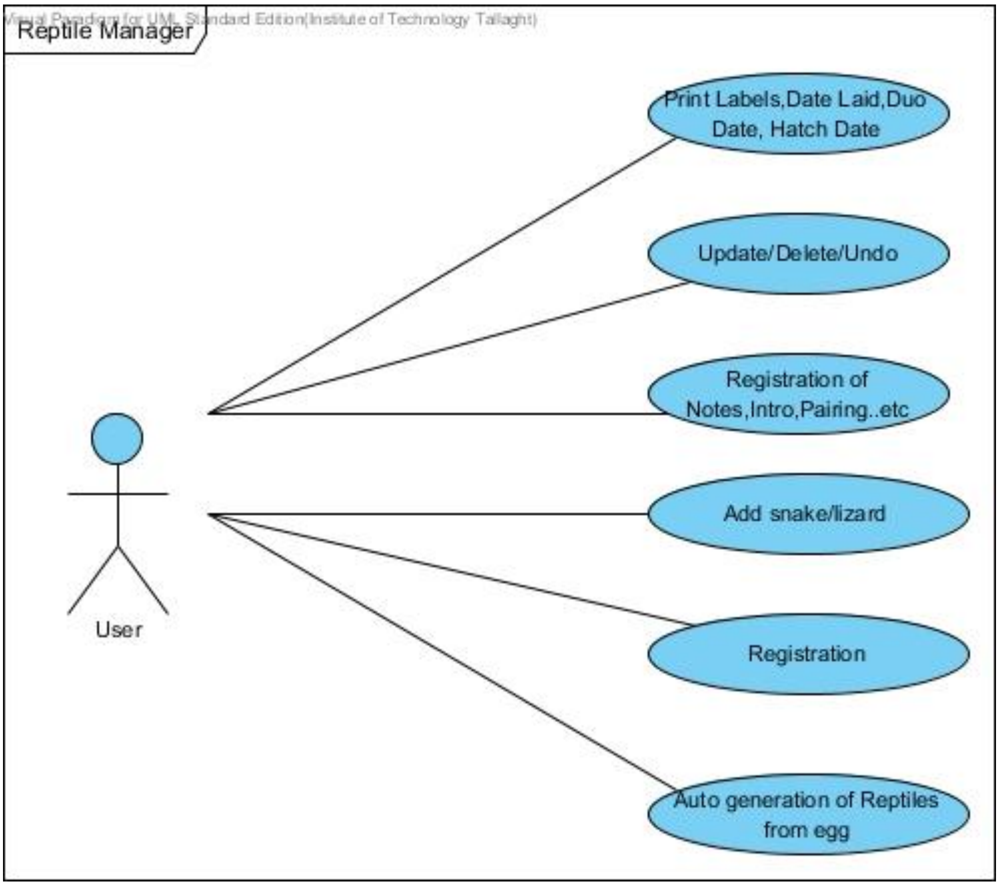
Technical Architecture

Stephen Kenny – x00094666

Use Case


Visual Paradigm for UML Standard Edition(Institute of Technology Tallaght)
Reptile Manager
Print Labels,Date Laid,Duo Date, Hatch Date
Update/Delete/Undo
Registration of Notes,Intro,Pairing..etc
Add snake/lizard
Registration
Auto generation of Reptiles from egg
User

Use Cases

1.

Title	Print Clutch Labels (Clutch is a bunch of eggs), Date Laid, Duo Date, Hatch Date, various labels, cards. PDF labels, cards
Primary Actor	User
Scope	The data will be generated by the user but some will be created automatically from the Database such as hatch date, judging by the species the application will know how long the incubation period will be.
Level	In depth
story	The user can view records in the database and have the option to print off the data.

2.

Title	Update, Delete, Undo
Primary Actor	User
Scope	
Level	Not very complex easy to implement with .NET framework. Undo should be a rollback function within the database.
story	The user can view a record they have the option of undoing the previous action. Update the current record or delete it from the database altogether.

3.

Title	Registration of Notes, Introduction, Pairing, Courting, Separation, Feeding Time, Breeding Time
Primary Actor	User
Scope	Data required will be entered by the user. Daily and or weekly.
Level	Not very complex
story	The user can enter Date and Feeding time in daily or weekly. As for notes if a snake is not looking well a note can be taken for the next user to see and keep track off. Pairing, when two reptile are going to be added together for mating the user selects the pair and they are group together. Courting, this is to keep track and see if the two reptiles have actually mated. Separation, when one reptile is removed this is record and noted.

4.

Title	Add snake/lizard
Primary Actor	User
Scope	A number of fields are required to be filled in, if not they are set as null.
Level	Not complex
story	When a user is adding a new reptile to the database they select the create button and are brought to a new page to fill in all the details.

5.

Title	Registration-This is very broad and covers a lot of functionality. Prey type and Prey size and Prey number. Cleaning, Defecation, Sperm plugs (Only assigned to snakes), Clutch - Egg/ Live Born.
Primary Actor	User
Scope	This is very broad but all is input from the user.
Level	Complex in terms of detail size
story	The user is register all these details about each reptile they have and if they happen to lay or give birth these details are all recorded also.

6.

Title	Auto generation of reptiles from a specific Clutch
Primary Actor	User
Scope	This will be created automatically by the database.
Level	Very complex
story	When a bunch of eggs are marked as hatched, a bunch of new records will be added to the database all auto assigned details when they were born and their gender. The user will have the option to change details or select how many of the clutch hatched out.

Technical Architecture

Software component

Azure

The web application will be hosted on Microsoft Azure cloud platform. Virtual machines will be started to execute the code running on windows servers. These virtual machines however sorted by Azure if updates are needed on them. This can be scaled up if user demand increases. The database will be creating using the relational database-as-a-service on Azure. The database will be created using the Entity Framework because its database creation is more rapid. Although it is not as powerful as OAD.NET development is faster and that’s the main aim. Triggers in the database will be set to send notifications via email or through the web application. The application engine will built using the .NET framework 4.5. Bootstrap is part of the MVC application design and themes can be easily constructed and implemented or a theme can be downloaded for free. C# is being used because it’s the language for .NET. C# is being used because it has LINQ built in. The web app will be all about data manipulation so LINQ will save time and less code will be written. The application will be written in Visual studio because it is integrated with Windows Azure. When the web application is done, launching it on Azure is easy and comes with metrics to see the traffic coming in and out of your application.

Platform libraries

The framework components will be built using the Model-View-Controller model. Microsoft web API 2.2 will be using in conjunction with the MVC to handle JSON and XML. AOD.net will be used at the end of the project to improve database efficiency after Entity Framework has the database up and running. OData will be used to manipulate data for the user using functions like Skip. Web API will use the GET, PUT, POST, MERGE services for updating, adding and removing data from the web service. D3 will be used to create visual graphs showing growth statics of the reptiles. These will be able to be compared to other reptiles to see if that animal is doing well.

Distribution and Deployment

The web application will be hosted on Azure, SSL encryption will be used for logging in with facebook, Google + and user accounts. Cookies will be in place to keep the encryption in place after the user has logged in for tighter security.

Risks

The risks are I am using c# for the first time so it’s a small learning curve. This will slow my development phase down. My visual studio crashes constantly at home leaving me to purely work in college to get my project done. OData and web API 2.2 I have never used, this will be a big learning experience. Reading user manuals and watching tutorials will be most of the work and trying to find ways of implementing the functionality into my project.

Having the database create automatic tables and other complexities will cause problems.