Module 5- Computer Systems (2023-24)
Project

UNIVERSITY OF TWENTE.

Requirement Analysis Phase 1

Project Name: Bird Activity Detection	***Team Members:*** Lucas Fuertes, Thomas van der Boon, Dinh Thuy Nhat Vy, Carmen Asbreuk, Dimitri von Benckendorff, Duong Thu Huyen
Team ID: 9	***Mentor(s):*** Denis Krylov, Reneta Trifonova

Security Policy	Confidentiality, Integrity, and Availability					
Security Requirements	**Security mechanisms**	**Remarks on why you considered these requirements?**	**Supplement requirements for your application**	**Risk identification/Threat Assessment**	**Appropriate Security Controls**	**Tick ✓if you have applied the given security controls as suggested in the left column**
Authentication (the process of verifying who someone is)	- Password protection à checking passwords. Two factor authentication à likely to be an email.	Since we are creating a web interface for the user to view their 'findings', we wish to protect this. This is for multiple reasons such as a possible extension that connects the web to the cloud and any other sensitive information one can find on the media such as detailed facial features of passing pedestrians. We are adding email verification for extra	User Story: 1. As a user, I can enter my username and password into a login page to authorize my access to the web interface. 2. As a user, I can enter a password containing alphanumeric and special characters that have a length longer than 12 characters, to mitigate brute-force attacks. Abuse Case: As an attacker, I can gain easy access to the system	1. If the password is too short, e.g., 3 characters, it will allow easy access through code cracking algorithms. 2. If the password is a sequence or something easy to guess such as 1234… or "password". 3. If we allow for multiple password entries, code cracking algorithms can brute force a password. If we give no minimum requirements for	1. The password should be a minimum of 12 characters long to ensure a reasonably good password. 2. The password field should require special characters to ensure that: "password" or "1234…" cannot be a password. The password can be attempted 5 times before a temporary wait time is started, after this happens one more time, the account will be blocked, and an email is sent to the owner asking them to contact the support team.	

		protection, but we might make it toggleable to adapt it to the users' preferences.	by entering well known/easy passwords.	certain characters or limit certain characters and sequences, the field could be a risk to SQL injection and guessing the password.		
Authorization (the process of verifying what specific applications, files, and data a user has access to)	1. Role-based access control for the database	We want to enable accessing the database with three roles. We want read-only permissions for the output layer, write-only permissions for the input layer, and all permissions for the administration by project members.	User Stories: As an administrator, I can access, read, and write in the database. As an authorized client (program on raspberry pi or online web application), I can either read from or write to the database. Abuse cases: As an attacker, I can access, read, and write in the database through a client. As an attacker, I can modify the existing role-based access control rules.	Risk 1. Administrators having unrestricted access to the database may accidentally or intentionally misuse their privileges, leading to data corruption, leakage, or unauthorized modifications. Risk 2. Attackers may compromise an authorized client (e.g., by injecting malicious code or stealing authentication credentials) to gain unauthorized access to the database. - Risk 3. Attackers with the ability to modify access control rules can grant themselves or other unauthorized users elevated privileges,	Control 1. The administrator authentication is secret and is not shared in any program related to the project. The users with access to this role are educated on the proper use of the database. Control 2. The clients (program on raspberry pi and online web application) need to be secured against malicious activities from an attacker, such as the request to view data from the database. • Control 3. The role-based access control rules are set when developing the system and cannot be changed afterwards by users who are not authorized.	

				manipulate access policies, or disrupt the system's security model.		
Audit (a review to check its quality, progress, or adherence to plans, standards and regulations	1. Deleting old and unused user data	According to the European Union's General Data Protection Regulation (GDPR), users have a right to be forgotten. This means removing user data no longer in use or when consent is revoked.	Abuse case: As an attacker, once I breach the database security, I will have access to all historical user data since the launch of the system. User case: As a user I don't want the system to store my personal data after I'm done using the system.	If a data breach occurs, the attacker would have access to much more private user data. There is also a legal privacy concern since the system must comply with GDPR regulations. Non-compliance may result in fines.	We can use an event scheduler within the database to execute a script weekly. The script removes any leftover user data. Administrators should be able to launch this script specifically to remove user data.	
	2. Quality and accuracy check of data processing techniques	The data collected about the birds (such as images) will be processed using AI to classify them into species. This means that the accuracy of the classifications is a concern.	User stories: As an administrator, I want the birds to be classified accurately so that the correct conclusions and insights can be drawn from it. As a BirdWeather user, I want the bird sighting data to be accurate, as I might use the data for my own	It may be the case that our AI classification model becomes inaccurate due to poor inputs (images) or new training data. Before sharing the data, it is important to verify its quality, so that users can trust our system. Inaccurate data can also lead users to draw the wrong conclusions, which	Quality assurance after training: We want our AI model to score better than random and aim for a 90 percent accuracy on a labelled set of bird pictures.	

			purposes, such as research.	can be problematic in research.		
	3. Backup files	Unfortunately, machine or human error-related primary coding or data problems might occur during the course of a project. Therefore, it is essential to have a backup file to restore from if one of the events stated happens.	As an administrator, I want to make sure that my program can be recovered whenever, especially when my computer is stolen and broken.	If the laptop is broken, the significant products will be lost and potentially lead to various challenges and consequences. In certain cases, trying to fix an error only makes it worse, in which case you can go back to the desired code version.	After changing some pieces of code, we will cautiously update backup files as much as we can while restricting access to this file to only authorized staff.	
Reviewed by:	Dimitri von Benckendorff Carmen Asbreuk Thomas van der Boon Duong Thu Huyen Dinh Thuy Nhat Vy Luca Fuertes Taweeapiradeemanee					

4. Introduction

4.1. **Purpose**:

The main purpose of our system is to be used for hobbies or research purposes.

For hobby purposes:

- There is a small, but active bird watching community around the world. Bird watching takes a lot of free time to do, as even with the use of a bird feeder it can take hours to be able to snap the perfect picture of a bird, especially when you are looking for a particular species. In addition, it is not always easy to identify what species of bird you are looking at. Our system aims to address these issues for hobbyists, making bird watching more accessible and giving them new tools to observe and track birds in their area. Motion detection can automatically detect the presence of a bird at the feeder, meaning that birds can be tracked at any time of day (such as in the middle of the night). Our system will also use AI to identify the bird's species, making it easier for non-expert hobbyists to identify birds, and giving beginners a new way to learn about what kind of birds are in their area.

For research purposes:

- Researchers (especially ornithologists) may find it necessary to monitor the presence of birds in certain areas. Our system can be a way for them to collect this data automatically. By monitoring bird activity in certain areas, scientists can detect trends. This may especially be the case when multiple bird feeders are placed in different locations. Additionally, there exist third-party websites such as BirdWeather that have over 200 active stations around the world and is sharing their data online. Projects such as these are research projects that aim to monitor bird activity at scale, and our system can act as another way to contribute to these projects by acting as a station, providing data about birds in the area.

4.2. **Limitations of the current system (If any)**:

The current limitations of already existing bird activity detection systems are:

- Limited Camera Quality: The camera modules available may not have the same image quality or resolution as a dedicated camera, which can impact the system's ability to accurately detect and identify birds, especially in challenging conditions. Users who expect clear and high-quality images for their bird detection system may become frustrated by the inability to achieve their desired level of image quality. Moreover, low-quality images can make challenges for the system to correctly identify bird species.
- Real-time Processing: Achieving real-time processing for bird detection can be difficult on Raspberry Pi, especially for high-resolution video streams. This can limit the responsiveness of our system. That limitation can lead to delayed responses, missed events, and decreased user satisfaction.
- Limited Bird Species: The system's accuracy may vary for different bird species, and it may struggle to detect less common or smaller birds. And The challenge's bird's diversity to find enough bird specifies to test our system in Enschede centrum. Users who are interested in comprehensive birdwatching or ecological research may find this limitation restrictive as it might not provide reliable data for less prevalent.
- Maintenance and Updates: The Raspberry Pi system requires regular maintenance, software updates and potentially hardware upgrades to keep them functioning optimally over time. Users who lack the time or expertise for

ongoing system management may experience issues with stability and performance, making the system less reliable for bird detection tasks.

- Data Privacy: We plan to capture images in public spaces, so the problem is considering data privacy and legal implications. Users must navigate complex privacy laws and regulations, potentially limiting where and how they can deploy the system, especially in areas with strict privacy requirements.
- Memory Constraints: The system has limited RAM, which can restrict the size of data (pictures which need lots of storage space), which can restrict the size of data sets and the complexity of machine learning models that can be deployed. The limitation of RAM can significantly restrict the user's ability to work with extensive datasets and deploy sophisticated machine learning model.
- Environmental Conditions: Outdoor bird detection may be affected by various environmental factors such as lighting, weather conditions, and obstructions. These can make it challenging to maintain consistent assurance and the camera will not be broken by some rain or any animals.
- Limited Storage: Raspberry Pi devices often come with limited onboard storage. Storing large image datasets/ recording video may need to use external storage solutions which can add complexity and cost. The limited onboard storage in Raspberry Pi devices can be a significant limitation for users, as it necessitates the use of external storage solutions, introducing complexity and additional costs into their bird detection project, potentially making it less accessible and convenient for users seeking a straightforward and cost-effective solution.
- Power Consumption: In remote or battery-powered setups, power consumption can be a significant concern. Raspbbery Pi can consume a noticeable amount of power, potentially limiting the duration of your bird detection system. The power consumption of Raspberry Pi in remote or battery-powered setups is a notable concern for users, as it can significantly restrict the operational duration of their bird detection system, potentially reducing its effectiveness and usability in off-grid or resource-constrained environments.

4.3. **Intended Audience**

The intended audience for this project encompasses a diverse range of individuals, with a common interest in avian life. This application is primarily designed for bird enthusiasts and researchers, as its features facilitate the systematic categorization and analysis of bird species within specific geographical locations.

Additionally, our project has stakeholders, including our mentors, project and module coordinators, and our development team. While each stakeholder may have distinct expectations for the final product, our collective aspiration is to deliver a seamlessly functioning and user-friendly tool. This tool will enable all of us to explore, document, and conduct research on the bird population in our respective areas of interest.

Furthermore, the Bird Weather website stands to benefit from our efforts as well. Our work will enable them to expand their network of bird sightings to include the city of Enschede.

4.4. **Define SMART Goals**:

Requirement	Specific	Measurable	Achievable	Relevant	Time-bound
Detect motion	Detect an object's movement near the bird feeder	Within 1 meter from the camera within the sensors angle range	Set up a camera within 30 cm of the bird feeder aimed at the bird feeder	Sends a signal to the camera	Week 1 in Sprint 3
Detect bird landing	Detect an object when it increases the weight on the knock sensor	When the weight on the knock sensor increases with minimal of 20 grams	Set up a knock sensor attached to the feeding platform	Sends a signal to the camera	Week 2 in Sprint 3
Capture pictures	When a signal comes in, capture a picture of the object	At least one picture should be captured when the 'movement' signal comes in	Set up a camera within 30 cm of the bird feeder aimed at the bird feeder	For analyzing and viewing of the picture	Week 1 of Sprint 3
Analyze if a bird is in the picture	Implement bird classification	Achieve at least 95% test accuracy	Utilize database and training a machine learning model	To verify that we are detecting bird activity	Week 1 Sprint 4
Classify birds using collected information	Implement bird species classification	Achieve at least 50% accuracy on test set and share score of new images	Utilize database and training a machine learning model	To verify that we are detecting activity from a certain bird species	Week 2 Sprint 4
Interactive web interface for user access	Create an interactive web interface	Ensure easy data access	Use web development best practices	Enhances usability	Week 1 Sprint 4
Visualize data and discern patterns (Statistics)	Display basic statistics in a dashboard	Provide clear visualizations	Use data visualization tools and libraries	Enables data analysis	Week 2 Sprint 4
Allow users to sort and filter catalogued birds	Implement sorting and filtering features based on time, weather, and bird species	Efficiently sort and filter data	Design user-friendly tools	Enhances data access	Week 2 Sprint 4
Manage metadata	Store relevant time and weather data	Store timestamp and the weather data for	Create metadata tables in database	External variables are considered for	Week 1 Sprint 4

		every hour at which bird activity was detected	and integrate system with open-source time and weather data via APIs	users who aim to discern patterns in the data	
Archive unwanted images	Archive blurry images or images that do not display a bird	Manual verification	Build two layers into the database for staging and live. The difference between these image sets e the unwanted images.	Makes sure only relevant pictures are displayed and the database is not immediately filled with unwanted pictures.	Week 2 Sprint 4
Blur personally identifiable data for privacy	Implement automatic data blurring for humans	Manual verification	If time allows us to, use image and video processing techniques	Safeguards privacy	Week 2 Sprint 4
Share retrieved information	Automated data upload to BirdWeather or another bird sighting database	Manual verification to see if uploads were successful at least once	If time allow us to, establish integration with BirdWeather API or another API	Share sightings for hobby and research purposes	Week 2 Sprint 4
Authentication and secure data transfer	Implement authentication with username and password and secure transfer through encryption	Authentication needs to be futureproof and secure transfer of data using strong encryption	If time allows us to, we use strong encryption and secure protocols	Protects user data and system integrity	Week 2 Sprint 4

4.5. **Scope:**

- Technology used to develop project:
  - Hardware requirements: Raspberry Pi 4, a PI camera, KY-031 knock sensor, KY-032 obstacle detector, KY-050 ultrasonic distance sensor:
    - Raspberry Pi 4: serves as the central processing unit for the entire system. It runs the operating system and the software responsible for reading data from the sensors and processing it.
      - PI camera: is used for capturing images of bird activity.
      - KY-031 knock sensor: can be used to detect vibrations or sounds caused by bird activity such as pecking on a feeder or chirping. It typically outputs a digital signal (0 or 1) when a knock is detected. Connect the KY-031 sensor to one of the GPIO pins on the Raspberry Pi and write Python code to monitor the sensor's output and trigger actions when a knock is detected.
      - KY-032 obstacle detector is an infrared sensor that can detect the presence of objects, including birds, within its detection range. Connect the KY-032 sensor to another GPIO pin on the Raspberry Pi and use GPIO libraries to read the sensor's data. An interrupt or polling mechanism might be set up to check for changes in sensor values when bird is detected.
      - KY-050 ultrasonic distance sensor measures the distance between the sensor and a bird. It works by emitting ultrasonic pulses and measuring the time it takes for the pulses to bounce back. Connect the KY-050 sensor to GPIO pins on the Raspberry Pi, typically one for trigger and one for echo. Hence, Python code is required to send trigger signals, measure the echo time, and calculate the distance to the bird.
  - Software requirements: Python, HTML, CSS, JavaScript, MySQL (and possibly Java for the website's API).
    - Algorithms:
      - Hashing and salting to store passwords in the database.
      - Image capture: use python libraries (e.g., PICamera) to capture images from the camera module.
      - Image preprocessing: process images to enhance their quality, such as resizing, cropping, and adjusting brightness and contrast.
      - Object detection algorithm to detect and locate birds within the image. Option may include YOLO (You Only Look Once) algorithm.
      - Machine learning model: train a machine learning model to classify the detected birds into different species that are stored in the database. CNNs (Convolutional Neural Networks) may be used to classify images and can be trained to recognize bird species. ResNet-50 is one of the most efficient CNNs that we might choose since it tends to give better accuracy than MobileNet thanks to its greater number of parameters compared to those in MobileNet.
      - Deep learning framework: TensorFlow.
      - Data collection: gather a diverse dataset of bird images for training and testing model and store the data in the database.
- User interfaces: create a user-friendly interface for the Raspberry Pi, allowing users to capture images and receive bird classification results and its information for research in the future.
- Duration of project: 10 weeks with 4 main sprints.

5. **Product features:**

- **The functional requirements of an automated bird activity detector are:**

  - [**H**] Autonomously capture and store media (pictures, videos, audio) when a bird is detected by its external sensors.
  - [**H**] Classify birds using the collected information (such as species and sound)
  - [**M**] Interactive web interface that allows the user to access data relating to the catalogued birds (such as images, videos, and time).
  - [**M**] Visualize the data allow users to discern patterns -- Statistics (dashboard in web app)

- **The nonfunctional requirements of an automated bird activity detector are:**

  - [**M**] Allow the user to sort and filter through the catalogued birds.
  - [**M**] Manage the metadata and catalogued birds, by archiving unwanted images. -
  - [**L**] Blur personally identifiable data for privacy of individuals captured on the media.
  - [**L**] Upload collected data to BirdWeather automatically.

- **The security requirements of an automated bird activity detector are:**

  - [**M**] An authentication layer (with username and password) for the web interface and database to store media in line with privacy.
  - [**M**] Secure transfer and storage of the media in the database. This includes protection measures for injection attacks and cross-site scripting.

6. **Conclusion:**

Now that we have presented our project concept and documented the relevant criteria, we would like to emphasize certain decisions made in the design process and anticipate some obstacles for the upcoming phases. Firstly, we intend to create various components that demand expertise spanning multiple areas within the realm of computer science. Consequently, our team must acquire knowledge in both embedded systems design and development, as well as machine learning modeling. These represent challenges we will strategize for in the forthcoming phases.

The main security measure for the web interface to consider is the user login. We will likely be using JWT tokens to authenticate the user's browser by storing the token as a cookie upon the user's successful login. We also need to make sure that the tokens expire after a set period. Of course, we will also need to securely store usernames and passwords in our database. For this, we will also have to take measures by hashing and salting the passwords. Due to the web interface, as well as a need to upload images to the database, we will need to build an API. This can be done through RESTful services using Java for example, but we still need to establish what software we will be using for this. Finally, role-based authentication is necessary for the API to be secure.

Since the bird feeder will be outdoors, we will also have to consider environmental factors when creating the design. The hardware should be protected from the elements, which will include the wind, damp, and rain.

The main way we wish to overcome this challenge is to keep all the hardware within the house, this would mean that we would need sensors that can pass through glass and see outside without being outside. Of course, this means that the window must be clean to ensure that the birds will be visible when they are eating. We are hoping this can be overcome using an open birdhouse that would cover a “large” area of the window from the elements such as rain and snow, and window cleaning to ensure the camera can get the best picture possible.

A back-up plan would include waterproofing the hardware, for a camera, this can be done by using a GoPro that would be waterproof and encompassing the feeder with a type of plastic wrap that could keep a knock sensor out of the rain. Currently there is no plan of protection against the cold or a damp environment that would come because of the rain protection, however we will continue to work on a plan regarding that.

Another significant challenge pertains to the necessity of hardware for the system's input. Establishing the linkage to the system and executing the integration of sensors can be intricate tasks, thus presenting considerable hurdles. Another comparable challenge lies in the interconnection of the various components within our system.

7. **Reference**:

- https://www.birdweather.com/about
- https://projects.raspberrypi.org/en/projects/getting-started-with-picamera