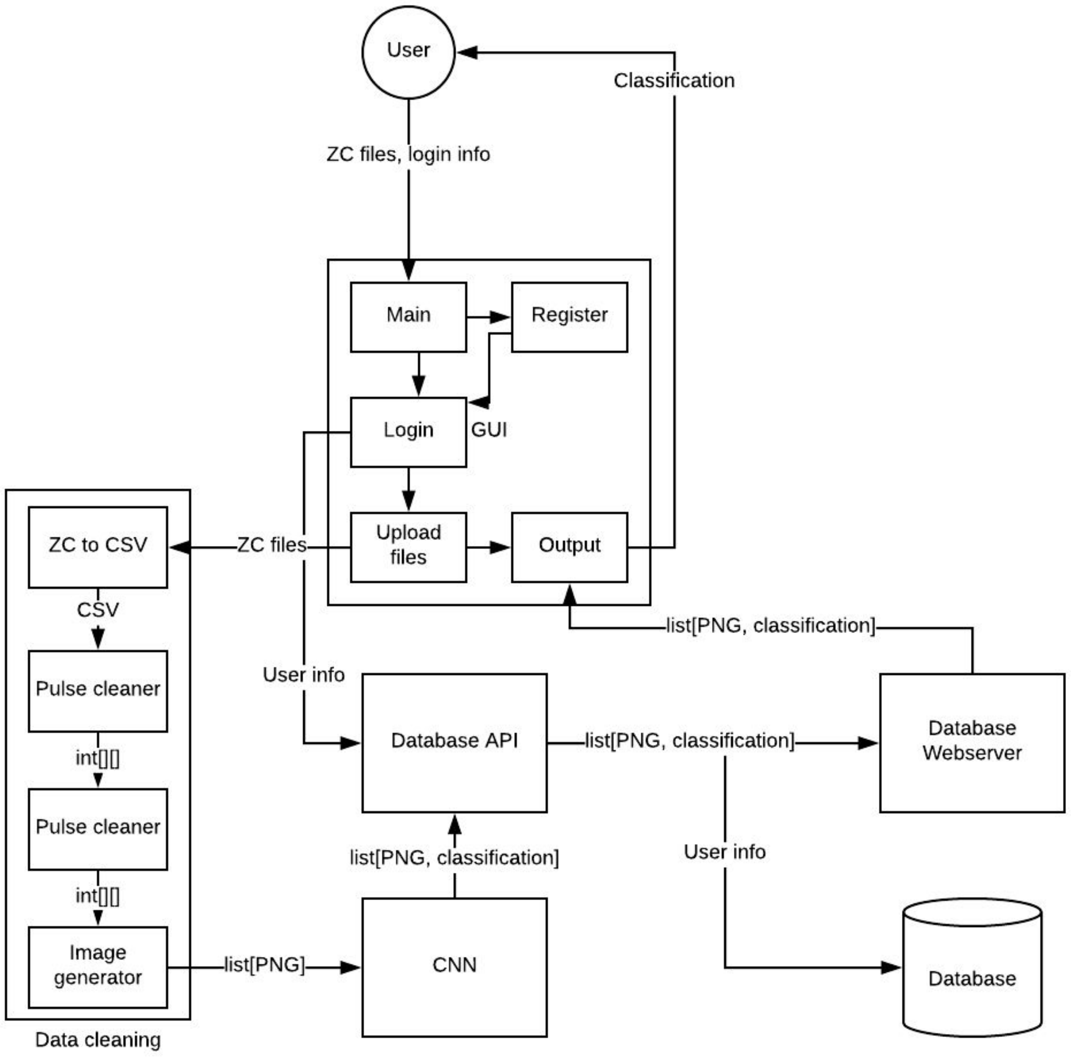
- Image conversion module



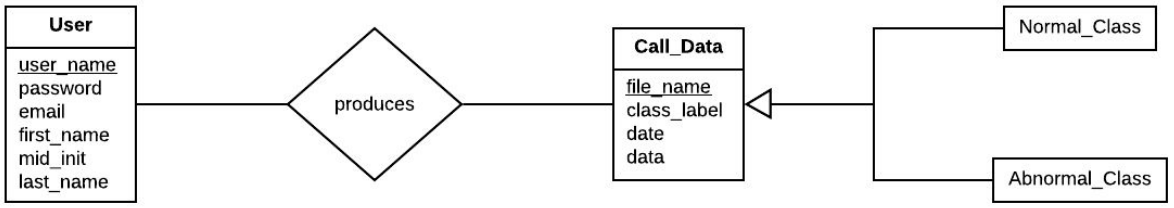
- CNN (*David*)


From Image Conversion Module
To Server
PNG File
PNG & Image Classification
Locally Saved H5 File
Restore CNN
Restored Neural Network
Neural Network
Save H5 File
Test & Train Files/ Recreate Neural Network/ Network Feedback
From Internal/External Source

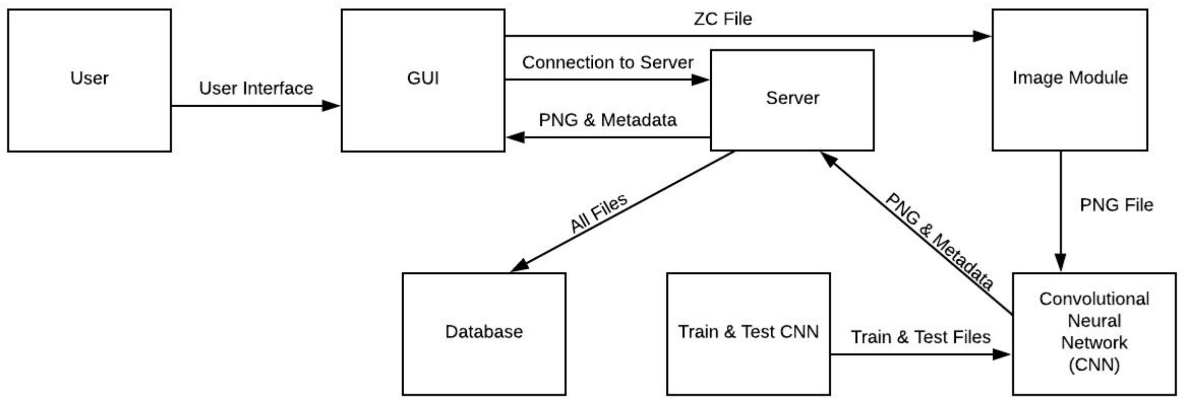
- Sub-System Communication
    - Controls (*Thien*), I/O, DataFlow (*Kevin*):



- Entity Relationship (E-R) Model (*Kevin*)



- Overall operation - System Model (*David*)



**5. System – Analysis Perspective** – *Group responsibility*

- Subsystems
    - Front-end
        - User registration page (*Rahim*) - Allows users to register on the website, so they can upload their ZC file. The users enter their basic information such as first, middle and last names. The register page also verifies the user email and username from the database. If username or email is present in the database, then it will display the error message and ask it to try again.
        - Zero-crossing file analysis (*Thien*) - Allows user to upload and see the file and classification.
            - If the user is logged in, they can upload a zc file or set of zc files.
            - While the file is being processed, the user will be greeted by a loading screen.
            - Once the files have been processed, they will get displayed in separate albums.
    - Back-end
        - Connection from web to Database (*Thien*)
            - Javascript Objects
                - jQuery - a library that helps navigate through documents, create animations and handle events easier.

                - Photoswipe - an image gallery for web development.
            - CSS stylesheets
                - Bootstrap - responsive grid system, extensive pre built components
            - Digital Ocean Droplet Server
                - A scalable compute platform with add-on storage, security, and monitoring capabilities to easily run production applications.
            - Python Django Framework
                - ORM - can be used to interact with application data from various relational databases such as SQLite, PostgreSQL and MySQL.
                - Dependencies
                    - Pillows - Python Imaging Library.
            - SqLite3 Database
                - An embedded database, server-less and can run within our app.
        - Database API - Handles GUI-database interactions and passes SQL queries to the database. The database stores PNG+metadata and user registration information.
        - Image conversion module - Converts ZC files to PNG images. The images will then be fed to the CNN module for classification.
        - CNN (convolutional neural network) module - Classifies bat pulses as normal or abnormal. Pulses are precleaned.

- System (Tables and Description)
    - Data analysis
        - Data dictionary (*Rahim*, *Kevin, David*)
            - Table: User
                - Data type: text (VARCHAR)
                - Description: This table will contain the user registration information such as user name, first name, middle name, last name, email, and password in the database. When the user enters their login information, the web application searches this table for the username and password and determines whether their

credentials exist in this table. If so, then the user will be able to login and upload their data.

- Table: Call_Data
    - Data type:
        - file_name: text (VARCHAR)
        - class_label: text (VARCHAR)
        - date: date-time
        - data: binary
    - Description: This table will contain a list of PNG images that represent the processed zero-crossing data. In SQL, the image data will be converted to binary large objects or BLOBs before being stored in the database. Each data entry will have an associated classification, either “normal” or “abnormal”, and the date and time it was collected.
- Variable: Model
    - Data type: Sequential object
    - Description: This variable will represent the CNN model used for training and predictions. The CNN is loaded and assigned to this variable prior to image classifications. It will include the network architecture, weights, and settings.

- Process models (*Rahim*)
    - The user inputs their basic information in the registration page subsystem. Once the registration process is finished, their information will be stored in the database, and they will be allowed to login to the website.
    - Once the user logs in, they will get an option to upload their ZC files. Once uploaded, the files are cleaned and converted to PNGs
    - When the file is converted into PNG, the CNN subsystem classifies the image as normal or abnormal. The PNG file plus metadata will be stored in the database.
    - Once the process completes, the database forwards the PNG and metadata into the GUI. Therefore the user will be able to see the output result.

- Algorithm Analysis
    - Entire system - $O(n)$ is the worst case of all subsystems.
    - CNN - $O(1)$: All images conform to the same pixel size. They will be processed one at a time. The network layout remains unchanged between images.
    - Registration page - $O(1)$: All the users follow the same steps for registering on the website
    - Zero-crossing file analysis - $O(n)$: Almost entirely array operations. An increase in file size corresponds to a linear increase in runtime.
    - Database API - $O(log\ n)$: The operations that the API pass to the DBMS are insert and search (SELECT query in SQL). Insertion is $O(1)$. Search is $O(log\ n)$ since each table will be indexed through B-trees.

**6. Project Scrum Report -** *Group Responsibility*

- Product Backlog (Table / Diagram)
- Sprint Backlog (Table / Diagram)
- Burndown Chart

**7. Subsystems**

**7.1 Subsystem 1** – Name 1 - *Individual responsibility*

- Initial design and model
    - Illustrate with class, use-case, UML, sequence ..... diagrams
    - Design choices
- Data dictionary
- If refined (changed over the course of project)
    - Reason for refinement (Pro versus Con)
    - Changes from initial model
    - Refined model analysis
    - Refined design (Diagram and Description)
- Scrum Backlog (Product and Sprint - Link to Section 6)
- Coding
    - Approach (Functional, OOP)
    - Language
- User training
    - Training / User manual (needed for final report)
- Testing

**7.2 Subsystem 2** – Name 2 - *Individual responsibility*

- Initial design and model
  - Illustrate with class, use-case, UML, sequence ..... diagrams
  - Design choices
- Data dictionary
- If refined (changed over the course of project)
  - Reason for refinement (Pro versus Con)
  - Changes from initial model
  - Refined model analysis
  - Refined design (Diagram and Description)
- Scrum Backlog (Product and Sprint -  Link to Section 6)
- Coding
  - Approach (Functional, OOP)
  - Language
- User training
  - Training / User manual (needed for final report)
- Testing

**7.3 Subsystem 3** – Name 3 - *Individual responsibility*

- Initial design and model
  - Illustrate with class, use-case, UML, sequence ..... diagrams
  - Design choices
- Data dictionary
- If refined (changed over the course of project)
  - Reason for refinement (Pro versus Con)
  - Changes from initial model
  - Refined model analysis
  - Refined design (Diagram and Description)
- Scrum Backlog (Product and Sprint -  Link to Section 6)
- Coding
  - Approach (Functional, OOP)
  - Language
- User training
  - Training / User manual (needed for final report)
- Testing

**7.4 Subsystem 4** – Name 4 - *Individual responsibility*

- Initial design and model

    - Illustrate with class, use-case, UML, sequence ..... diagrams
    - Design choices
- Data dictionary
- If refined (changed over the course of project)
    - Reason for refinement (Pro versus Con)
    - Changes from initial model
    - Refined model analysis
    - Refined design (Diagram and Description)
- Scrum Backlog (Product and Sprint -  Link to Section 6)
- Coding
    - Approach (Functional, OOP)
    - Language
- User training
    - Training / User manual (needed for final report)
- Testing

**8. Complete System** – *Group responsibility*

- Final software/hardware product
- Source code and user manual – screenshots as needed - Technical report
    - Github Link
- Evaluation by client and instructor
- Team Member Descriptions

***This is just a guide, and use it to create/improve your report. Feel free to add sections. You are responsible for your own subsystem/s, not other members. You have to contribute to the team's goals and objectives, and develop your subsystem/s, write your documents and slides.***