



Project: Beetle

22. april 2015

qzj710 - 121095 - Enes Golic
rpc308 - 070493 - Yunus Emre Okutan
cbh239 - 250594 - Casper Lützhøft Christensen
mhb558 - 250795 - Tor-Salve Dalsgaard

Instructor - Kasper Passov

Contents

1	Introduction	3
1.1	Abstract	3
2	FACTOR	4
2.1	Functionality	4
2.2	Application-domain	4
2.3	Conditions	4
2.4	Technology	4
2.5	Objects	4
2.6	Responsibility	4
3	Program Specifications	5
3.1	Functional and Non-Functional Requirements	5
3.2	Use Case Model	6
3.3	Specific Use Cases	7
3.4	Class-diagram	8
3.5	BCE-model	8
3.6	Sequence-diagram	8
4	System Design	9
4.1	Current Progress	9
4.2	Summary	9
5	Testing	9
5.1	Search-engine test	9
6	Interaction and Design	10
6.1	Design	10
7	Internal Cooperation	12
7.1	Summary	12
8	Annexes	13
8.1	Version Control	13
8.2	Changelog	13
8.3	Timeline	13

1 Introduction

1.1 Abstract

Our project is, in broad terms, a search-engine for insects, which is to be provided to our client, the University of Hamburg.

Through German law and regulation, it is specified that in order for a university to receive governmental funding, it must allow public access to the research done by the specific university, and it is in this context our project is necessary.

The entomology department of the University of Hamburg has a catalogue of information on different insects, their names, species, genus and a picture of the insect's anatomy.

This information needs to be available to the public, and the University of Hamburg wishes this done through the usage of a search-engine connected to a database.

The database must contain the insects as entries, with their names, species, genus and anatomy-pictures as attributes. The search-engine must then allow a visitor to input a search-term, which will be tried against any of the database's attributes, and return any entry that matches the search-term. It must also contain an advanced search-function that allows you to match a specific term to a specific attribute.

Furthermore, the University of Hamburg requires a method through which they can update the database themselves. Both to allow them to edit entries that may contain errors and to add/remove entries to/from the database. This method must only be available to authorised employees and not at all available to visitors on the website.

We are creating the search-engine in PHP, and it is to be implemented onto their website. The database is being created in MySQL and it will be editable through a method created either in PHP and implemented onto the website itself, or a stand-alone program, created in Java, given to the authorised people.

2 FACTOR

2.1 Functionality

Allow visitors to use the 2 search-functions, basic and advanced, to search through a database of insect-entries with data collected by the University of Hamburg. Allow employees with the necessary login credentials to edit the database to either remove entries, change existing entries or add new entries.

2.2 Application-domain

Allow free, public access of data collected through research done by the University of Hamburg, so that everyone may learn from the conducted research.

2.3 Conditions

The product is being developed as part of the course PKSU at the University of Copenhagen. It is done entirely voluntarily and is non-profit. The client's wishes to the product are all taken into account and will be attempted fulfilled, although no complete product can be expected.

2.4 Technology

The product is developed on completely normal personal computers - no special tools are necessary in development apart from a PC. The product is designed to run in any browser without any necessary plugins, meaning that nothing but a PC and a working internet connection is required to use the product.

2.5 Objects

Anonymous visitors and employees can respectively view and edit the database entries.

2.6 Responsibility

The responsibility of the product is, through a search-engine, to ensure free and public access to research conducted by the University of Hamburg.

3 Program Specifications

3.1 Functional and Non-Functional Requirements

If we refer to the abstract outlining the program's specifications, we get a general feel of the architecture of the software and how it is going to work. Drawing on those points we can, for the sake of overview, specify functional and non-functional requirements for the program.

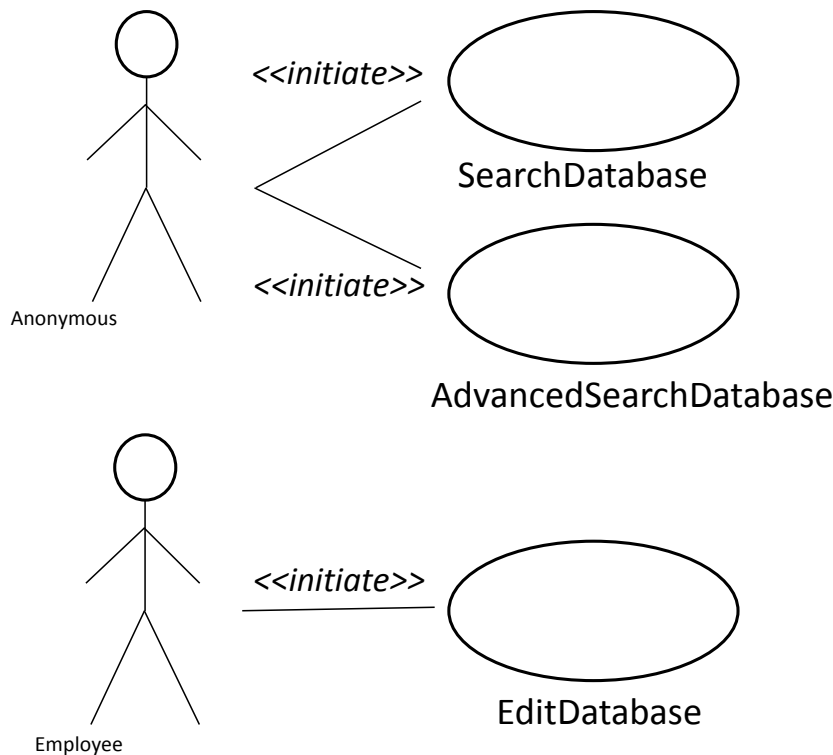
Functional Requirements

- The product must offer a search-option that returns any and every entry that contains an attribute that matches the search input.
- The product must offer a search-option that allows matching of specific search-inputs to specific attributes and return any and every entry that matches the terms in their specific attributes.
- The database must allow the creation of new entries, editing of existing entries and deletion of existing entries.
- The database must be unable to be edited by anyone save the authorised employees.
- A fitting message must be displayed if no matching entries are found.

Non-Functional Requirements

- The search-engine must be developed in PHP so it can run on their website in any browser with no additional requirements.
- The method of editing the database must be simple enough for people without pre-existing IT skills to use it.
- The product in its finished state must look nice and fit with the colour-scheme already present on the University of Hamburg's website.

3.2 Use Case Model



SearchDatabase

Anonymous users use the search-engine and input a term in order to search the database and have the results they require displayed. Searching the database does not require any special permissions.

AdvancedSearchDatabase

Anonymous users use the advanced search-engine and input their search-terms into the fields representing the attributes they wish to match them against, in order to have the results they require displayed. Searching the database does not require any special permissions.

EditDatabase

An employee edits the database by accessing the edit-page through entering the necessary login credentials, modifies the database in which way desired and updates the database. EditDatabase cannot be used without valid login credentials.

The figure above is a use case model, which showcases the different possibilities of the product and to which kind of users they are available.

It is for instance evident through the model and the description of the functions, that any user will be able to search the database both in a basic and advanced way, whereas only an employee with the necessary login credentials will be able to edit the entries in the database.

3.3 Specific Use Cases

With the use case model created, we have an overview of the functional architecture of the website. With that in mind, we can explore some of the specific cases a bit further:

Use Case Name	SearchDatabase
Participating actors	Initiated by Anonymous
Flow of events	<ol style="list-style-type: none">1. An anonymous user enters the website.2. The anonymous user enters the search-term into the search-field and presses send.3. The matching results are displayed.
Entry condition	None
Exit conditions	<ul style="list-style-type: none">•Anonymous exits the site.•Anonymous submits a new search term.

The above use case is the ordinary search of the database. It showcases in detail the specifics of the use case by displaying who initiates it, who, if any, it communicates with, the flow of the case itself as well as any conditions necessary. We will now also inspect the remaining 2 use cases:

Use Case Name	AdvancedSearchDatabase
Participating actors	Initiated by Anonymous
Flow of events	<ol style="list-style-type: none">1. An anonymous user enters the website.2. The anonymous user enters the search-terms into the fields of the matching attributes and presses send.3. The matching results are displayed.
Entry condition	None
Exit conditions	<ul style="list-style-type: none">•Anonymous exits the site.•Anonymous submits a new search term.

As we can see in the above case, no additional permissions/credentials are necessary to use the advanced search function.

Use Case Name	UpdateDatabase
Participating actors	Initiated by Employee Communicates with Anonymous
Flow of events	<ol style="list-style-type: none">1. An employee user enters the website.2. The employee clicks on the edit hyperlink.3. The employee enters his/her login credentials.4. The employee makes the desired changes to the database and presses submit.
Entry condition	•Must have login credentials
Exit conditions	<ul style="list-style-type: none">•Employee exits the site.•Employee cancels the editing.

In this use case however, it's evident that in order to update the database, login credentials are needed. These are only to be provided to employees.

3.4 Class-diagram

3.5 BCE-model

3.6 Sequence-diagram

4 System Design

4.1 Current Progress

The Prototype of our product can be found and accessed at: <http://echiever.de/ProjectBeetle/prototype.php#>

4.2 Summary

As of now, we're in the beta stages of the product. We've established a database containing 2 entries through MySQL using test-entries defined by ourselves. The basic search-engine is perfectly functional and returns matching results or an error message if none are found. The advanced search-engine has been designed and contains the 4 different search-fields, it is however not yet operational. A hyperlink leading to the editing of the database has been created, but the method of editing itself is still yet to be implemented. Finally, the design of the product so far is very basic as we're still in testing/implementing phase. Going forward, our tasks of importance are:

- Make the advanced search-engine operational
- Implement the editing portion of the product
- Establish the "correct" database using data provided by the University of Hamburg
- Improve the design to better match that of the University of Hamburg's existing website

5 Testing

5.1 Search-engine test

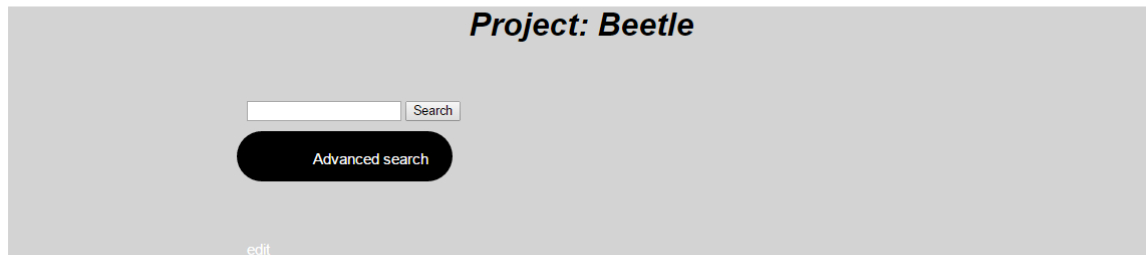
Through testing the basic search-engine, we've seen that sending in a search-input with no match will cause the error message "No Entry". We've also seen that inputting any term matching one of the 4 attributes of our own 2 entries will yield the appropriate entries being displayed. The current "search-able" terms are:

- Lepidoptera
- Nymphalidae
- Cethosia
- Cyane
- Heliconius
- Ismenius

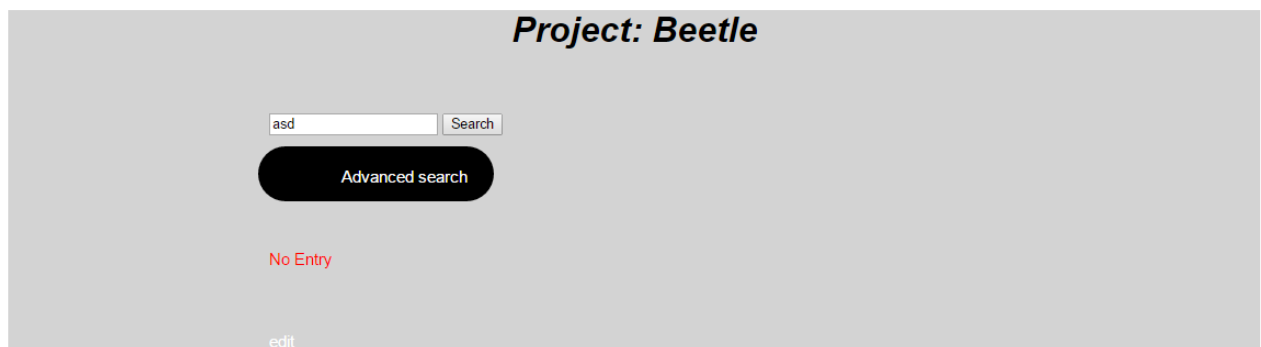
6 Interaction and Design

6.1 Design

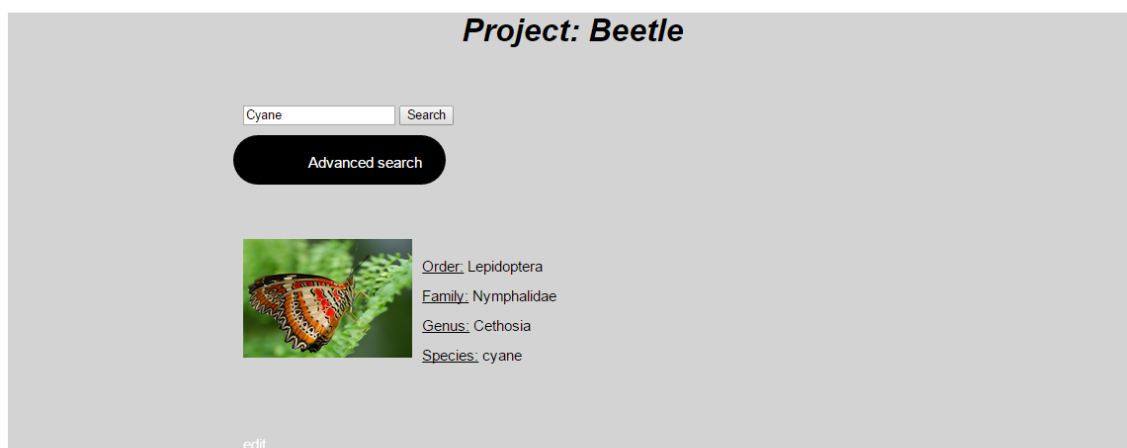
In order to showcase the unit-interface of the product, a series of screenshots have been provided.



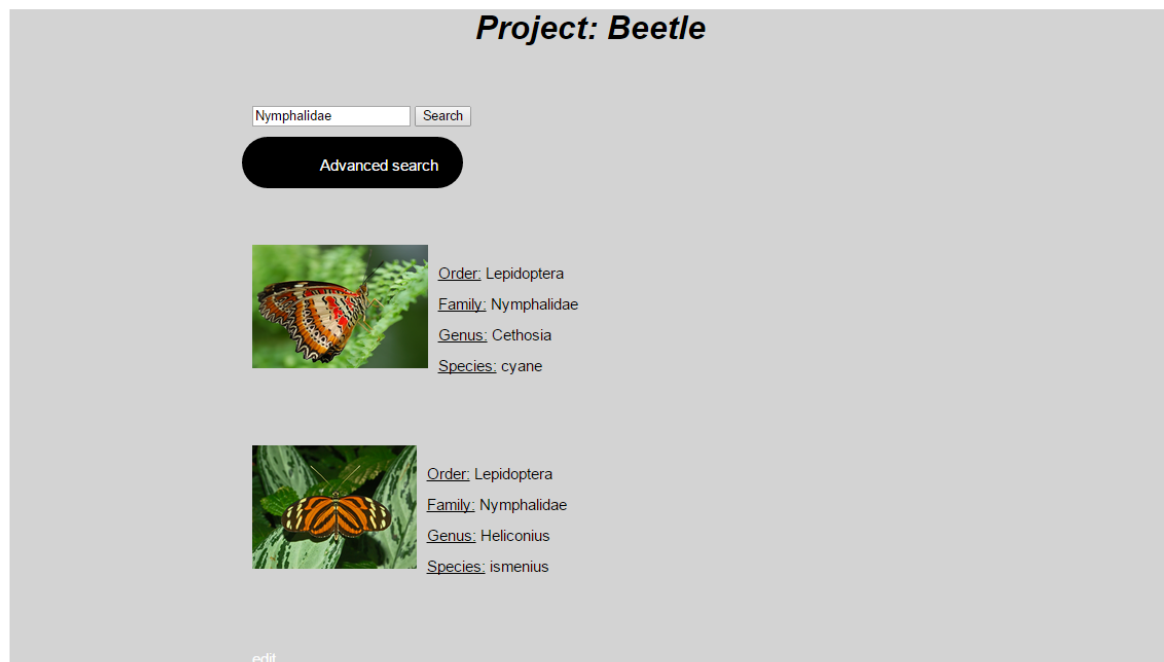
Above is the frontpage of the website before any interaction has been done.



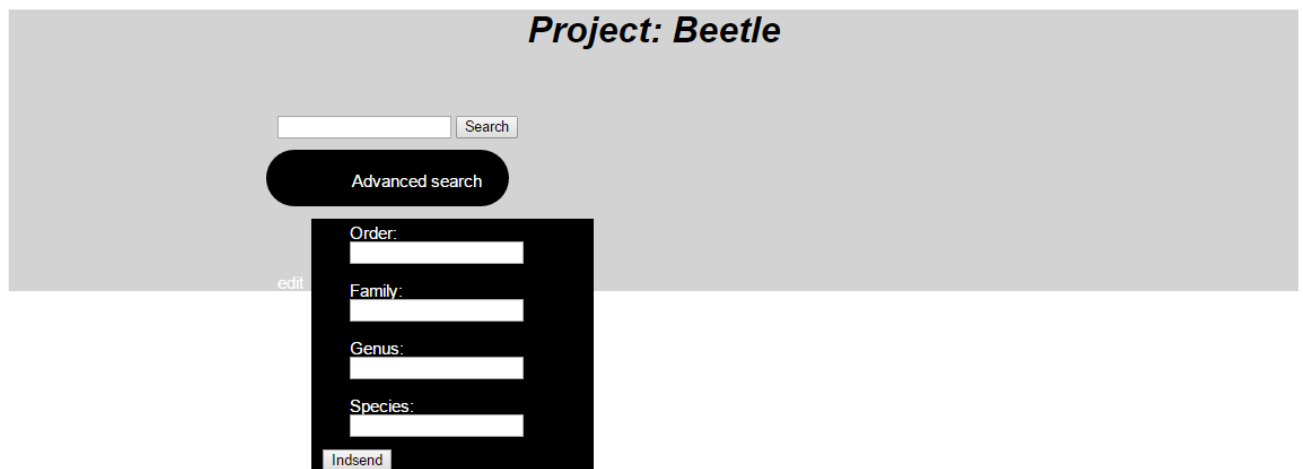
Here is what occurs if no match is found for the input.



The return of one matching entry.



The return of two matching entries.



The advanced search-engine, yet to implemented, shown by hovering your mouse over the button.

7 Internal Cooperation

7.1 Summary

Come thus far, the cooperation of the University of Hamburg is satisfactory.

We have had a total of 3 correspondences with the client, in order to respectively request server-access, request the necessary data for establishing a correct database and to showcase the current state of our product. All the correspondences have been fulfilling and we will be gaining access to the server as soon as their IT-department have created a user for us.

Furthermore, we will be receiving the necessary data for creating the proper database the 22th of April, after which we'll have everything we need.

The client also expressed satisfaction regarding the current state of our product, although a desire to have it looking more akin to the current layout of their website (which is only natural given that this is a prototype).

Our current way of work is based on us setting up workdays and meetings depending on necessity. This means that we, rather than using a gridlocked schedule, have been deciding on days to meet when the need was expressed.

For organisation we have been using a group created in Skype for communication as well as a repository in Github for file-access.

We have been internally communicating the progress of the product along the way and have also internally decided which partials have had the most importance at a given point in time, as to ensure we are all on the same page and that deadlines will be met.

So far we are making great progress in the development of the product itself, and we will soon have all the available means to ensure a succesful deliverance of a working product.

Our effort in organising meetings and working days has been less than ideal and together with our non-gridlock scheduling style have caused us to have many subsequent days of work, causing more of a burden than what is needed.

In order to make our developing from here on more efficient, more emphasis will be placed on proper scheduling, in order to spread out the workload and easen the burden on the entire group. This will also allow more time for quality control.

8 Annexes

8.1 Version Control

8.2 Changelog

8.3 Timeline