Institute of Technology, Carlow

Computing

Course: BSc (Hons) Software Development Year 4

Author: Eamonn Gaynor

Student ID: C00197458

Tutor: Mr. Nigel Whyte

Document: Functional specification

Plant Disease Identification Application

Institiúid Teicneolaíochta Cheatharlach

INSTITUTE *of* TECHNOLOGY

CARLOW

At the heart of South Leinster

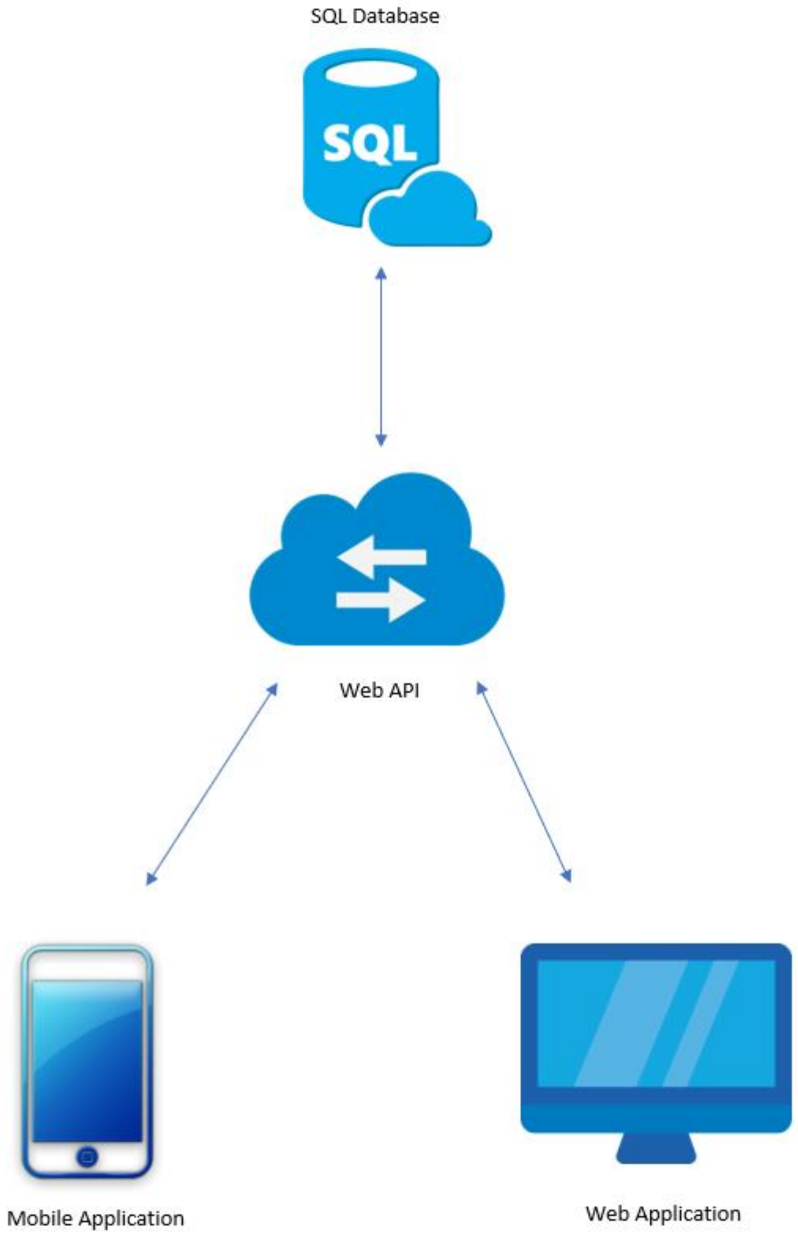
Abstract

The purpose of this functional document is to provide an overview of both the functional and non-functional requirements of the plant disease identification application. The main core of the project is outlined functionally for all components, with a complete iteration plan. The iteration plan will include the definition of each planned iteration, the schedule in detail, identification of the specific activities, timings, and resources.

The aim of this project is to create an application for smart devices which will allow users to identify plant disease through the means of artificial intelligence using computer vision techniques. The goal is to automate visual inspection and eliminate the ambiguity associated with a diagnosis, in turn reducing the subjectivity of the process.

Contents

System Architecture



Project Inspiration

Scientists at EnviroCORE research innovative environmental technologies along with biotechnologies which enhance the economic and social development in an environmentally friendly manner. (Envirocore, 2017). Currently at IT Carlow, scientists at EnviroCORE carry out visual inspections of plants leaves and make subjective opinions on the overall health of the plant. The aim of this project is to create an application for smart devices which will allow users to identify plant disease through the means of artificial intelligence using computer vision techniques.

Plant diseases and pests cause the loss of up to 40 % of the annual harvests, a challenge for a growing world population. (The Crop Site, 2017). Dependable, accurate and precise examinations of plant disease is crucial in predicting yield loss, monitoring disease resistance and preventing epidemic outbreak. Assessments of plant disease which are inaccurate may lead to a misleading diagnosis being drawn from the examination, thus leading to poor disease management and/or loss of the plant itself. (Bock et al., 2010). Smart farming, which is the application of modern Information and Communication Technologies into agriculture, now called the *Third Green Revolution* can have a huge impact on this area of farming, thus reducing the burden for farmers globally. (Smart-AKIS, 2017)

Overview

The overview of this project is to create an application which will allow a user to successfully identify a plant's disease based on an image either taken or uploaded which will be processed using computer vision techniques. The computer vision processing will decipher the images border, for identifying the area in which the leaf is infected, and determine the plants infected area via binary copies. The application will refer this information back to the user, with mitigation plans/ideas and the also the ability to send the data collected to cloud storage for later retrieval by the user. The project consists of these major components:

1. The android application must have extreme ease of use.

2. The application must be capable of performing computer vision techniques.

3. The application should contain a library of associated plant diseases and refer this information to the user.

4. A cloud database should be set up which will allow the user to both store and retrieve data.

5. An administrator page which will allow for the creation of accounts, the maintenance of plant library, report generating, and overall general database management.

Functional Requirements

1. Hardware Requirements

1.1 **Android Device**

An android device will be required to run the application software

2. Software Requirements

2.1 **JAVA**

The application will be written in Java, as Java is the predominant language used for writing applications for android devices.

2.2 **OpenCV**

OpenCV is an open source computer vision library for programming functions, it is used in conjunction C++, Python and JAVA. As this application is essentially being used for visual inspections, it is imperative that the artificial intelligence image processing is of the upmost quality.

2.3 **HTML**

The administration page of the project will be written in basic HTML 5. Hypertext Mark-up Language is the standard language used for writing web applications and web pages, it is predominately used in conjunction with JavaScript and CSS.

2.4 **Database Requirements**

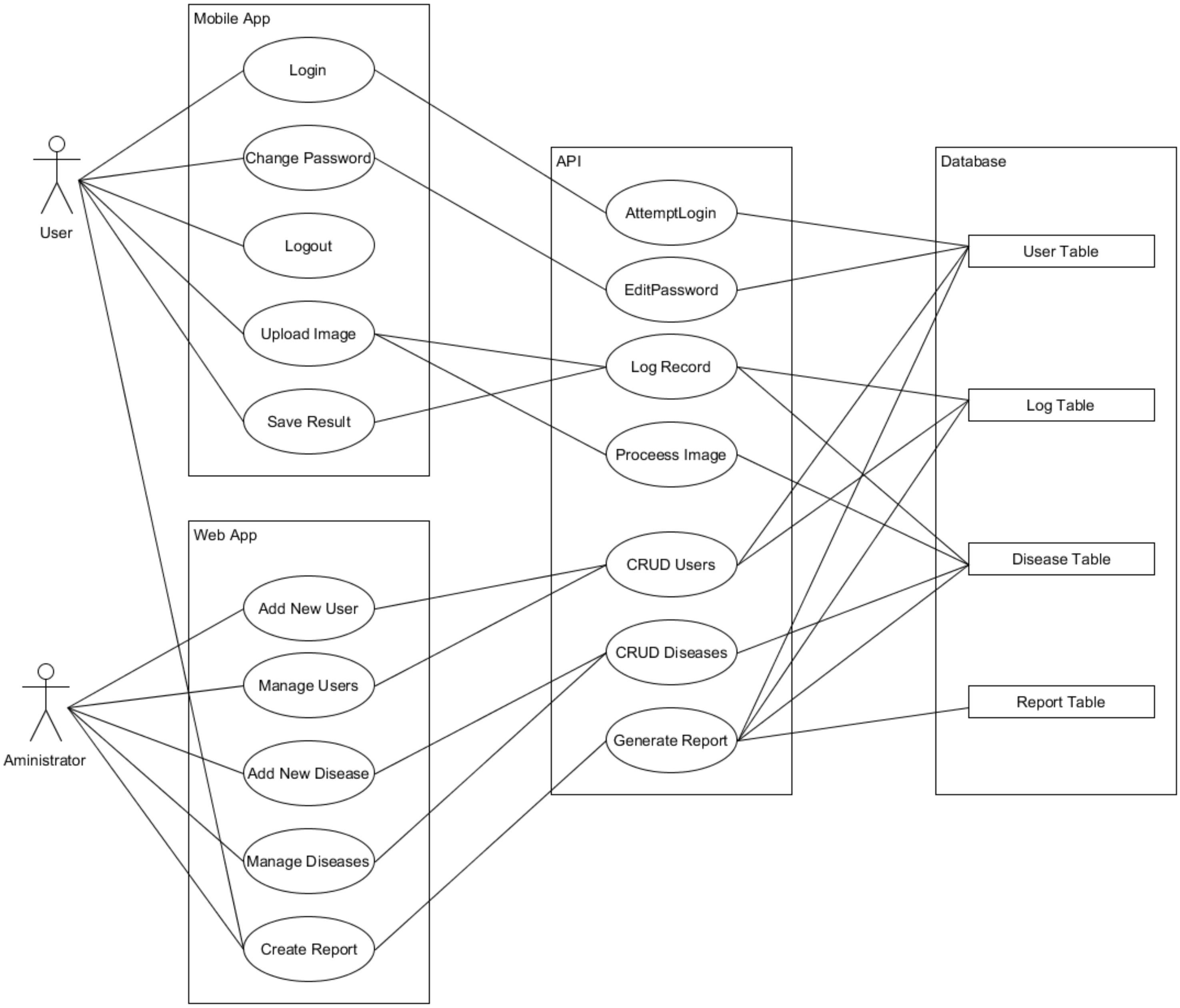
The database must store four tables, as described in the Use Case Diagram section of this report (page 6). These tables include the user table, log table, disease table and the report table. The database must handle query's, for the purpose of the generating reports from the web application.

2.5 **Performance and reliability**

This application will be required to process an image and return the result back to the user within a reasonable timeframe. If the processing takes too long, as in over 10 seconds it will become impractical, anything below 5 seconds will be what is aimed for. High accuracy is also a key objective, as the main goal of this project is to diagnose disease, it is imperative that the accuracy of this application is at least over 90%.

Use Cases

Use Case Diagram



Brief Use Cases

Name	User Login	
Actors	User, Mobile App, API	
Preconditions	The mobile application has successfully loaded, and has presented the login screen to the user. The guest must have already signed up for the application.	
Activity	This begins when a user selects the login function from the main menu. The user enters his/her credentials (Username and password). These credentials are validated by the form and through the API.	
Consequence	The user’s credentials are validated and he/her has now gained entry to the applications user only functions.	

Name	Logout	
Actors	User, Mobile App, API	
Preconditions	The mobile application has successfully logged into the application.	
Activity	This begins when a user selects the logout function from the main menu after they have logged in. The API logs the user out of the session.	
Consequence	The user’s session is terminated, and they are successfully logged out of the application.	

Name	Change Password	
Actors	User, Mobile App, API	
Preconditions	The mobile application has successfully loaded the login page after the user has logged themselves into the application.	
Activity	This begins when a user selects the change password function after logging in. The user enters the old password, and then the new. The API validates this information and confirms.	
Consequence	The user’s credentials are validated, and the password is then changed.	

Name	Upload Image
Actors	User, Mobile App, API
Preconditions	The mobile application has successfully loaded, and a camera is available on the device.
Activity	This begins when a user selects the upload image function from the main menu. An image is taken by the user of a plant leaf, and uploaded to the API. The API performs computer vision techniques on the image, and reflects the collected information back to the user.
Consequence	The user has successfully uploaded an image to the API, and has received back a diagnosis of the plants disease.

Name	Save Result
Actors	User, Mobile App, API
Preconditions	The user has successfully captured an image, and uploaded it to the API to produce a diagnosis.
Activity	This begins when the user selects the save result option from the diagnosis page, once image processing has been performed. The data, image and diagnosis information included, are sent to the API, which in turns sends the data to be stored on a database.
Consequence	The user successfully stores their information in the database.

Name	Add New User
Actors	Administrator, Web Application, API
Preconditions	The admin has successfully launched the web application and logged in.
Activity	This begins when the administrator selects the add new user function from the main menu of the web application. The admin enters the credentials of the user, including privileges. The web application cross checks this information and validates it against the database for errors such as duplicates.
Consequence	A new user is successfully added to the system by the administrator.

Name	Manage Users	
Actors	Administrator, Web Application, API	
Preconditions	The admin has successfully launched the web application and logged in.	
Activity	This begins when the administrator selects the manage users function from the main menu. From here, users can be modified or deleted. Modifications including names and privileges. The information is validated with the system for errors and then updated to the database.	
Consequence	The administrator has successfully modified the user table.	

Name	Add New Disease	
Actors	Administrator, Web Application, API	
Preconditions	The admin has successfully launched the web application and logged in.	
Activity	This begins when the administrator selects the Add New Disease function from the main page of the web application. From here, the admin can then fill in the form for adding a new disease to the database. Information such as disease name, type, description will be added. The system will then validate this information for any errors that may occur, e.g. prevent duplicates.	
Consequence	The administrator has successfully a disease to the disease table.	

Name	Manage Diseases	
Actors	Administrator, Web Application, API	
Preconditions	The admin has successfully launched the web application and logged in.	
Activity	This begins when the administrator selects the Manage Diseases function from the main page of the web application. From here, the admin can then modify or delete the information regarding the current diseases located in the disease library. Information such as disease name, type, description can be modified. The system will then validate this information for any errors that may occur, e.g. prevent duplicates.	
Consequence	The administrator has successfully modified a disease in the disease table.	

Name	Create Report	
Actors	User, Administrator, Web Application, API	
Preconditions	The user or admin has successfully launched the web application and logged in.	
Activity	This begins when the administrator or user selects the Create Report function from the main page of the web application. From here, the user or admin can then create a report based on the information stored in the database. The admin can access all information, but the user may only access the information that is in relation their account. A report will be generated from selections and presented to the user on submit. The system will process these selections in correlation with the databases.	
Consequence	The administrator or user has successfully created a report from the database.	

Iteration One

The initial iteration, iteration one of the project began in October 2017. The projects initial specification was explored through discussion and critical thinking, identifying the core technologies, functions and designs. This was done by researching prior applications that may have had similar objectives to that of this project. The initial technology chosen to identify the disease was that of OpenCV, a C++ inbuilt library for computer vision. This library was explored through writing different image processing techniques such as Histograms, Fuzzy C-means, Canny Edge detection etc. These example programs were written in C++ through Eclipse using the loaded OpenCV library. However, after further discussion of the specification, for the purpose of this project where the different colours of the leaf were the main focal points, the more practical option would be to use the API which was would be written in PHP. Within the PHP language, there is a built-in function called 'colouratindex', which retrieves an index value of pixel coordinates which can then be converted to RGB values. What also drove the decision to move away from OpenCV was that the mobile application was written in JAVA, making it difficult to incorporate OpenCV within the application as it is written predominantly in C++. A MySQL database was chosen as the storage option for the data, with the images uploaded being stored on the root directory of the server.

Once the initial languages, core functions, design features had been identified, building had commenced. For the first iteration I had chosen to start on the mobile application. Applying key design techniques and building what will be the core functionality of the application as a whole. The login, register, choose photo functions were also built. Where the user may register an account on the server, from there logging in under those details. The image captured was displayed on screen, from this point I had created what was a starting foundation from which I could build upon.

Conclusion

In conclusion, the functional specification outlines core functionalities of the disease identification application. These functionalities are outlined clearly through the use of use case diagrams, brief use cases, functional requirements and a full iteration plan. By adhering to these specifications and plans, it will enable the project to successfully build and implement the application.