

Batalog

Project Team: Kevin Keomalaythong, Thien Le, David Massey,
Hadi Soufi, 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)
- Display the data in a meaningful way (Thien)
- Pretty, functional website (Hadi)
- Allow the user to upload zero-crossing files for analysis (Thien)
- Output the results of the analysis (Thien)

2.2. Usability

- Web based graphical user interface (Hadi)
- Performance - low priority (Hadi)

2.3. System

- Capable of running a modern browser
 - Stable internet connection (Hadi)
- Database - 50 GB of storage (Kevin)

2.4. Security

- Username/password to use tool. Need more communication with Biology Dept for other specifications (Hadi)

3. Project Specification

3.1. Area

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

3.2. Libraries

- TensorFlow + Keras API (Hadi)
- Numpy (David)
- Matplotlib (Kevin)
- Dash (Thien, Rahim)
- Pillow 5 (Thien, Rahim)
- Django 2 (Thien, Rahim)

3.3. Framework

- Django 2 (Thien, Rahim)
- SQLite3 (Thien, Rahim)
- Python 3 (Thien, Rahim)

3.4. Development Environment

- Jupyter Notebook (Hadi)
- Atom (Thien)
- Sublime Text (Rahim)

3.5. Platforms

- Web (David)

3.6. Genre

- Research tool (Rahim)

4. System – Design Perspective – *Group responsibility*

- Identify subsystems – design point of view
 - Illustrate with class, use-case, UML, sequence diagrams
 - Design choices (Optional)
- Sub-System Communication (Diagram and Description)
 - Controls
 - I/O
 - DataFlow
- Entity Relationship Model (E-R Model)
 - Example -
https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model
- Overall operation - System Model
 - Simplified Sub-system to System interaction

5. System – Analysis Perspective – *Group responsibility*

- Identify subsystems – analysis point of view
- System (Tables and Description)
 - Data analysis
 - Data dictionary (Table - Name, Data Type, Description)
 - Process models
- Algorithm Analysis
 - Big - O analysis of overall System and Subsystems

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.