Product Opportunity Document
Smart Insect Biomass Scale (SIBS)

Concept Authors:
Josefiina Falck, Liina Hilkamo, Heini Könönen, Meri Raelahti, Thomas Tallqvist, Tuomas Tapper

This document presents the concept of a Smart Insect Biomass Scale (SIBS). Equipped with weighing, temperature, humidity and movement sensors, SIBS provides increased data accuracy and a real time view on the life of vital night pollinators — moths. The concept is developed as part of the IDBM Master studies in the ELEC-E9900 — Networked partnering and product innovation – NEPPI course at Aalto University 2019.

Table of contents

1. Design concept

In the beginning of the project, our team agreed on a shared vision of not creating anything unnecessary. We wanted to design for a worthy cause, and potentially for a market where technology was yet to be utilised. We acknowledged the caution needed in designing IoT products to avoid disposable items that only increase the risk of accelerating environmental destruction (Light and Goodman, 2015).

During our research, we came across the concept of the ongoing sixth mass extinction which researchers also call ‘a biological annihilation’, which is ironically a result of humankind’s overconsumption and overuse of nature’s resources (Carrington, 2017). Studies show that up to 50% of all individual animals, including mammals, birds, reptiles and amphibians have been lost in the past 40 years (Carrington, 2017). Additionally, the sixth extinction is progressing faster than expected and the disappearance of any species could cause irreversible damage to the climate (Carrington, 2017). We describe our research methods and insights further in Chapter 2: Background research.

To preserve biodiversity, current research is focusing on tracking if new areas appear where species are lost, especially ones like pollinators from which our food, clean air and water are highly dependent on (Carrington, 2017). But according to our interview with Pasi Sihvonen, an insect researcher with extensive academic background, understanding the changes in insect populations is lacking in coverage (Sihvonen & Lilley, 2019). In Finland, moth research is highly dependent on hobbyists who build their own traps and report their contents, but the amount of traps has significantly declined from 152 traps in the year 1996 to 42 to 45 traps this decade (Leinonen et al., 2016), hence new solutions are in dire need. This motivated us to explore if we could design an IoT solution that would help in understanding the world better by exposing previously unseen patterns, and this is when we came up with the idea of the Smart Insect Biomass scale (SIBS) — a product that can either replace hobbyists’ moth traps or act as an extension to them.

SIBS generates real time data of the size and behavior of the moth population, and in combination with the dashboard, a web based application, researchers can at any time access the data provided by traps registered to the service. SIBS provides data on times the moth population is active and how many individuals are active combined to humidity and temperature data, enabling researchers to analyse how the environment impacts on moths by answering three research questions: how many moth individuals are on the move at what times in which locations, is the moth population increasing or decreasing and if and what kind of weather conditions impact moth activity. We explain the web based application concept further in Chapter 4: Progressive web app design.

SIBS improves both the accuracy and coverage of research data. Accuracy is increased by combining the activity of an individual moth with the exact time of day, humidity percentage and temperature degree. This is data that an analog trap does not offer. Coverage can be increased by researchers and volunteers being capable of setting up and tracking multiple traps with close to no manual work, whereas an analog trap requires visitations to the trap to count every individual insect by hand. Location data can be used to sort data based on vegetation zones, a key factor in analyzing moth behavior. Combining the data points produces daily, weekly, monthly and yearly trend graphs that can be utilised in drawing conclusions about biodiversity and the correlation of pollinator population with climate change (Leinonen et al., 2016).

Figure 1. Smart Insect Biomass Scale (SIBS) in one of its environmental contexts.

As a solution, SIBS has global potential. Through the web based application, more hobbyists, even ones that don't currently report their trap's contents could integrate their data into a service from where any party benefiting of the data could access it. However, with few hundred traps across the Nordics and possibly Europe (Sihvonen, 2019), there's no market potential that would indicate SIBS has business potential in a traditional sense. The traps have been so far self-built and low cost, and as every trap is benefiting on a societal scale, it would be of no value to have the current traps disappear. Hence we kept the idea of a do-it-yourself product which could be assembled and connected by anyone from a researcher to someone who would simply want to experiment with technology.

However, as the solution simply becomes worthless if the web based application is not functioning and maintained by a support team who benefits from enabling global data integration, the SIBS solution should be a part of a business ecosystem model. The focal actor could be such as the Centre for Economic Development, Transport and the Environment, also known as ELY Centres. They would provide funding for the web based application as well as a starter kit of some or all the parts it takes to build the trap. ELY Centres are to promote regional competitiveness, well-being and sustainable development and curbing climate change (ELY Centre, 2019), and they maintain moth traps across Finland and store data from them (Leinonen et al., 2016). We asses the ecosystem model further in Chapter 5: Viability and potential impact.

Figure 2. A self-built, analog moth trap (Leinonen et al., 2016).

The rest of this document describes the product concept in more detail. All presented ideas and definitions are made based on our research and prototyping of the product. We believe the desirability aspect of the product has been validated to a place where the next phase would be to validate the feasibility of the product by field testing it. This validation should be executed in co-operation with researchers and the focal actor of the business ecosystem after which the web based app could be tested and validated, and if successful, the solution could be built and taken into use.

1.1 Overall architecture

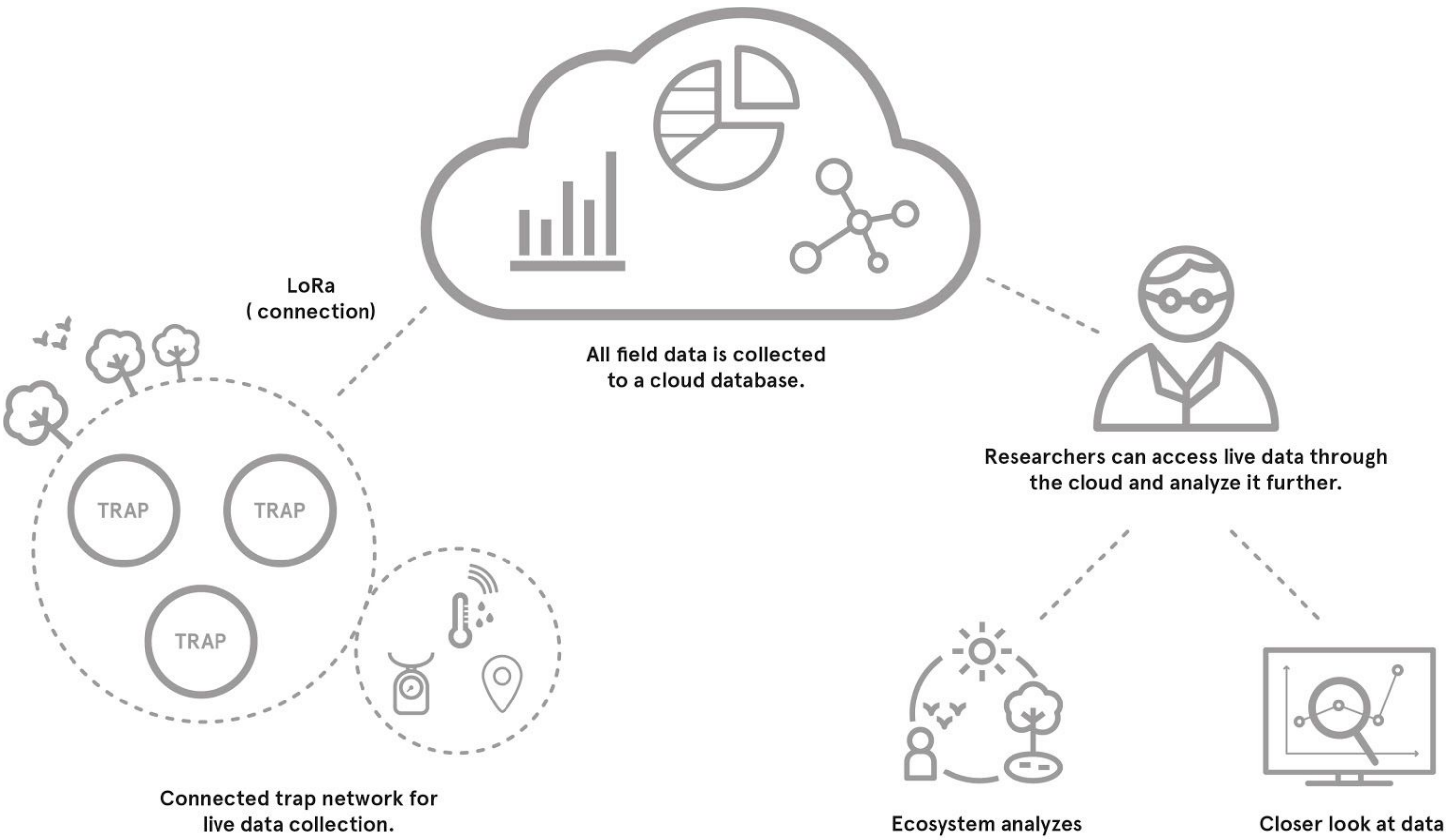

Figure 3. SIBS solution architecture

The Smart Insect Biomass Scale (SIBS) solution consists of an edge service, which is the moth trap, a cloud service that stores the data and a progressive web application that interprets the data into a visual and verbal form. The weight and movement sensors provide data of the moth population and the temperature and humidity sensors are providing live data about the surrounding conditions. The edge service needs to be connected to the cloud by using a mobile phone to guarantee the accuracy of the location. The web app will use the device location to define the trap location, and the trap itself won't have a GPS tracker.

The default data sending interval is every twelve hours to ensure that after every night, data is secured to cloud. But to guarantee real time access, the cloud service makes a call to the edge service when logged in. When a user logs in the web app sends a request for data to the cloud service which then requests data from the user's integrated edge services and returns data back to the web app. The edge service could utilise a 5G connection for example the low frequency connection LoRa that provides longer battery life and wide coverage (Ram, 2018).

To avoid constant burdening all of the edge services, real time access will be limited only to the user's own edge service, and the data globally is only refreshed every twelve hours. The data is processed centrally in the cloud to avoid loss of data and to guarantee scalability of storage space. The focal actor of the business ecosystem would be responsible for subscribing, monitoring and increasing the storage space.

It is yet to be decided which power outlet the edge service would use. One option is an AC connection to power outlet, which would guarantee continuous access to energy. The advantage of solar power and battery combination would be that the trap could be placed into areas where there is no power outlet access. This should be validated further with field research.

1.2 Interaction design

The main uses of the Smart Insect Biomass Scale (SIBS) edge service are:

- User must be able to access the toxin container to empty and add toxin
- User must be able to access the bug container to empty and clean it
- User must be able to assemble the trap
- User must be able to disassemble the trap
- User must be able to switch any of the physical parts of the trap

The main uses of the Smart Insect Biomass Scale (SIBS) web app are:

- User must be able to register (profile creation)
- User must be able to log in and log out
- User must be able to confirm the location of the trap
- User must be able to access and interpret data of their trap
- User must be able to access and interpret data of all connected traps
- User must be able to export data
- User must be able to sort and filter data
- User must be able to contact support
- User must be able to fill in additional data e.g. the environment of the trap, keep a trap journal and add multiple traps to their profile

Naturally the most important moment of interaction is taking the solution into use. The user must assemble the product themselves by piling the technology unit, tray, cover, funnel and roof of the edge service. Then the user should form a wireless connection to the trap in order to register it and themselves into the web service. After this the user needs to be able to access the web app at any preferred time to interpret the data collected. We explain edge service interaction further in Chapter 3: Embedded design.

The web service is to provide functionality of sorting and filtering the data. For example user should be able to select if they want to see daily, weekly, monthly or yearly data, as well as filter the traps they wish to see data based on location e.g. country or vegetation zone. We explain the web based application interaction further in Chapter 4: Progressive web app design.

The moth season covers roughly seven months, in between seasons the parts of the trap including the poison need to be stored. The poison used is tetrachloroethylene, which is deadly to people if inhaled or ingested (International Labor Organisation, 2019). This means that the poison cannot be stored with the parts of the trap, so it is vital the trap owner can safely access its container.

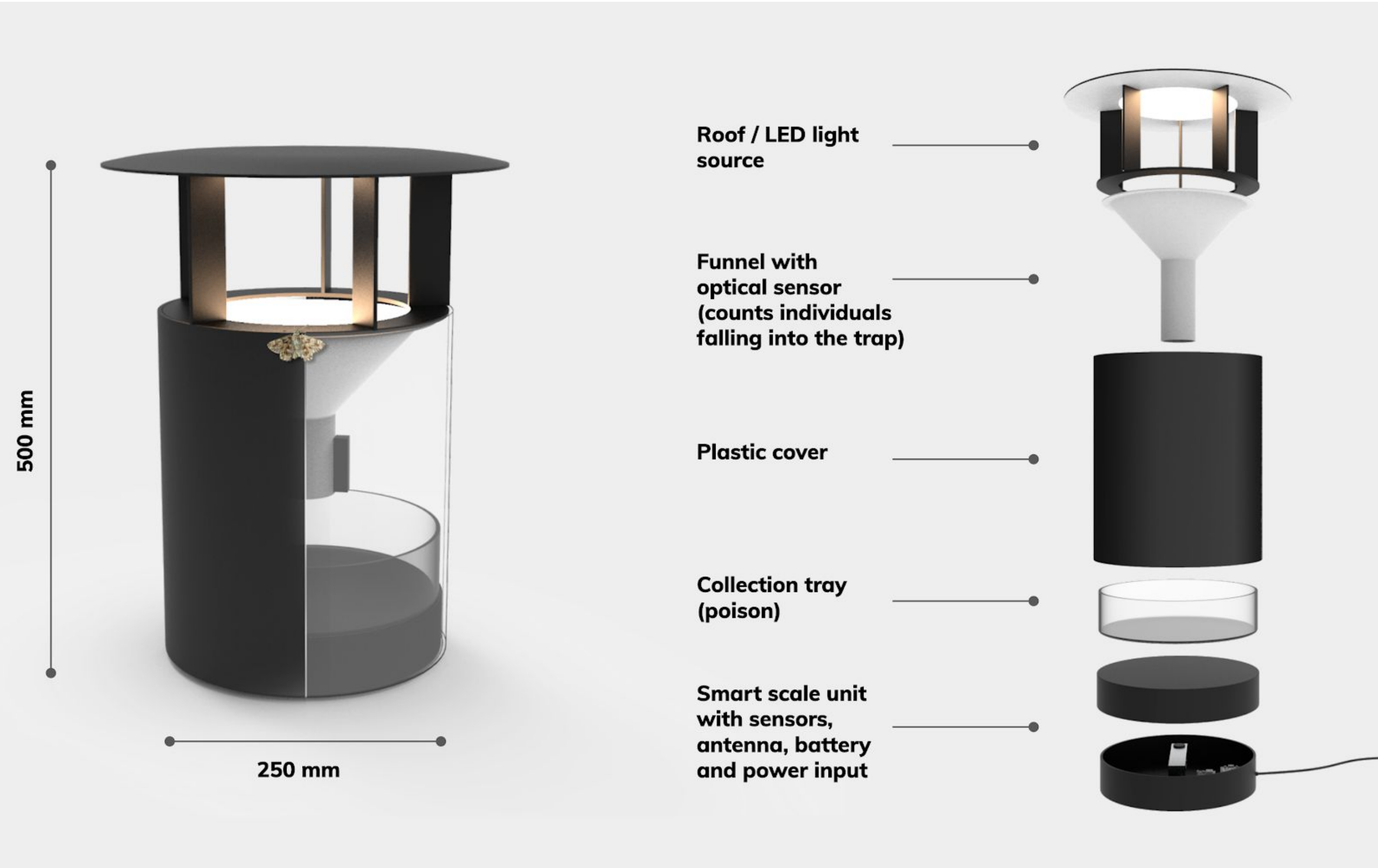

Figure 4. 3D render and specifications of the connected trap

2. Background research

The studies done display alarming evidence on the disappearance of pollinators, and more studies are needed to identify changes in populations at local, national, continental and global scales (Potts et al., 2010). While interviewing Sihvonen and Lilley (2019) the idea to improve both coverage and accuracy of moth research with an IoT solution was brought up.

Moths belong to the order of butterflies, though most moth species are nocturnal pollinators (Smithsonian Institute, 2019). Pollinators play a vital role in plant communities and agriculture of which both have suffered due to climate change impacting on moth populations' decline which in return is the causality for the increasing speed of climate change (Potts et al., 2010) as pollinators thrive on areas with multifaceted flora, where they help flora pollinate even more. Global warming is changing the quality of air which is also unfavorable for the plants, trees and flowers. (Leinonen et al., 2016).

Finland is one of the few countries that has taken systematic actions into moth research amongst which only Hungary and Great-Britain have similar, widespread, long-term studies about moths (Leinonen et al., 2016). Figure 5 below visualise the alarming study results from 1993 to 2012; moth populations are moving to the south, because they don't find suitable living conditions in Finland anymore. In addition, moths are nocturnal pollinators and find bright summer nights in Lapland unattractive.

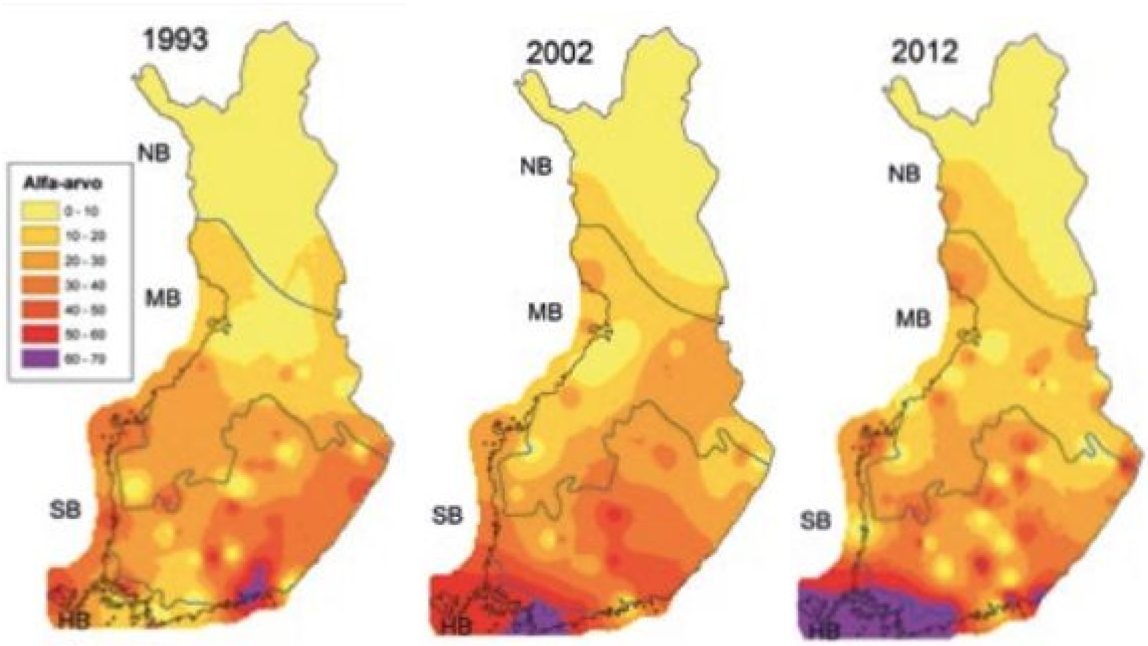

Figure 5. Moth populations in Finland 1993-2012 (Leinonen et al., 2016).

Both the interview with Sihvonen and Lilley (2019) and the research paper of Leinonen et al. (2016) have been important resources for our work. However, the issue is that the data for the study is provided by both researchers and hobbyists who have been doing manual monitoring of moth population since 1993. While this work has enabled long-term data documentation, it's not accurate enough in terms of scientific research (Leinonen et al., 2016). Additionally the data is distorted by the fact that most of the traps are situated in areas like coniferous forests where the moth population is naturally lower (Leinonen et al., 2016).

In addition, global studies like Potts et al. (2010) highlight that butterfly species investigation is vital for the continuum for long term data collection as other pollinator species lack similar structured research.

2.1 Discovery

As our team had a large scope of interests towards different topics, determining the focus took a while, but during our discovery, we found our research inspiring the many potentially viable IoT concepts. For the discovery phase we chose a qualitative research approach in the forms of interview and desk study. As researchers were also potential users of our solution, we combined expert and user interviews into one (Cooper-Wright, 2015). We conducted interviews both by visiting the experts' work environment and over a phone call. The sessions lasted around 30 minutes. Before the interviews, we had agreed on a discussion guide to guarantee comparability of the sessions.

We conducted an interview for two bird researchers: Juho Valkama, the team lead of Ringing Centre, and Markus Piha, a designer in the Museum of Natural History. As it turns out, more and more of bird species are exposed to biological annihilation and the pattern follows all of the animal species in the food chain (Valkama and Piha, 2019). Following the bird interviews, we had an interview about insects with Pasi Sihvonen, an insect researcher with extensive academic background, and Thomas Lilley, a zoology curator at the University of Helsinki. They came to the interview well prepared and presented us with three different problems threatening the insects.

To dive deeper into both themes, birds and insects, all team members conducted a desk study of their own aiming to have a better understanding of the subjects, especially the seriousness of the species disappearance and the level of IoT or other technical solutions available. As the discovery phase was only three weeks, part of which was spent finding a shared interest amongst the team members, we did not engage in other research methods, but found experts to be trustworthy enough for initial validation on product desirability, especially as our desk study proved to support the interviews' discussion points.

We presented three optional concepts on November 15, 2019 at the Aalto University to the course lecturer Salu Ylirisku and Kalle Airo, manager of the Aalto Ventures Program. Of the concepts pitched, we decided to focus on the smart insect biomass scale as we found insects often go unnoticed and have close to no solutions to improve research coverage.

Already in the very beginning, we can see our process following the double-diamond model with its first steps discover and define.

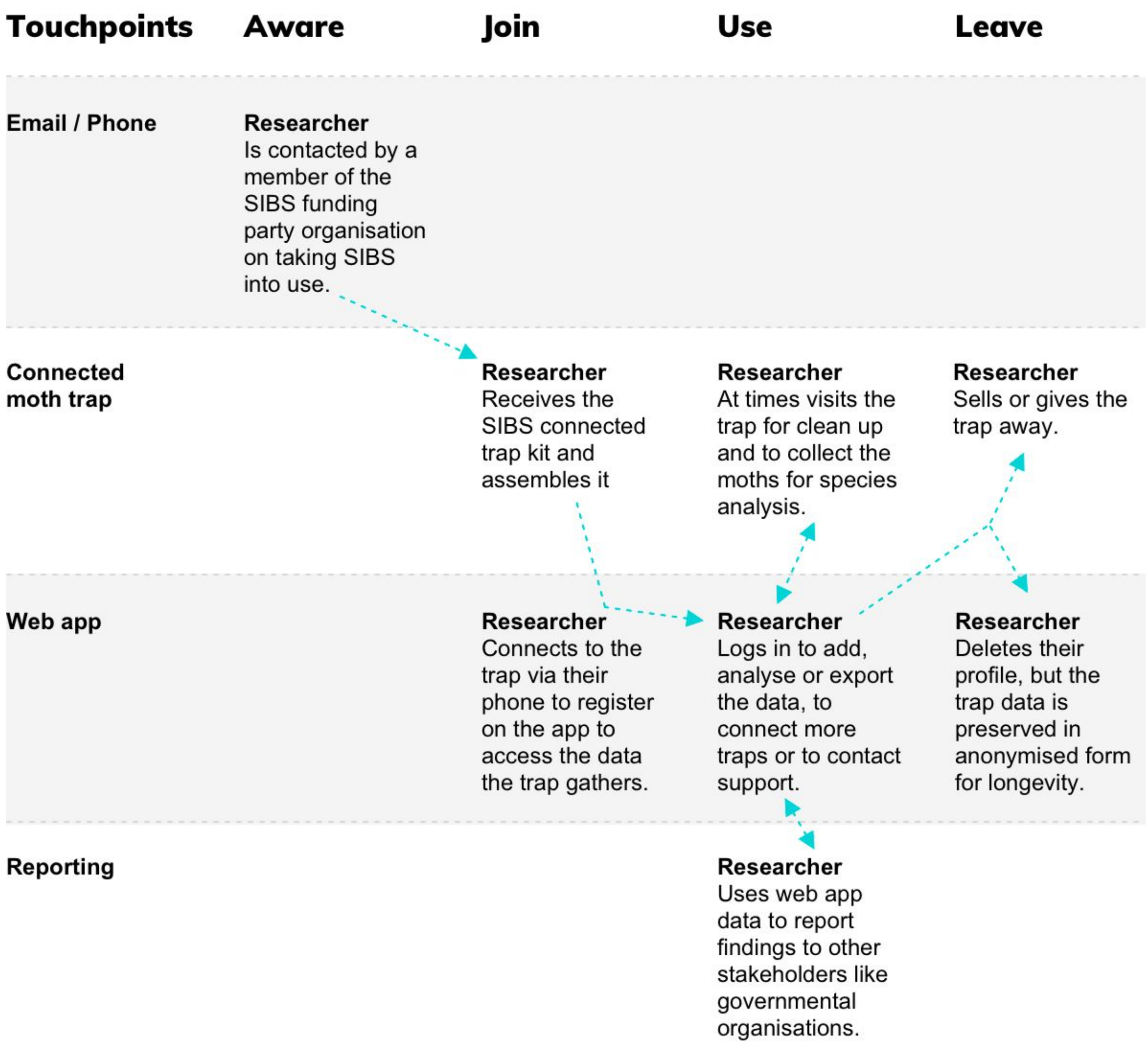

Figure 6. Journey chart example of researchers, one of the target users of SIBS.

2.2 Prototyping

In the development phase, we listed all the needed key features up and drafted some sketches of how the product would look like. We returned to the study of Leinonen et al. (2016) to ensure that we are taking the right direction to the problem. The team members responsible for embedded design figured out how to build the technology component for

SIBS, and developed Azure Cloud connection with the help of the assistant teachers. At the same time web designers figured out how to fetch the data from Azure, and what kind of a platform the data would be integrated into. Concept designers made sure the need of the end users was constantly kept in mind when the team was developing the concept further into details.

2.2 Validation

Once the prototype was built and a short draft for product opportunity document was made, we listed up the open questions that we still needed clarification for. We contacted Pasi Sihvonen to validate the value of our concept as well as to get direction for selecting between options like if a single user was to only see their own SIBS data or if the data should be accessible to all SIBS users. Sihvonen was impressed by the concept, and based on his comments we were able to make some crucial decisions still before the final presentation. A key insight was that our idea on shared data would be a one of a kind solution, enabling data sharing globally. This kind of solution had not existed before due to data sharing via research reporting only.

To truly get validation for our product, we need to wait for the summer, when the insects are active, and test different prototypes in nature.

3. Embedded design

The design of the edge service, a connected moth trap, is largely based on assumptions made during the six week course. The design would require rounds of field testing in cooperation with researchers, hobbyists and the party funding the web based application. The component pricing estimate we present is based on small scale production of a few hundred units. The partner funding the web app would also be responsible for manufacturing or assembling the technology unit and ordering the parts and reselling the package or distributing it for free if such funding is received. Optionally or in addition the funding party could provide assembly instructions for the trap or how the technology unit could be integrated with the current analog traps in use. Both approaches need to be further validated with the end users.

The function of the connected trap is similar to the current analog trap. Both traps use light to attract the moths and a toxin called tetrachloroethane to stun and eventually kill the moths. The key difference is in the connected nature of SIBS that we explain in Chapter 1: Design concept. Much like the analog trap, the connected trap has a roof fixture with light (1) that attracts the moths at night, a container for the poison (5), a funnel (3) through which the stunned moth will fall through and a container (4) for the moth bodies.

3.1 Components and price estimates

The following components are an initial assumption of the parts that could make up the connected moth trap that could measure small weights (mg), detect proximity (count) and measure temperature and humidity. This data is then possible to send to a cloud computing service such as Microsoft Azure from where it can be fetched to a web application.

The setup guide should be provided by the funding party, and explain at least how to assemble the trap and form a connection to a cloud service. A cloud connection can be formed by utilising existing scripts by simply loading the code from computer to microprocessor using Arduino software. However, in case some of the traps would be self-built, one should be careful with calibration of the load cell and making sure the numbers are accurate, as failing in these can have a bad effect on the reliability of gathered weight data.

Sensors

- load cell that can detect micrograms & amplifier (Sparkfun HX711)
- temperature, humidity and optical sensors

Connectivity specifications:

- LoRa connection 5G
- LoRa antenna unit

Sending and receiving data:

- infrequent data drop max. 10 KB
- on-demand data drop max. 10 KB
- Data can be send to cloud using ESP32

Power input:

- AC power connection
- Possibility of solar power or batteries needs more investigation

Other components:

- Microcontroller (ESP32)
- LED light color temperature

Physical structure:

- Cylindrical plastic pipe diameter 150-250mm
- Smart scale unit needs to have a housing with IP54 protection
- Physical structure needs to be stable and not tip over easily. If needed structure need to be attached to the ground with pegs.
- The trap needs to be protected with a roof that is 25% larger in diameter than the diameter of the pipe to prevent rain and other debris can not get into the trap.
- Estimated weight of whole construction 2-4 kg depending on the size and structure.
- Height 400-600mm

Figure 7. Connected moth trap 3D renders.

3.1.1 The Circuit diagram bill of materials (BOM)

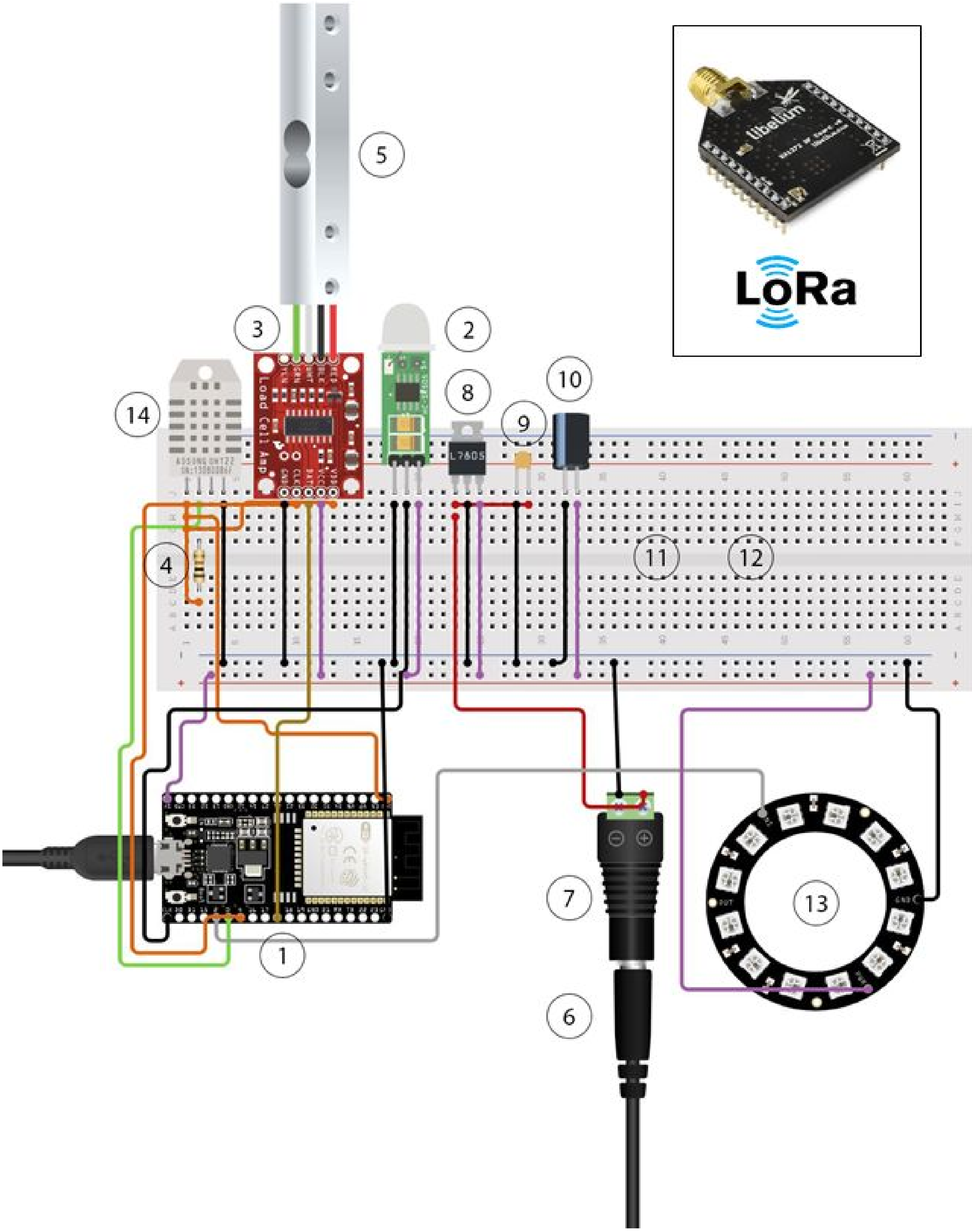

Figure 8. Circuit diagram 3D render.

Numeric reference	Part name	Price
1	Microprocessor: ESP32 – DevKitC	9,02 €
2	Infrared proximity sensor Mini PIR - HC-SR505	0,98 €
3	Load Cell Amplifier SparkFun HX711	9,0 €
4	10K Ohm Resistor	0,015 €
5	Load Cell Bar	9,18 €
6	Wall Adapter Power Supply 12VDC 2A	7,91 €
7	Female DC Power adapter 2.1mm jack to screw terminal block	1,8 €
8	Voltage Regulator 5V	1,35 €
9	Capacitor Ceramic 100nF	0,37 €
10	Electrolytic Capacitor 1uF/50V	0,086 €
11	BreadBoard / PCB	5,6 €
12	HeatSink TO-220	0,32 €
13	NeoPixel Ring 12 x 5050 RGBW LEDs	6,78 €
14	DHT22/11 Humidity and Temperature Sensor	9,0 €
15	LoRa antenna unit	20,0 €
16	Other	4,0 €
		85,38 €

3.1.2 The product exterior bill of materials (BOM)

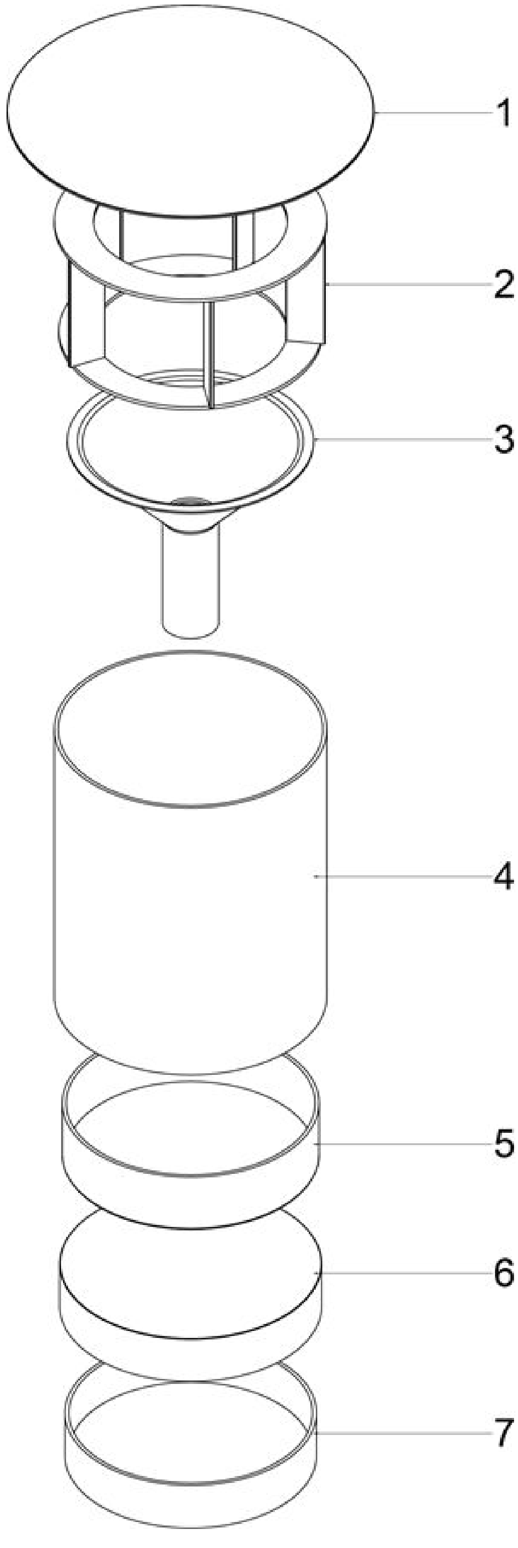

Figure 9. A wireframe of the connected moth trap product exterior.

Numeric reference	Part name	Material	Special requirement	Price
1	3D printed roof with LED fixture	ABS	25% larger than diameter of base pipe	5,0 €
2	Spacer bracket (3D printed)	ABS	N/A	5,0 €
3	Funnel (3D printed)	ABS	Optical sensor in funnel	10,0 €
4	Plastic pipe 150mm	PVC	N/A	5,0 €
5	Petri dish 140 mm	Polystyrene	N/A	0,5 €
6	Scale unit lid (3D printed)	ABS	IP54 protected housing	10,0 €
7	Scale unit base (3D printed)	ABS	IP54 protected housing	10,0 €
				45,50 €

3.2 Programming

The code for the prototype was done using Arduino IDE and written in C. The libraries used:

- HX711
- WiFi
- HTTPClient

Reading the sensor and sending the data is rather straightforward and can be done in relatively few lines of code. The main functions of the script are to connecting to the internet, reading the sensor values and sending them into the storage. Things such as how often data is sent can be adjusted according to availability of power and quality of network. One should still send data at least hourly to as researchers need to know how the amount of moths varies in certain hours.

Different devices must have their own Wifi credentials as well as unique sensor ID, that allows one to track from which SIBS the data is coming from. Otherwise they can use the same script. All the devices should also be calibrated so that the results are comparable, however this can be programmed to happen automatically when the scale is restarted so that it sets the initial weight as 0.

4. Progressive web app design

The web app of SIBS is an essential part of the product. With the web app, insect researchers and hobbyists can easily access data about the amount of moths across Finland (and Europe), as well as closer investigate the biomass in a certain measurement point. Additionally, through the website trap owners can add their traps to the database and control their own traps.

4.1 Functionalities

The most important function of the web app is to give users access to the data that is received from the trap. For this to be possible, there are two important features: a database that stores data received from the embedded device, and a user interface that allows users to access the data.

The embedded device sends data about measurements (measurement point, date and time, weight, temperature, humidity, amount of insects) to cloud regularly, and the data is stored in a database. With the help of the database, it is possible to draw heat maps and charts about the measurements, as well as create tables that researchers can use to analyze changes in biomass.

The user interface is a website that gives an overview of the measurement points and their status and historical changes in biomass. In addition, the users can download measurement data in Excel format for more in-depth analysis. Users can filter the data based on vegetation zones, countries, trap status and more.

Trap owners can register their own traps to the site and control the traps through the web app. They can for example remove their own traps from the service whenever needed. In the future, they could also receive email notifications whenever a problem occurs; for example, if the web app stops receiving data from the trap. There are no limitations on how many traps a user can add to the service.

4.2 Architecture

The chosen cloud service for this project is Microsoft Azure. The Azure services needed to get the functionalities working are IoT Hub, Stream Analytics, SQL database, Azure Functions, and SignalR. PowerBI is also used. These functionalities are used for three different purposes:

1) To show real-time data on the website
2) To store data and enable the download of data from the website
3) To visualize changes easily on the website

The role of the different functionalities is as follows: Azure IoT Hub enables the communication between the trap and other services, whereas Stream Analytics processes the streaming data. To be able to show real-time data on the website, Azure Functions and SignalR are also used. With SignalR and Azure Functions, messages are broadcasted to the user interface.

The process for showing real-time data on the website step-by-step: whenever the device sends data to IoT Hub, the Azure function activates and alerts SignalR. SignalR connects with the website backend, and sends the data forward to the website. The user interface updates accordingly. (van Uden, 2019.)

To easily visualize the changes of biomass over time on the website, data is also streamed to PowerBI. From the IoT Hub, data can be streamed first to Stream Analytics and forward to PowerBI. In PowerBI, reports and graphs about changes in biomass can be easily created and also embedded directly to the website.

One of the most important features of the user interface is the possibility to both browse and download data as an Excel file from several measurement points. For this to be possible, all data received from traps is saved to an SQL database. Data from the traps is automatically synced to the database from Azure, and fetched from the database whenever a user downloads a file from the website.

4.3 User interface design

For our product, interface isn't the most important part. As stated earlier, the most important thing is for researchers to be able to collect data. Ideally the interface won't show data from only one trap as in our current UI depicted below. We created the simple version to test our connections and prototype the scale.

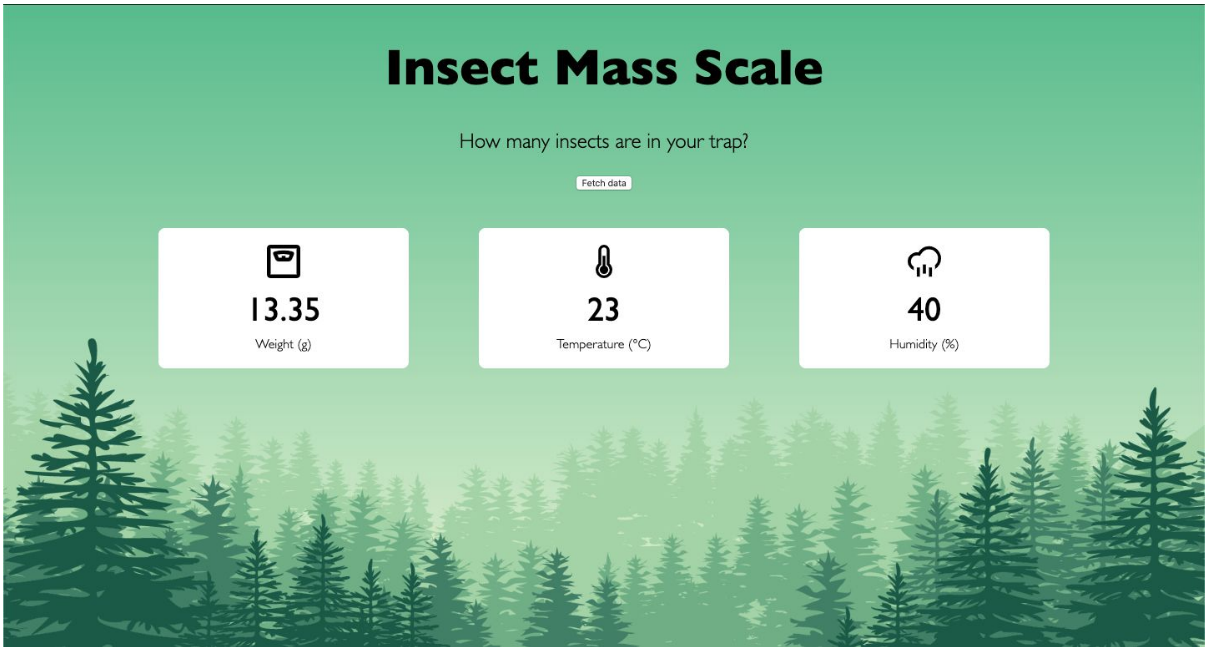

Figure 10. Current UI.

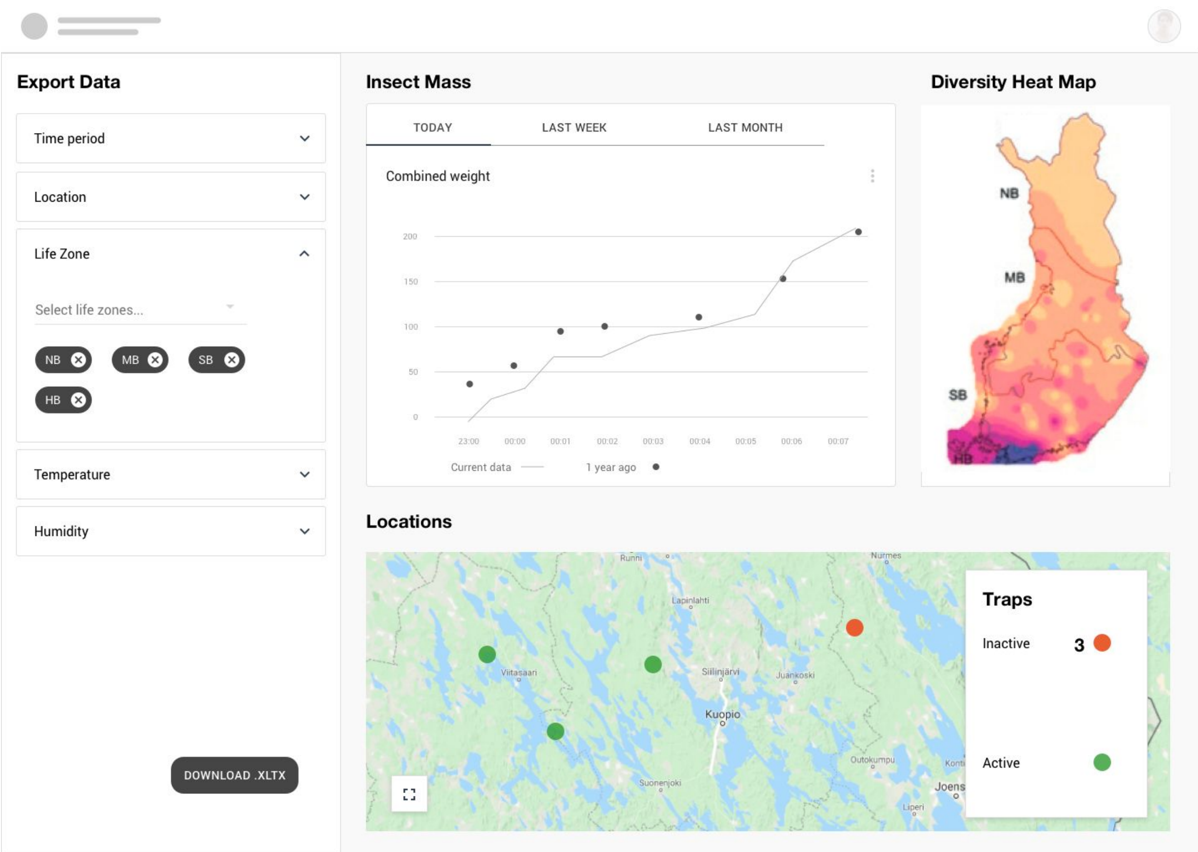

Figure 11. Wireframe of the future dashboard UI.

However, in the future, the UI should include data from all the traps. This way a kind of a dashboard could include more functions and give more insights for researchers.

One function that could make interface necessary is a map that would show which traps aren't working correctly. This would enable the researchers to reduce the amount of times they have to visit the traps. Previously the traps have been visited weekly but with this, the traps could stay in the forest, even the whole season, and the researchers would only go check them when they're notified of a problem.

The researchers should also be able to download the data in the form they want. In an ideal case, the dashboard could even create different types of graphs from the data automatically. For example, it could show the researchers how the weight has collected during the night in different parts of Finland.

4.4 Security and ethical considerations

The website is open for everyone, and visitors can see data from traps without credentials. However, if a trap owner wants to add their trap to the database, they need to create credentials to the site. The data that is stored about a user is their name, email, and trap locations and IDs. SIBS naturally also complies to GDPR: to ensure that the user knows what kind of data is collected about them, a privacy policy is added to the website. In addition, by contacting the SIBS support, the users can receive information about stored data and also request for the stored data to be removed. The site does not not store any info about the users who visit the page without logging in, which means that there are no big risks regarding privacy and user security for them.

The greatest security risk in the service is linked to downloading a file from the website. In case the site faces a cyber attack, there is a risk that the data file is replaced with a file that contains a security threat, such as a virus. To avoid any possible security threats, there are some actions that should be taken to secure the website. First of all, the software should be kept up-to-date to ensure that there are no security holes. In addition, the site should use HTTPS to add an extra layer of security.

In addition, it should be noted that the site shows the location of all SIBS traps publicly. This can become a security issue if traps are located near private property, and they are visited by externals. In the SIBS user manual, there should be clear instructions about where a SIBS can be located, and that by connecting the SIBS to Internet, the location will be publicly available for anyone to see.

5. Viability and potential impact

The main users are the insect researchers, because SIBS doesn't have that big of a market segment on the consumer side. However, anyone can still have their own trap without the poison. If we would create a freezer-based trap which doesn't need poison, anyone could join the research. The only physical addition we are offering, is the IoT technology and data storage. What needs to be researched further is if hobbyists can be incentivised with societal fulfillment they'd get from participating.

The Minimum Viable Product (MVP) of our project would be an existing trap with a smart IoT scale in the bottom of the collection bucket. This way we can try the concept and see if it works and can provide extra value for the research. We can utilize the existing AV power sources, too.

5.1 Competition and Value Creation

Currently there is no direct competition for our idea, besides the current methods of researching moths, which are inferior and provide less data. However, different projects compete for the funding for research projects, so we need to convince the decision makers that our project is valuable.

The potential impact of the device can be pretty large, at least for the researchers in the field. It is a rather cost-effective and scalable way to learn more about insect populations. Even though it does not directly help in stopping the decrease in insect populations, the result can give valuable information about the current state. As the technology used is rather simple and cheap, one can implement the traps in many different ways, for example as full products or scale modules that are added to existing traps.

Also, we need to contact and research similar studies in other countries to see how they are conducting the studies and how we can combine the efforts. We can collaborate across EU to create more impact and value for the research, and better our chances of getting funding.

5.2 Monetization

Our product won't make money with sales, so we're looking into alternative funding methods. The most likely option is to apply for funding from the European Union for the whole project, which would also include funding for developing the traps.

If the European Union decides not to fund the project, the options for funding are quite limited. We would have to research and contact more non-governmental organizations that fund research on insects and mass extinction. An advantage of the project is that it is rather affordable, yet extremely useful for scientists. The trap could be developed further for example as a student project and ordering ready-made devices is easy and inexpensive.

One option is to try to get funding by crowdfunding. This would mean opening a campaign on GoFundMe or similar platform, where people aren't expecting anything in return for their money. This wouldn't likely work as the researchers are part of the University of Helsinki, and they should get their funding from certain, trusted methods.

If we find out there actually is demand from hobbyists to get these traps, we could also put a price on the traps. This could be subsidised by the research funding or other methods, if the data could still be used by researchers. Getting more volunteers to participate could be more important than getting money from them, so this option is still under consideration.

5.3 Price estimates

As calculated in the chapter 3, the price estimate for components is 85,38 € and for the exterior 45,50 €.

For the software side, prices are harder to estimate because they're dependent on the costs of developing and technologies used. The calculations might be far from the end price, as they are based on guesses, and the number of devices and the used technologies might change. Here are some estimates how much Azure would cost:

Microsoft Azure Estimate			
Your Estimate			
Service type	Region	Description	Estimated Cost
Azure SQL Database	North Europe	Single Database, vCore Purchase Model, General Purpose Tier, Serverless, Gen 5, 2 Billed vCores, 32 GB Storage, 0 GB Backup Storage	$6,27
Azure IoT Hub	North Europe	Standard Tier, Free: 500 devices, 8,000 msgs/day, $0.00/mo, 0 IoT Hub Units	$0,00
Azure Stream Analytics	North Europe	1 Standard streaming units 730 Hours, 1 Stream analytics on edge devices	$88,60
Azure Functions	North Europe	128 MB memory, 100 milliseconds execution time, 0 executions/mo	$0,00
Azure SignalR Service	North Europe	Free Tier, 1 Unit	$0,00
Support		Support	$0,00
		Licensing Program	Microsoft Online Services Agreement
		Monthly Total	$94,87
		Annual Total	$1 138,41
Disclaimer			
All prices shown are in US Dollar ($). This is a summary estimate, not a quote. For up to date pricing information please visit			
This estimate was created at 12/12/2019 11:24:23 AM UTC.			

On top of these costs, we would either need to collaborate with someone to create and improve on the prototypes. For If we can collaborate with universities, we can cut costs, but this might not be possible. We can also use collaborative resourcing and crowdsourcing. This is why we are also making the project open source.

However, if we can't find willing participants, we need to hire hardware- and software development businesses, which can get expensive. The research budget might be too small, which means the IoT project won't go forward.

6. Publicity and license

This paper, as well as all of our code, are public and open source. Also everything is CC-BY-NC, so they are free to use when credited, but not for commercial purposes.

We want to make this into an open source project, because we want more people to help with this important cause. We could upload all of the code, results and instructions for building the physical traps online. This is because we want to encourage more people to help with the research and continue our work forward. Also, this way sharing the ideas and methods with international community of researchers is easier.

We gladly leave all our work to be used by future students, researchers, engineers, and others who are interested in the subject and would want to take the project forward. Especially as the researchers we interviewed were interested in collaboration with some students to take the IoT possibilities forward, we are happy to share this research with anyone interested. Our team doesn't have the needed knowledge about creating IoT products to continue further, so we want to use this opportunity to connect people that would otherwise have a hard time connecting: insect researchers and engineers.

7. References

Carrington, D. (2017, July 10). Earth's sixth mass extinction event under way, scientists warn. Retrieved from https://www.theguardian.com/environment/2017/jul/10/earths-sixth-mass-extinction-event-already-underway-scientists-warn

Cooper-Wright, M. (2015). 12 Design Research methods to get inspired by users. Retrieved from https://medium.com/design-research-methods/12-design-research-methods-to-get-inspired-by-users-cae4789a094b

International Labor Organisation. (2019). Tetrachloroethane. Retrieved from https://www.ilo.org/dyn/icsc/showcard.display?p_card_id=1486&p_version=1&p_lang=en

Leinonen, R. Pöyry, J, Södeman, G. and Tuominen-Roto, L. (2016). Suomen yöperhosseuranta (Nocturna) 1993-2012, The Finnish Moth Monitoring Scheme 1993-2012 (In Finnish). Suomen ympäristökeskuksen raportteja 15/2016, 71 p. ISBN 978-952-11-4565-0 (PDF).

Light, A. and Rowland, C. (2015). Chapter 11: Responsible IoT Design. In C. Rowland, E. Goodman, M. Charlier, A. Light, & A. Lui (Eds.), Designing connected products: UX for the consumer Internet of things (First edition, pp. 419-466). Sebastopol: O'Reilly.

Potts, S.G. Biesmeijer, J.C. Biesmeijer, J.C. Kremen, C. Neumann, P. Schweiger, O. and Kunin, W.E. (2010). Global pollinator declines: trends, impacts and drivers. Trends in Ecology and Evolution, 25(6). 345-353.

Ram, P. (2018). LPWAN, LoRa, LoRaWAN and the Internet of Things. Retrieved from https://medium.com/coinmonks/lpwan-lora-lorawan-and-the-internet-of-things-aed7d5975d5d

Sihvonen, P. and Lilley, T. (2019). Phone interview on ecosystems and insect research. 8.11.2019.

Smithsonian Institute. (2019). BugInfo: Moths. Retrieved from https://www.si.edu/spotlight/buginfo/moths

Valkama, J. and Piha, M. (2019). Interview on tracking birds. Conducted at the Finnish Museum of Natural History. 6.11.2019.

van Uden, B. (2019, May 2). Sending IoT data real-time to the browser with Azure IoT Hub, Azure Functions, and SignalR Service [Blog post]. Retrieved from https://www.bartvanuden.com

Appendix

JSON

```
/*
By: Tuomas Tapper and Botond Sléber
*/
#include <WiFi.h>
#include <HTTPClient.h>

#include "HX711.h" //This library can be obtained here
http://librarymanager/All#Avia_HX711

#define calibration_factor -450.0 //This value is obtained using the
SparkFun_HX711_Calibration sketch

#define SCALE_PIN_DAT  2
#define SCALE_PIN_CLK 0

HX711 scale;

const char* ssid = "network_name";
const char* password =  "";

String sensorname = "SIBS";

float sensorvalue = 0.333;
String sensorstatus= "ok";

String initjson = "{\"name\":\"rysa\",\"value\":0.333,\"status\":\"ok\"}";
//Specify JSON message

String sensorid = "5de4f2fb4e1d7d0011908eda";
String httpaddress;
String json;

void constructJson(String _sensorname, float _sensorvalue, String
_sensorstatus){
  char charvalue[8]; // Buffer big enough for 7-character float
  dtostrf(_sensorvalue, 6, 2, charvalue); // Leave room for too large numbers!
  String stringvalue = charvalue;

  json = "{\"name";
  json = json + "\":\"";
  json = json + _sensorname;
```

```
  json = json + "\",\"value\":";
  json = json + stringvalue;
  //json = json + charvalue;
  //json = json + _sensorvalue;
  json = json + ",\"status\":\"";
  json = json + _sensorstatus;
  json = json + "\"}";
}

void updatedata(String _sensorname, float _sensorvalue, String _sensorstatus){
  String response;
  if(WiFi.status()== WL_CONNECTED){   //Check WiFi connection status

   HTTPClient http;
   httpaddress =
"http://neppi-server.northeurope.cloudapp.azure.com:3000/sensor/" + sensorid;
   http.begin(httpaddress);  //Specify destination for HTTP request
   http.addHeader("Content-Type", "application/json");           //Specify
content-type header
   constructJson(_sensorname, _sensorvalue, _sensorstatus);
   Serial.print("Update data: ");
   Serial.println(json);
   int httpResponseCode = http.PUT(json);   //Send the actual POST request
   Serial.print("Update response: ");
   if(httpResponseCode>0){

       response = http.getString();                        //Get the response to
the request

       //Serial.println(httpResponseCode);   //Print return code
       Serial.println(response);        //Print request answer

   }else{

       Serial.print("Error on sending POST: ");
       Serial.println(httpResponseCode);

   }
   http.end();  //Free resources
 }else{
       Serial.println("Error in WiFi connection");
 }
 Serial.println("");
}

void setup() {

  Serial.begin(115200);
  delay(4000);   //Delay needed before calling the WiFi.begin

  scale.begin(SCALE_PIN_DAT, SCALE_PIN_CLK);
```

```
  scale.set_scale(calibration_factor); //This value is obtained by using the
SparkFun_HX711_Calibration sketch
  scale.tare();  //Assuming there is no weight on the scale at start up, reset
the scale to 0

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) { //Check for the connection
       delay(1000);
       Serial.println("Connecting to WiFi..");
  }

  Serial.println("Connected to the WiFi network");
  Serial.print("SensorId: ");
  Serial.println(sensorid);
}

void loop() {

  float weight= scale.get_units();

  sensorname = "rysSIBSä";

  sensorvalue = abs(weight);

  Serial.print("Reading: ");
  Serial.print(scale.get_units(), 1); //scale.get_units() returns a float
  Serial.print(" g"); //You can change this to kg but you'll need to refactor
the calibration_factor
  Serial.println();

  sensorstatus = "lit";

  updatedata(sensorname, sensorvalue, sensorstatus);

  delay(100);
}
```

HTML

By: Heini Könönen, Liina Hilkamo & Abraham Martinez Ornelas

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="number.css">
    <title>Document</title>
</head>
<body>
    <div class="bg">
        <header>
            <h1>Insect Mass Scale</h1>
            <h3>How many insects are in your trap?</h3>
        </header>
        <button id="test-sensor">Fetch data</button>
    </div>
    <div class="container">
        <div class="counter col_fourth">
                <img src="Images/scale.svg" class="icon" id="weight"
alt="scale"></img>
                <h2 class="timer count-title count-number" id="success-text"></h2>
                <p class="count-text"> Weight (g)</p>
            </div>
            <div class="counter col_fourth">
                <img src="Images/thermometer.svg" class="icon"
alt="thermometer"></img>
                <i class="fa fa-coffee fa-2x"></i>
                <h2 class="timer count-title count-number">23</h2>
                <p class="count-text ">Temperature (°C)</p>
            </div>
            <div class="counter col_fourth">
                <img src="Images/rain.svg" class="icon" alt="humidity"></img>
                <i class="fa fa-lightbulb-o fa-2x"></i>
                <h2 class="timer count-title count-number">40</h2>
                <p class="count-text ">Humidity (%)</p>
            </div>
    </div>
    <img src="Images/forest.png" class="backgroundimg" alt="forest"></img>
```

```
    <script src="number.js"></script>

<!-- Communication with the server -->

<input type="text" class="hide" value="Sensor name" id="sensor-name"
onfocus="this.value=''">

<button class="hide" id="add-sensor">Add sensor</button>

<script type='text/javascript' src='bug-min.js'></script>

<br><br>

<br><br>

<div id="success-text"></div>


<script lang="javascript">

// sensor_id = 5de4f2fb4e1d7d0011908eda

    // This will hold the ID for the sensor

    var my_sensor = "";


    // Add script to the Add Sensor button

    document.getElementById('add-sensor').onclick = function() {

        // Read the value of the sensor-name input field

        var sensor_name = document.getElementById('sensor-name').value;

        // Create the JSON for server call

        var json = "{ \"name\" : \""+sensor_name+"\", \"value\" : \"none\",
\"status\" : \"free\" }";


        // Send the request to the server

        request = new XMLHttpRequest();

        request.open('POST',
'http://neppi-server.northeurope.cloudapp.azure.com:3000/sensor');

        request.setRequestHeader('Content-Type', 'application/json');

        request.send(json);


        // This is executed when the request is done

        request.onload = function() {

            // This contains the JSON from the server - example: {"message":"Sensor
successfully created","id":"5de4e8524e1d7d0011908ecc"}

            var response_object = JSON.parse(request.response);

            my_sensor = response_object.id;
```

```
            document.getElementById('success-text').innerHTML = "<hr><p>Connection
successful!<br>Sensor: "+sensor_name+" ID:[<strong>" +my_sensor+ "</strong>] added.
<br>Please, copy the id to remember it!</p>";

        }

    }

    // Test if sensor exists

    document.getElementById('test-sensor').onclick = function() {

        // Read the value of the sensor ID from the input field

        var sensor_id = "5de4f2fb4e1d7d0011908eda";

        // var sensor_id = document.getElementById('test-sensor-id').value;

        // Create the JSON for server call

        var request = new XMLHttpRequest();

        var url = 'http://neppi-server.northeurope.cloudapp.azure.com:3000/sensor/'
+ sensor_id;

        request.open('GET', url);

        request.send();


        // This is executed when the request is done

        request.onload = function() {

            // This contains the JSON from the server - example: {"message":"Sensor
successfully created","id":"5de4e8524e1d7d0011908ecc"}

            var response_object = JSON.parse(request.response);

            document.getElementById('success-text').innerHTML = "<h2>"
+response_object.value+ "</h2>";

        };


    }

</script>

</body>

</html>
```

CSS

```css
h1 {

    font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;

    font-size: 60px;

    margin-top: 40px;

    margin-left: auto;

    margin-right: auto;

    color: black;

    text-align: center;

}


h3 {

    font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;

    font-size: 25px;

    font-weight: lighter;

    color: black;

    text-align: center;

}

h2 {

    font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;

    font-size: 40px;

    margin-top: 10px;

    margin-bottom: 0;

    text-align: center;

    font-weight: normal;

    color: black;

}

h4 {

    font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;

    font-size: 60px;

    color: black;

}

body {
```

```
    font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;

    background: rgb(240,242,216);

    background: linear-gradient(0deg, rgba(240,242,216,1) 0%, rgba(112,182,142,1)
100%);

    background-repeat: no-repeat;

    height: 100%;

    background-attachment: fixed;

    text-align: center;

    margin: 0;

    text-align: center;

  }

.icon {

    width: 40px;

}

.weight {

    width: 35px;

}

.container {

    position: relative;

    display: inline-flex;

    width: 80%;

    margin: auto;

    margin-top: 25px;

    align-content: center;

    justify-content: space-around;

    flex-flow: row wrap;

}

.backgroundimg {

    width: 100%;

    position: fixed;

    bottom: 0px;

    left: 0;

}

  /*-=-=-=-=-=-=-=-=-=-=-=- */
```

```
  /* Column Grids */

  /*-=-=-=-=-=-=-=-=-=-=-=- */

  .col_fourth {

      position: relative;

      display:inline-flex;

      flex-wrap: wrap;

      display: inline-block;

      width: 300px;

      margin-bottom: 20px;

      margin-top: 20px;

  }

  /* Column Grids End */

  .wrapper { width: 800px; margin: 30px auto; position: relative;}

  .counter { background-color: #ffffff; padding: 20px 0; border-radius: 8px;}

  .count-title { font-size: 40px; font-weight: normal;  margin-top: 10px;
margin-bottom: 0; text-align: center; }

  .count-text { font-size: 16px; font-weight: lighter;  margin-top: 10px;
margin-bottom: 0; text-align: center; }

  .fa-2x { margin: 0 auto; float: none; display: table; color: #218999; }

.sensor-name {

    position: fixed;

    bottom: 0px;

}

.hide {

    display: none;

}
```

JavaScript

```
    // Test if sensor exists

    document.getElementById('test-sensor').onclick = function() {

        // Read the value of the sensor ID from the input field

        var sensor_id = document.getElementById('test-sensor-id').value;


        // Create the JSON for server call

        var request = new XMLHttpRequest();

        var url = 'http://neppi-server.northeurope.cloudapp.azure.com:3000/sensor/'
+ sensor_id;

        request.open('GET', url);

        request.send();

        // This is executed when the request is done

        request.onload = function() {

            // This contains the JSON from the server - example: {"message":"Sensor
successfully created","id":"5de4e8524e1d0011908ecc"}

            var response_object = JSON.parse(request.response);

            document.getElementById('success-text').innerHTML = "<p>Sensor found!
Name: "+ response_object.name +" value:[" +my_sensor+ "] Status: ["+
response_object.status+"].</p>";

        };

    }
```

Interview discussion guide

1. Please describe in short who are you and your role at work
2. What kind of repetitive problem scenarios do you encounter at you work? Why do these happen? Could technology solve any of them?
3. Is your job made of desk and/or field work? What kind of tasks does the work entail?
4. Are IoT solutions familiar to you? In your area of expertise, have any IoT solution been taken into use? If so, for what?
5. Do you personally use any IoT solutions for your work? For what kind of tasks?
6. Can you think of any value for being able to access real time data remotely?
7. How specific does the data you collect need to be? Would it be of any use to have more accurate data and/or more data coverage?
8. What are the most important things to track in your job?
9. What kind of data do you combine to understand the environment and the inhabitants of the species you track?
10. Where do you collect data to? Do you have a shared database?
11. Why do you follow species (x)? What can be concluded from their behavior?
12. What should we consider if taking technology to nature and the species' (x) habitat?