Industrial Team Project Report

Team 1
Kari McMahon, Mark Goddard, Robert Mason, Ewan Mount and Zhihua Liu
AC41004 - University Of Dundee

Abstract

From September 15th 2014 to October 3rd 2014, 8 teams worked on developing a diagnostic phone application to aid the field diagnosis of potato diseases for the James Hutton Institute's Malawi potato project as part of the industrial team project module. This report focuses on the software development lifecycle undertaken by team one whilst developing the application. This report outlines various aspects of the lifecycle such as the need for the application, requirement gathering, design, implementation and evaluation of the application. As well as focus on the team's selection of the agile development process and the necessary project planning/overseeing that goes along with the process. The team uses this report to reflect on various aspects of the project, the success and challenges as well as lessons learnt from the project.

Introduction

The aim of the project is to develop a diagnostic phone application to aid the field diagnosis of potato diseases for the James Hutton Institute's Malawi potato project. The length of the project was 3 weeks and was undertaken by a team of 5 (Kari McMahon, Mark Goddard, Robert Mason, Ewan Mount and Zhihua Liu).

Background

Professor Lesley Torrance from the James Hutton Institute in Invergowrie approached the School of Computing at the University Of Dundee about the possibility of building a diagnostic phone application to aid the field diagnosis of potato diseases. This project was given to the fourth year students as their brief for the industrial team project module.

Professor Lesley Torrance is the leader of the cell and molecular science group at the James Hutton Institute and Professor of Biology at the University of St Andrews. Professor Lesley Torrance and her team at the institute have been working on a Scottish government supported potato project in Malawi. The "project aims to contribute to poverty reduction and food security through strengthening the development of sustainable potato production and marketing systems for improved productivity and trade." (James Hutton Institute, n.d.)

Malawi is a country where 40% of the households cultivate and is the 7th biggest consumer of potatoes in the world. Potatoes are an important crop in Malawi but suffer greatly from pests, diseases, potato agronomy and lack of storage (James Hutton Institute, 2014). Professor Torrance and her team at the institute feel farmers could benefit greatly from a low cost infield diagnostic tool, which would help identify a variation of problems in their potato fields at the early stages before it starts to spread that could cost farmers money and vital crops. The diagnostic tool will be in the form of a phone application and will be integrated into the James Hutton Institute Malawi potato project in hopes to further strengthen sustainable potato production. The potato project has had many successes so far such as renovation and upgrading facilities for mini tuber production, the introduction and evaluation of new cultivars and capacity building and the team hope this application will also be a success for the project.

In terms of similar products there are few out there which are iPhone applications such as Nutrient Deficiency and potato pests by Tsor. The client told us about these applications but said they felt more like a book or were not ideal. The client said she hoped for an application which was more image oriented so if you did not know what disease you were looking for you could compare images in the app click on the image and it would direct you to the disease. In terms of Android apps the team could not really find anything from research, the only one we could find was sweetpotato diagnotes which is an application to aid sweet potato disease diagnosis and again this application was text oriented and not what the client was looking for.

Specification

The requirements for the project were gathered in a number of ways. The main method for the requirement gathering was the initial client meeting on day one of the project. Additional methods, included a scout trip to the James Hutton Institute, email communications with the client and a number of other meetings during the three weeks.

The initial client meeting was a presentation to the class. The client gave the team's background on the Malawi potato project, the need for the application and what the requirements are. After the meeting the team go together and generated these requirements from notes into concise user stories.

Requirements:

1. As a stakeholder, I want an image recognition system to identify potato diseases to aid in the diagnosis of potato diseases.
2. As a stakeholder, I want a searchable glossary of symptoms, which link to a further information page about the symptoms.
3. As a stakeholder, I want to field tests videos to help farmers understand how to do these tests.
4. As a stakeholder, I want to be able to zoom into images to aid diagnosis.
5. As a stakeholder, I would like to be able to share images taken from the application to aid diagnosis.

Non-functional requirements:

1. As a stakeholder, the application should work off-line as it will be used out in the field where Internet connection will not be available.
2. As a stakeholder, the application should be scalable as not all pests and diseases will be covered.
3. As a stakeholder, the application should conserve battery power.

Technology:

1. Android and iPhone (Not necessary to do both).

With these requirements the team agreed that they did not think developing image recognition was possible within the time frame. From this we came up with an alternate solution, which would be an NHS 24 like expert system where the user would answer certain questions about symptoms and it would give the user the disease relating to these symptoms. The team planned on asking the client if this requirement would be suitable at the scout trip to the James Hutton Institute.

The trip to the James Hutton Institute constituted of a team member from each of the 8 teams. The selected team member for our trip was Kari McMahon. At the meeting the scouts were given an hour-long presentation on what Professor Torrance's team works on at the James Hutton Institute. We were also shown how to do the FTA and LFD test and could participate in this if we wanted too. We also had the opportunity to ask professor Torrance any questions about the application.

Kari voiced her concerns to Professor Torrance about the ability to develop image recognition, where there appeared to be a misunderstanding. When Torrance spoke about image recognition she meant the user looked at the images for the symptom and compared it to the plant on the field. This description already fitted the team's current second requirement in the requirements, which is a glossary that leads to a further information page where you can compare images of plants. Kari also asked professor Torrance about the expert system, which she liked the idea of. Professor Torrance also said during the visit that a user should be able to update the application. During the trip some images were taken while recording the videos for the application and the plant images can be viewed in the appendix.

After the meeting the team wrote up a new set of requirements (user stories). Then emailed these to Professor Torrance to check these were correct who then confirmed the requirements. The email can be seen in the appendix. The final requirement specification is below[1]:

Requirements listed in priority:

1. As a stakeholder, I want a searchable glossary for leaf, nutrient, pest and tuber symptoms which when a symptom is selected leads to a further information page for each symptom that contains six images, basic facts, diagnostics and control information.
2. As a stakeholder I want to field tests videos to help farmers understand how to do these tests.
3. As a stakeholder, I want a decision support tool that will allow the user to compare symptoms of plants growing in the field that allow me to make decision about what may be causing my problem and then guide to other pages with information on what to do about it.
4. As a stakeholder, I want to be able to zoom into images to aid with diagnosis.
5. As a stakeholder, I want to be able to share images taken from the application to aid diagnosis.
6. As a stakeholder, I want to be to update the data in the application.

Non-functional requirements:

1. As a stakeholder the application should work offline as it will be used out in the field where Internet connection will not be available.
2. As a stakeholder the application should be scalable as not all pests and diseases will be covered.
3. As a stakeholder the application should conserve battery power.

[1] One of the requirements changed in priority after the email communication as we felt the videos were higher priority over the expert system based on the email reply and trip to the institute

Technology:

1. Android and iPhone (Not necessary to do both).

The requirements are managed on Trello and can be found at the following address:

https://trello.com/b/kE5Tl8AA/industrial-team-project

The requirements are written as user stories as the agile specification suggests. The reasoning behind the agile approach is because the project has short time constraint to produce a working solution that meets client's expectations, appropriate testing & evaluation as well as the necessary documentation to go with it.

The agile approach allowed the team to take the highest priority requirements they believe they can get done in a time frame of usually a week. Taking the agile approach means we could achieve all the key aspects of the application whilst being realistic and not over estimating which could lead to a disappointed client.

The team told professor Torrance we would be taking the agile approach and we planned on achieving the top two requirements and if we had time we would try to achieve the rest. Professor Torrance agreed to this approach, which can be seen in the email in the appendix.

Due to the short time period of the project and not always being able to have access to the client as they were in Invergowrie. The team could not take a full agile approach and had to schedule agile meetings around pre-set times in the timetable. This meant the approach we took in terms of the sprint was not fully agile and sometimes felt closer to an iterative approach.

Our work schedule plan for the project was:

- Planning stage – Mon 15th Sept to Wed 17th of Sept 2014.
- Sprint 1 – Wed 17th Sept to Wed 24th Sept 2014.
- Client Meeting - Wed 24th Sept at 4pm where we review our application with client.
- Sprint 2 - Thurs 25th Sept to Thurs 2nd Oct 2014.
- Client Meeting – Mon 29th September at 9am where we would give a progress report.
- Presentation workshops Mon 29th September and Tue 30th September.
- Scheduled user testing / evaluation Wednesday the 1st of October and Thursday the 2nd of October.
- Final presentation on the Friday 3rd of October as well as project deliverables to be handed in.

As well as the schedule above the team also meets with Professor Arnott for our managerial meetings twice a week and our Kari, our team leader meets individually with Professor Arnott once a week. In an agile project often you do not select a team leader but in this scenario we had to select a team leader, which was Kari McMahon. Kari would project manage and organise the team during this project. See appendix for sprint backlog.

Design

Once the team had the specifications for the project several design decisions had to be made.

Device selection

The first design decision tackled was whether to develop the application for Android, iPhone or both. The client had made it clear to the teams that she would like the application to be on both devices but that the teams did not necessarily have to do both.

To develop for iPhone, developers must have access to a mac which not all members in the team have hence this would mean if team members wanted to work from home they would not be able to which was a disadvantage. The language used to develop iPhone applications is objective C, which no one in the team had any experience with and it is said to have quite a steep learning curve. With around two and half weeks to develop a working prototype to the client we felt it would be best to develop for a device where the team already had some development experience and the time could be focussed on building the project and not having to take time out to learn a new language.

The team all had experience with development on Android and already had the development kits set up on their laptops. This made Android an appealing choice because we could spend more time developing rather than trying to organise setting up the development tools and learning the quirks of a new language.

The team also conducted some research into phone usage in Malawi and found that Airtel Malawi the telecommunications market leader in Malawi were the main provider with a market share of 75% (Airtel Malawi, n.d.). The team researched into phone prices in Malawi using figures from Airtel Malawi (Airtel Malawi, n.d.), and Cellular Abroad (Cellular Abroad, n.d.). The prices can be seen below:

Android phone prices:

HUAWEI IDEOS, £46
HUAWEI Ascend Y100, £67.27
Samsung Galaxy S7562, £159.5

iPhone prices:

Apple iPhone4, £245.5
Apple iPhone4s, £307
Apple iPhone5, £390

From these values we can see that Android devices are a lot cheaper than the iPhone. We can therefore say that an Android device would be more cost effective for this usage hence the team felt it would be much more appropriate to develop for Android. There would always be the possibility in extending it to iOS in the future. We tried to find out the percentage of people who use Android devices and iPhone devices in Malawi but were unable to find these figures. Professor Arnott spoke to Professor Torrance who had an email discussion with an academic in Malawi, who said the most popular phone in use is the Samsung Galaxy which reassured our team it would be most suitable to produce an application for Android as they were available at reasonable prices and must be used by a significant amount of people, if the most popular phone is an Android device.

There was one other option that enabled us to develop for both devices which, was a development tool like PhoneGap or LiveCode, which allows you to develop applications using HTML, CSS and JavaScript and then export the application to both devices. This seemed a good idea and appealed to the team but unfortunately the team had no experience in developing with these tools and again we were concerned how quickly we would be able to pick up an understanding of developing in a system like PhoneGap or LiveCode and create something substantial within such a short time period for the project. In the teams minds it was more important to build a strong application than fiddle with new technologies.

After analysing all the points discussed above the team decided that developing purely for Android had the most benefits for this project.

Development Tools Decision

At the design stage the team had to decide on the development tools we would use to develop the project on. Most of the team already had the Android development kit on their laptops and used either Eclipse or Intellji as there IDE depending on each team member's preference.

In terms of versioning control we decided to use Github as all of the team apart from Liu had experience with git and already had it installed and setup. Kari also had a private account on Github, which meant she could set up the group git to be private which also aided the decision to use this tool. The Github can be found here https://github.com/karivmcmahon/IndustrialTeamProj. Although the team had some reservations about using Git because it can be tricky to get used to and it takes a while to understand how to resolve conflicts it is a good way to manage a project. Git gives a log of the full project, you can see how much a team member is committing, and you can revert back to an old working version if there is an issue with the current version. Git enables you to not only commit code to it but also accepts a variation of files so the team felt it was the right choice for storing all the files for the project.

For testing the application the team used the emulator that comes with Android SDK and two members of the team had Android phones, which were also used for testing and also to demonstrate to the client and to the manager at managerial meetings.

During the design stage the team also thought it would be a good idea to integrate unit testing into the application to help keep the code maintainable and help tease out any errors that may occur in the code which the team may have not thought about. The team chose to use the Android JUnit tests as it's built in with Android development kit and it is designed to be tested on Android devices. The team also have experience with Android JUnit tests so it seemed like the most appropriate choice.

The team were aware we might have time to build a website for updating the application. So the team spoke with Mahamadou about getting webspace and asked what type of website the server could handle. He gave us two choices ASP.net/C# or PHP. Mark is unable to install the required software to develop for ASP.net on his computer, which meant ASP.net would not be a good option as he wouldn't have been able to contribute. The team decided if we were to build a website we would use PHP to query the SQLite server side database. No one on the team had much experience with PHP so we were aware this might take some time to learn. The team also decided we would use bootstrap to aid building the website as Mark has had experience with bootstrap and it is known to be time-effective when needing to mock up a website within tight time constraints.

Database Design

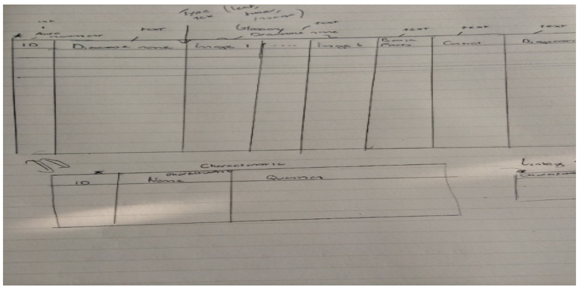
After the teams decision of developing on Android we needed to decide on the type of database the device would use to store the potato disease information. The client made clear she would like the application to work offline but also to be scalable.

For the application to work offline we decided to use the in phone SQLite database to store the information for the glossary. We planned on developing the application with the in phone database so the application could work offline as the requirement to have a glossary of symptoms and videos of the test were the highest requirements and the ones we promised to achieve.

The team designed a solution for this, which was an admin portal, which would be implemented if the team had time or could be implemented in the future. The solution was a website where the user could update the information to a server version of the SQLite database and then when the phone had an Internet connection it would sync the in phone database with the server database and update the application. Hence also making the phone scalable and work offline.

SQLite database was chosen for the reason that Android comes with a built in SQLite database implementation It uses an application file format, making it easily accessible and cross platform, which is useful for implementing syncing between the server and the android database. It also has a very small code footprint and makes efficient use of memory, disk space and disk bandwidth. This is useful as not all phones can cope with databases with large amounts of memory. SQLite also claims to support terabyte-sized databases and gigabyte-sized strings and blobs, which makes this database choice useful for scalability (SQLite , n.d.). SQLite supports all the relational database features and it is basically a lightweight version of MySQL, which shouldn't cause any problems seeing as the team have come from a MySQL background.

Below is a sketch of the database design we believed we would need for the application. We believed we needed a glossary table, which would contain information about the disease such as its name, type (tuber, leaf, insect), 6 images (as text references to the images in the folder), basic facts, control and diagnostics. The second table is a characteristic table, which would contain the characteristics of each disease and a question for this characteristic for the expert system. These two tables will have a linking table they will be linked by the IDs of each. A sketch of the database design is shown below and a larger image can be found in the appendix.

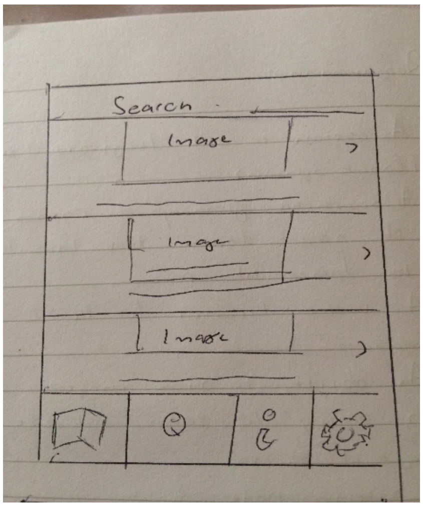


Sketches

Below are the finalised design sketches of the application the team came up with from the requirements. The sketches were used as an aid to the team when developing the application. Larger versions of the sketches can be found in the appendix.

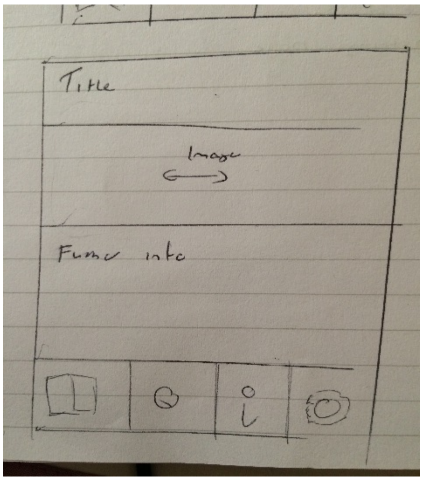
Glossary Page Sketch

The glossary page would show the name of the disease and an image relating to it when clicked on takes you to the further info page.



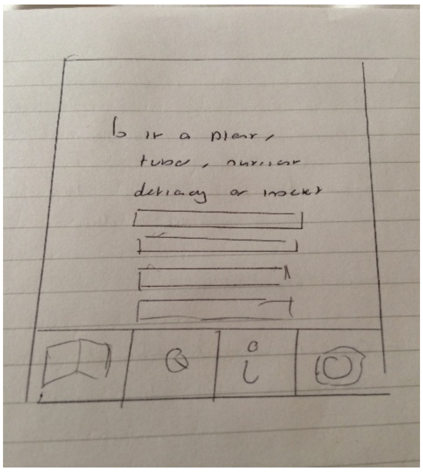
Further Information Page Sketch

The further information page shows the user more details about the disease such as a flick able slideshow and details such as basic facts, control and diagnostics



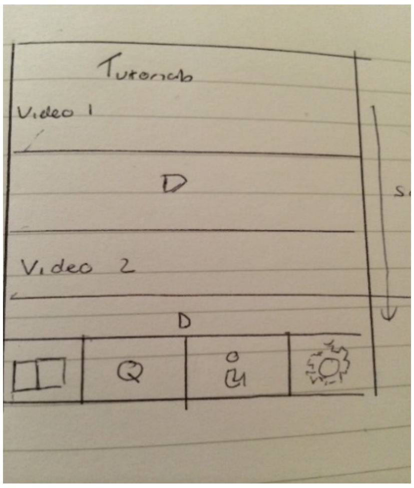
Expert System Page Sketch

The expert system page would have questions, which you would answer and it would then lead to a diagnosis of which disease it thinks it is.



Video Tutorials Page Sketch

The video tutorials page shows the user videos of the two field tests.



Handling Images within the App

One of the biggest design concerns that we had was the way in which were going to deal with images in the app. Storing images, especially large images take up a lot of memory and we realised that this may cause problems where a phone does not have the required memory to process the images.

As a group we came up with three possible solutions on how we would store the images effectively, they are as follows:

1. Storing images in the DB as blobs. This solution could cause memory issues depending on the size of images we are retrieving and processing.
2. Storing the images on internal memory and save path to image in the DB.
3. Storing images in the Android drawable folder. Images won't however, be able to be dynamically updated.

We decided that the best solution was to store the images as blobs as this would allow us to update the images from an external server. We did however, have concerns about how the database would perform with about 120 images stored and said that we would have to test that we can retrieve and process a number of images at the same time.

Before we started implementing this solution we decided that it would be best, that we started off storing images in the drawable folder until we had out external database and android database setup.

Project Planning

At the design stage in the project the team discussed several ways to manage the project. The first was to minute the daily meetings. The minutes would summarise what happened at the meeting as well as the tasks achieved that day or from the previous day and tasks to do before the next meeting.

The plans for the daily meetings were from 9am – 5pm where we would work on the project in the labs. The reason for these meetings was to stick to the Agile approach of working 9am – 5pm in the same environment as each other as it should make it easier to communicate changes or updates in the project, show progress and make sure the project is on track. The team also agreed at the start of every meeting the team would have a quick ten-minute SCRUM where each member would say what they did the day before, what they were going to do and any obstacles that they had encountered. This is an Agile technique which helps all the members in the team know where each other is at in the project and try to quickly solve any obstacles that may be causing issues for team members. The minutes would document information from the scrum and any other details from the rest of the meeting throughout the day. The majority of the minutes throughout the project were documented by Kari and can be found in the appendix.

Often with team projects there tends to be more risk involved than with an individual project where you are just managing yourself. During the project planning and design stage of this project the team developed a risk assessment, which all members of the team could access on the Github, which stated the biggest risks to the project and two preventions to deal with the risks. This was so that if any of the risks occurred the members of the team would know how to manage the risk before it had detrimental effects on the project.

As stated above the team decided on using Trello to manage the project requirements as it is flexible to move requirements around and edit them, it's easy to colour code the requirements in terms of what's to be done, in progress and finished and it is online so all team members can access Trello wherever they are. To manage the requirements selected for a sprint the team choose to use a sprint backlog which outlines the dates of the sprints, the requirements being done and tasks related to this requirement, how long each task will take and who's been doing them. The sprint backlog enables the team to see if the project is on track, if team members are over allocated and under allocated on tasks as well as a good way to list additional tasks that don't relate to the project but are also deliverables like the user guide, final report etc. The sprint backlog was the main tool to aid project planning throughout the implementation stages. The team decided on the use of the backlog as we all had seen the benefits of using one when taking the agile module in third year. The sprint backlogs can be found in the appendix.

The session with the careers team on personality profiling was also useful in terms of understanding where our skills lie within the team. At the end of the session Kari and Ewan were given a cool blue profile and Robert and Liu were given a fiery red profile. Unfortunately Mark did not make it to the personality profiling session so the team did not know what colour he was. From the session the team were aware they were missing a yellow and green personality and would have to be mindful of this throughout the project and make an effort to bring out their green and yellow personality skills, which may sometimes be hidden. The session made us aware of the benefits and challenges of having each colour in our team enabling us to be mindful of how to deal with each personality type within our team which would aid the team in project planning and managing. Below is an image of our team from this session.

As stated in the specifications the team developed an Agile like work schedule around the pre-set meetings with the client. We planned on the schedule below during the planning and design stage:

- Planning stage – Mon 15th Sept to Wed 17th of Sept 2014.
- Sprint 1 – Wed 17th Sept to Wed 24th Sept 2014.
- Client Meeting - Wed 24th Sept at 4pm where we review our application with client.
- Sprint 2 - Thurs 25th Sept to Thurs 2nd Oct 2014.
- Client Meeting – Mon 29th September at 9am where we would give a progress report.
- Presentation workshops Mon 29th September and Tue 30th September.
- Scheduled user testing / evaluation Wednesday the 1st of October and Thursday the 2nd of October.

The schedule above was designed/planned to enable us to build some functionality, gain feedback from the client then from this feedback build/change functionality over the space of the week where we are able to contact the client about changes over email. Then with the new functionalities to gain some finalised evaluation of the application from users in time for the final presentation of the product to the client at the end of the project.

Ethics

At the design stage of the project the team wrote an ethics form which outlined the teams plan to do user testing to gain understanding of the usability of our application and if the users find the application clear and easy to understand. As well as doing user testing the team also planned on doing an evaluation of our application against another application on the market for certain tasks to see if our application is more effective for aiding diagnosis of potato disease. The study will involve filling out a quick demographic form, the user testing or evaluation task then filling out a SUS form on the usability of the applications. This will help us understand the usability of the application and evaluating if our application has been successful. In the appendix you can find the ethics form along with the demographic sheet, consent form and SUS usability form.

Implementation

As stated earlier the team plan for development was split over two sprints.

In sprint 1 we agreed with the client we would achieve these two requirements. Confirmation of this agreement can be seen in emails in appendix:

1. As a stakeholder, I want a searchable glossary for leaf, nutrient pests and tubers symptoms which when a symptom is selected leads to a further information page for each symptom that contains six images, basic facts, diagnostics and control information.

2. As a stakeholder, I want to two field tests videos to help farmers understand how to do these tests.

Sprint 1 was quite slow as it took the team a while to get back to grips with developing with Java on Android. Over the course of the sprint the team achieved both the requirements set with the client and made a start on the expert system (the decision tool requirement).

During the course of the sprint the team came across several technical hurdles often due to new development experiences or having to manage storage limits on the devices. Examples of these hurdles was in the teams design sketch where there was a sketch of a flick-able slideshow viewer which would be on the further information page which would show various images of the disease to help aid diagnosis. The slideshow was a challenge to build as we had to use viewpager to build it which none of the team had experience with prior hence it took a bit longer to get to grips with and to style the viewpager the way we wanted it. Then when the image from the slideshow is clicked it opens up a dialog that shows the image in full-size with a close button. The team found it difficult to get an image to fit the dialog exactly and to get the dialog to dynamically change size based on the image's size. In the end we solved the issue by using some simple mathematics to get the dialog to be dynamic. Developing a custom list view had a similar issue to the viewpager where the team had little experience with using it and took a while to get the exact design of the image, name and an arrow on each row of the listview.

The team struggled with managing the storage limits on the device. The client gave the team a folder of images and videos to go on the application along with text data. The size of the original folder was 88 MB once downloaded which the team were aware would be far too large for the application to store. Our initial solution to the issue of image and video size was to compress the images and video and then see if the quality was still of an acceptable standard for diagnosis. We used compression tools called Caesium and Handbrake which compressed the images to 25 MB and the videos together to around 5 MB and were still of reasonable quality. The images were being stored in the drawable folder and the database stored the path of each. With the compressed images in the folder they were working well when loaded on the emulator but then when placed on the phone, if images were more than 500kb were called would cause the phone to crash. At this point we had to add some code into the application to scale down the images before they were loaded into the application and this solved the issue as well as making the length and width of the images smaller.

The videos within the application did not cause any issues once compressed. They still played well and were of a good quality. The only issue we struggled with was when the videos were compressed on handbrake they are saved as H.264 profile main which android does not play and we discovered the compression needs to be saved as H.264 baseline for the video to play android. It took the team a while to recognise the issue. The design in the sketches for the video player page was that it would have the title of the video and then below would be the video. In the application having this layout meant the scrollview kept scrolling to the middle as it was focussing on the second video, which the team found difficult to fix and in the end had to fix the scrollview to the top of the page. Also if the user played both videos and scrolled it would merge each of the videos media controls, which would was confusing and looked messy. So the team decided to hide media controls when the user was scrolling. Although we had these technical hurdles in sprint 1 we managed to achieve both the requirements we agreed with the client and started an additional requirement in this sprint, which was an achievement.

In the application the team used an SQLite database to store the majority of information for the application. An SQLite database is stored in the application which enables the phone to access data even when offline which meets the clients requirements. The majority of the team have never worked with SQLite and although it is similar query wise to SQL, writing the queries in Android is quite different and takes some getting used to. To aid development, maintainability and code readability we used the MVC model. We separated the SQLite logic into the model section of the application and any aspects that relate to the view are in the view application. These are in separate packages within the application. In some cases people hardcode their SQLite database into their application if it is a small database but our application will be storing a reasonable amount of data with the possibility to extend in the future. We decided to build the sqlite application in an sqlite datebase browser called Navicat where we could export the database file. We would then take the exported file save it in the android's asset folder and then have code to copy the file in the asset folder into the application. The team choose this option because it was easier to replace the database with the file than having to change the hardcoded file and it also makes the application more flexible.

In the sprint we used some development techniques such as unit testing using Android JUnit tests. The majority of the application is GUI based hence hard to test but since the application is communicating with a database we developed a tests to check if we were retrieving data correctly from the database which would quickly help diagnose if an issue in the application was being cause by the database. We wanted to use unit tests because they help you think about scenarios that may cause issues, which you wouldn't otherwise think of which makes your application more stable. It is also good to get into the habit of running unit tests every day to help check everything is working as it should. As well as unit testing we also tried to use the agile practices of refactoring and pair programming as much as we could. Pair programming was particularly useful as we had Liu on our team who is a guest and it was a great way to show him how we work and to have him also contribute to the work. Pair programming was also useful in situations where we came up against a technical hurdle that needed to be resolved before anyone could continue with development.

In terms of changes from the sketches during implementation there really wasn't too many. Originally we had the tabs at the bottom of the application in the sketches but when using the application it was hard to scroll through the information with tabs at the bottom and we were accidently clicking them so we moved the tabs to the top, which we believed worked better for the readability of the application. The style of the listview for the glossary page also changed for the sketch, in the sketch the image and text are centred in the middle with an arrow at the right hand side. When implementing this design it did not look as professional in the application so we decided to move the image to the left hand side, text centred in the middle, aligned with the middle of the image and an arrow on the right hand side. The images on the tabs also changed based on what icons were available under the correct licenses.

On Wednesday the 24th of September we had a short meeting scheduled with the client, which we treated as our sprint review as on that Wednesday we finished the sprint. We showed the client the two requirements we completed as well as the one we started that wasn't agreed for that sprint. In general the client was happy with the application. In the application there is a listview with categories like leaf, which then takes you to another listview listing all the leaf diseases and an image relating to them. The client asked if she could just have it image oriented without text. She was happy with the image and video quality. We suggested the ability to search for a disease and it immediately takes you to the related page and the ability to pinch zoom on the full size image which she agreed she would like that and the team agreed to achieve it for next sprint. At the meeting we recognised she struggled clicking the close button on the full size image dialog, which the team recognised and needed to be improved. The client also wanted to discuss the ability to update the application we told her this would be a complex process, which we did not think we could build within the time frame of the next sprint. The team told the client we could build a mock website that would show how the application would be updated and could be built upon in the future and if we had time in the sprint we would try to get it to actually update. From this point in the meeting we agreed the requirements to be agreed for the next sprint.

In sprint 2 we agreed with the client we would achieve these three requirements. Confirmation of this can also be seen in the emails in the appendix:

1. Changes asked for based on the client meeting.
2. As a stakeholder, I want to be able to zoom into images to aid diagnosis.
3. As a stakeholder, I want a decision support tool that will allow me to compare symptoms of plants growing in the field that allow me to make decision about what may be causing my problem and then guide to other pages with info on what to do about it.

As well as the requirements above we also agreed with the client we would develop a mock website which will either update the applications offline website or will have the ability to have this feature built in at some point in the future if the client decided to choose the application.

After the client meeting on the final day of the first sprint Kari managed to implement the pinch zoom requirement which she had started looking into during sprint 1 meaning we completed one of the sprint 2 agreed requirements within sprint 1. It was difficult to implement this requirement as Kari was working on the task and doesn't own an Android phone and you could only test the pinch zoom on an Android device hence only being able to test when around other team members with Android devices. The team then found an emulator called Genymotion which has pinch zoom built in meaning Kari could test the zoom from the laptop. We wanted to implement the zoom because it would be easier to see aspects of the image and aid diagnosis.

During the course of sprint 2 the team came across more technical challenges. For the website we had to build the website in PHP for reasons discussed in design decisions. The team have little experience with PHP so it took some getting used to and you cannot compile PHP script to find errors so sometimes it is hard to spot minor mistakes like missing semi-colons. In general the website was fine to build the only issues was the PHP queries communicating with SQLite.

The team implemented an expert system where the user would answer several questions about certain symptoms, which would then narrow the answer down to a specific disease, which it would then lead you to the disease it diagnosed. The expert system is similar to an NHS 24 system. The team felt during implementation that a lot of work could still be implemented into the expert system such as images displayed that relates to the symptom and displaying a list of other possible diseases as well as the one the system narrowed it down to. Overall the expert system currently implemented shows how the team wants it to work but definitely could be built upon and improved in the future.

In sprint 2 the team promised the client that the team would be able to build a mock website with the ability to update the application. The reasoning for this was time constraints and we did not think we would be able to implement full updating functionality within the time of the second sprint. The team managed to get syncing of the application for the insert and update of the website. In the website you can update only the text information and not the images at the moment and in the website you can insert both the text and image information.

Syncing works through this process. The application stores a shared preference where the application was last updated. When the user presses sync in the settings feature of the application it sends the date it was last updated and the current date to a php script called sync. Sync then returns the rows that were added after the last updated date till the current date. It returns these rows via JSON, which the android application then retrieves the data from and adds to database if it doesn't exist or updates if it does exist.

The team handles images in the php by retrieving image and encoding to base 64. The base 64 is then added to JSON. When Android retrieves the JSON it decodes it to a byte array, which can then be inserted into the database as a blob. In sprint 1 the images were stored as drawables with a reference to the path of the image in the database but now they are being stored as blobs in the database.

In sprint 2 again the application did not change much for the sketches. The only addition is the settings page contains the sync button. Originally the team did not sketch the settings page because the team were unsure about what should be in it. In sprint 2 we also added a search bar as requested by the client which was not originally in the sketches but made the application more user friendly as you could direct to the correct disease page if you wanted to gain some information.

A video user guide was created to show how to use the application and website developed over the two sprints this can be found in the appendix. The source code developed can also be found in the appendix.

The team managed the development through daily meetings and a sprint backlog. The sprint backlog shows the requirements, tasks relating to each requirement, which member is carrying out that task and the estimated time to complete. The sprint backlog was vital to help manage the project as the team could see all tasks that were still to be done or in progress. The sprint backlogs for each sprint can also be found in the appendix.

Throughout the project the team had to be aware to always maintain professionalism. This involved making sure all the images used within the application had the correct rights and licences, that we keep all information given to us by the client confidential, maintain ethics for user testing and reference all code used that was developed by someone else. In the appendix lists the rights to the images being used.

The implementation over the two sprints came with its challenges but the team managed to achieve all the requirements agreed with the client.

Evaluation

As stated earlier in the report the team submitted an ethics form for doing user testing and an evaluation, which will be attached to the report. Originally the team planned on doing user testing and an evaluation of our application against an application on the market, which, was submitted, to ethics. Due to time constraints the team decided to just do user testing. The procedure for the user testing for evaluation is as below:

1. Participant will be given an information sheet and invited to read it.
2. Participants will be given a consent form.
3. Participants will sign consent form. Participants will be given a demographic questionnaire.
4. Participants will be introduced to the system, which is demonstrated by the experimenter.
5. Participants will be asked to user testing for around 20 minutes, which, will be without too much direction so we can learn about the usability of the system and how clear it is.
6. Participants will be given a System Usability Scale evaluation form.
7. Once tasks are done, participants are free to go.

The aim for participants for the user testing was 2-4 people preferably from the James Hutton Institute. The participants should be from the ages of 18 – 60 and the gender split should be even.

The team asked the people from the James Hutton Institute if they were available to user test. We were able to organise testing with two people from the Institute, which was a very valuable experience.

The rest of participants were people who were interested in our application and participating in the study.

The results from the demographic are below:

ID No	Gender	Age	Exp with potato apps	Exp with tech
1	Male	21	No	Good
2	Male	20	No	Good
3	Male	21	No	Good
4	Male	20	No	Not bad
5	Female	48	No	Not bad
6	Male	49	No	Not bad
7	Male	23	No	Good
8	Male	50	Yes	Not bad
9	Female	21	No	Not band

One participant did not fill out a demographic form but did fill out a SUS due to time constraints at user testing.

To evaluate the application we noted what was said during the user testing when they looked at different aspects of the app and asked them to fill out a System Usability Scale form. The system usability scale is a “quick and dirty, reliable tool for measuring the usability” with benefits such as it’s an industry standard, it’s valid and can be used on small sample with reliable results. (usability.gov). The average SUS score for our evaluation is 81.3 which gives our application an A grade from the 10 participants that took in our study. An average of an A grade shows that application to be very strong as any score above 68 is considered above average and can be interpreted as a very usable application. The full SUS results can be found in a document in the appendix.

The comments made during testing were participants found it clear, they would use it again and the zoom image is handy. Improvements suggested by users were having when the description mentions a test having a link to that test in the video and making the search more sophisticated for example searching based on symptom.

Overall the results from our study are really good. The SUS scores show we have a very usable application and comments made show we have a good application that still has room to be built on. The team believe the evaluation shows the application has strong potential.

Over the course of the project other forms of evaluation took place. Client evaluation took place three times. The first client evaluation took place on the 24th of September on the final day of the team’s first sprint. The team gained good feedback from the client with minor changes being asked for such as could the glossary page, which was originally a listview be more image oriented and remove the name of the disease next to the image. This then resulted in the application having a gridview to display the glossary, which was more gallery based and image oriented. In the meeting the client made clear she thought pinch zoom would be a good feature and this was implemented after the meeting. In the meeting the team observed the client had issues closing the zoomed images so they enlarged the buttons for this. The second client meeting occurred on the 29th of September where the team showed the progress being made during sprint 2. In general the client had no negative feedback from the meeting and made some suggestions about the search bar, which we made to the application. On Wednesday 1st September the client spoke to one of the members of the team when they were undertaking user testing. He showed her the most recent version of the application and again she had no negative feedback to give but some suggestions for the future of the application to build on.

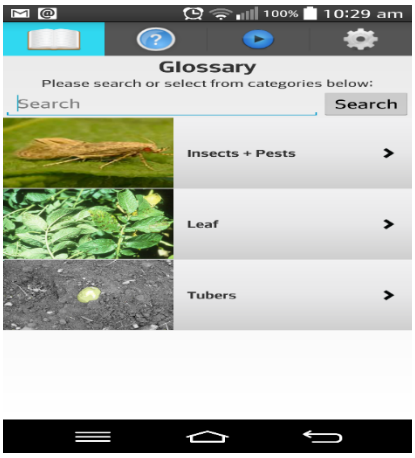
The application didn’t change too much from the original sketches in the majority of cases. Design changes to the application came from client feedback for example the client suggested a header in the glossary page just explaining it’s the glossary page or the change from listview to gridview to make the application more image oriented as suggested by the client. Suggestions like this caused changes from the original design in the application.

In the non-functional requirements the client stated they didn’t want the application to use much battery and after using the application over a certain amount of time on an Android device we have found the application uses a minute amount of battery, which meets the client’s requirements. The application also works well offline, which meets another non-functional requirement and in terms of sunlight affecting the quality of the images really depends on the device screen quality and brightness on the phone and is out with our control. In terms of scalability the use of the website to update/insert into the application it means the application will also be scalable although this does have some areas to be built on but does meet the client’s non-functional requirement of scalability.

From the evaluations we can see that we have tried our hardest to build the application to the client's expectations and requirements based on feedback over the 3 weeks. We have managed to meet not only the functional requirements but also the non-functional requirements for the application. We received strong feedback from our user-testing showing the application is usable, clear and easy to understand showing it has clear potential for the future.

Final product

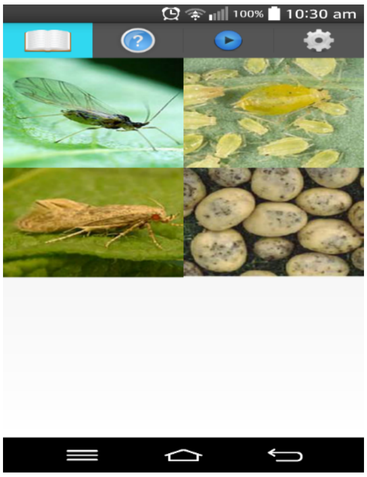
The final product is an android application and a website. The application has 4 tabs at the top of the application, which represents glossary, expert system, video and settings.



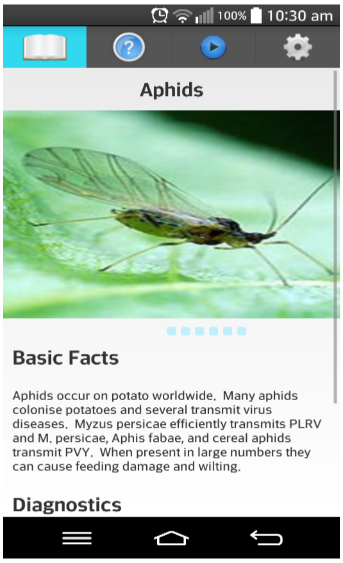
Glossary

Please search or select from categories below:

Within the glossary tab is a fragment where you can search a disease or view diseases by type. If you search a disease it will return all diseases relating to the name in a listview and then if you click on the item it takes you to an information page about this disease. If you search by type it will take you to a gridview of images from certain leaf diseases and then if you click on the image it will also take you to the further information page.



The further information page displays information about the disease and a slideshow showing images of the disease, which you can click on. If you click on an image it will show you a full size version of the image, which you can then pinch zoom in on.



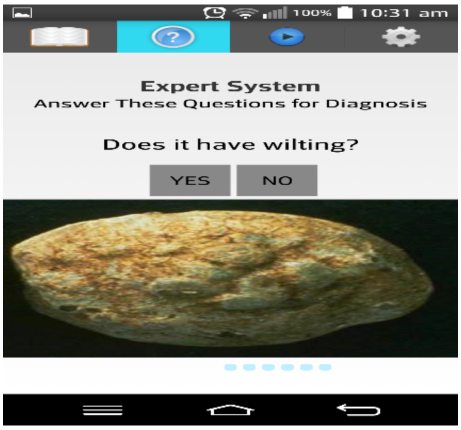
Aphids

Basic Facts

Aphids occur on potato worldwide. Many aphids colonise potatoes and several transmit virus diseases. Myzus persicae efficiently transmits PLRV and M. persicae, Aphis fabae, and cereal aphids transmit PVY. When present in large numbers they can cause feeding damage and wilting.

Diagnostics

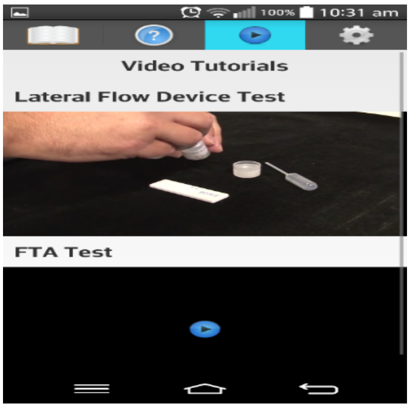
Within the expert system tab is a fragment where you can answer questions about the symptoms you have and it will diagnose the disease and then take you to the diagnosed disease's further information page.



Expert System

Answer These Questions for Diagnosis

Does it have wilting?



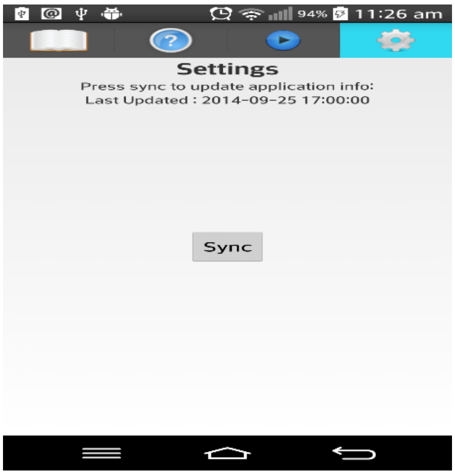
Video Tutorials

Lateral Flow Device Test

FTA Test

Within the video tab is a fragment, which contains the two videos the client asked to be displayed.

Within the settings tab contains a button, which enables the app to sync with the server side database.



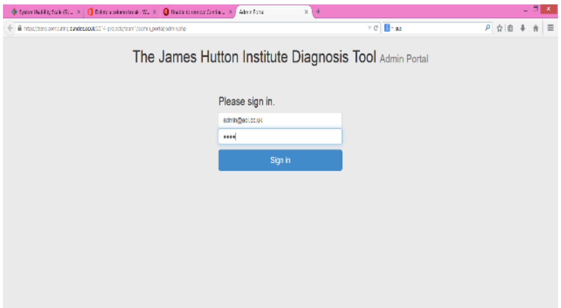
Settings

Press sync to update application info:
Last Updated : 2014-09-25 17:00:00

Sync

The website is stored at this location https://zeno.computing.dundee.ac.uk/2014-projects/team1/admin_portal/admin.php and the login for the application is username: admin@aol.co.uk and the password is pass.

Once you have passed the login page you can view the current data in the database. The user can select one of the rows of data and it will take you to a detailed view of the data. At this point you can edit the data you are viewing and update it. Then when the user presses sync in the android app they should then have the updated data.

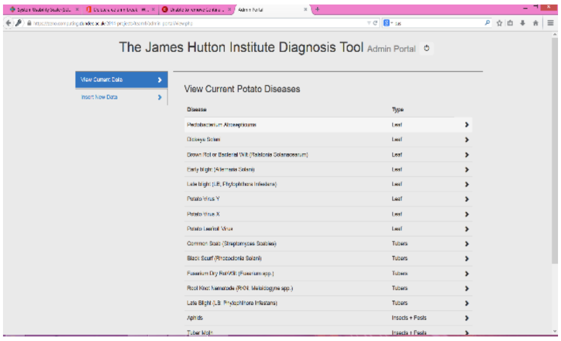


The James Hutton Institute Diagnosis Tool Admin Portal

Please sign in.

admin@aol.co.uk

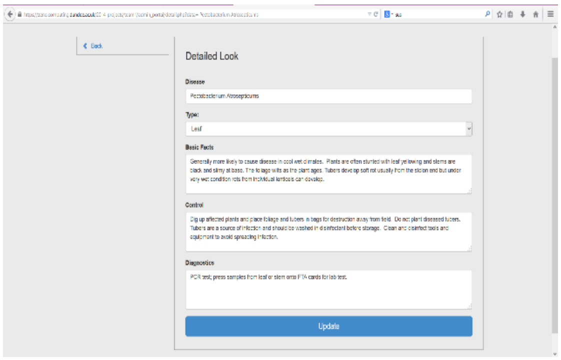
Sign in



The James Hutton Institute Diagnosis Tool Admin Portal

View Current Potato Diseases

Disease	Type
Pectobacterium Atrosepticums	Leaf
Dickeya Solani	Leaf
Brown Rot or Bacterial Wilt (Ralstonia Solanacearum)	Leaf
Early blight (Alternaria Solani)	Leaf
Late blight (LB; Phytophthora Infestans)	Leaf
Potato Virus Y	Leaf
Potato Virus X	Leaf
Potato Leafroll Virus	Leaf
Common Scab (Streptomyces Scabies)	Tubers
Black Scurf (Rhizoctonia Solani)	Tubers
Fusarium Dry Rot/Wilt (Fusarium spp.)	Tubers
Root Knot Nematode (RKN; Meloidogyne spp.)	Tubers
Late Blight (LB; Phytophthora Infestans)	Tubers
Aphids	Insects + Pests
Tuber Moth	Insects + Pests



Detailed Look

Disease

Pectobacterium Atrosepticums

Type:

Leaf

Basic Facts

Generally more likely to cause disease in cool wet climates. Plants are often stunted with leaf yellowing and stems are black and slimy at base. The foliage wilts as the plant ages. Tubers develop soft rot usually from the stolon end but under very wet condition rots from individual lenticels can develop.

Control

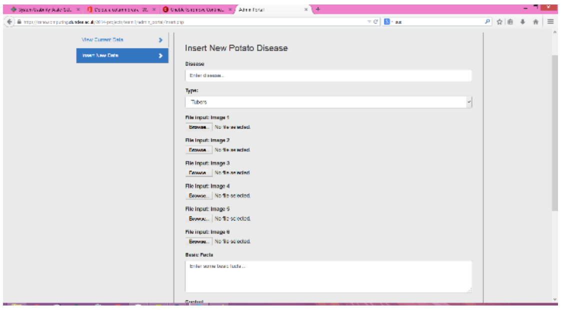
Dig up affected plants and place foliage and tubers in bags for destruction away from field. Do not plant diseased tubers. Tubers are a source of infection and should be washed in disinfectant before storage. Clean and disinfect tools and equipment to avoid spreading infection.

Diagnostics

PCR test; press samples from leaf or stem onto FTA cards for lab test.

Update

You can also insert new data by going back to the view page and on the left hand side pressing insert new data, which will give an area to insert new data into the application and again when you press sync in the application you will retrieve the new information.



Insert New Potato Disease

Disease

Enter disease...

Type:

Tubers

File input: Image 1
Browse... No file selected.

File input: Image 2
Browse... No file selected.

File input: Image 3
Browse... No file selected.

File input: Image 4
Browse... No file selected.

File input: Image 5
Browse... No file selected.

File input: Image 6
Browse... No file selected.

Basic Facts

Enter some basic facts...

Control

More detailed images can be found in the appendix.

Critical Appraisal

In terms of design decisions made over the course of the project the majority were successful. The team felt the choice of using PHP server side was good because it was easy to encode JSON with an image to send into the application, which was a concern we had in the design stage. The team also felt the choice of working with Android was the best for the reasons stated during design decisions and the fact that time constraints for the project were tight. Our sketches from the design were also accurate in the majority of cases and generally look similar to how we envisioned the application, which was good as it showed we understood the client's needs.

In terms of implementation the team learnt a lot from our two sprints. We worked with a lot of new aspects of android we had never worked with before such as fragments, handling a large amount of images, sqlite and handling videos. Although we learnt a lot from using these new techniques for this project we needed some skills in handling compression and learnt a lot from the experience although a lot of time was wasted in the project handling memory out of bounds error and perhaps we could have benefitted from being taught about image/video handling in android before the project. Our main issue with implementation was technical hiccups like memory out of bound issues which we had to deal with which, took quite a bit of time to resolve and ate into time to build other features or work on documentation.

The agile method we feel was a good choice to use as we were able to prioritise the core features of the application for the client. Through this method we achieved all the set requirements for each sprint meaning we did not disappoint the client and reached the agreements made. Completing a number sprints allowed us to get feedback for core features as before our second client meeting we had completed the first sprint this enabled our application to be representative of what the client wanted. We had quite a few meetings with the client where we always acted on the feedback given to make the application better. The team believes we worked well on meeting the client's needs and the majority of feedback was always good.

The user testing was a success and showed our application to be a strong and usable product. The evaluation was great way to learn if the application we built was useful and find out possible improvements for the future.

Project management techniques such as the sprint backlog, risk assessment and minutes were all useful in understanding where the team was at, how to handle issues and what tasks still needed to be done. The team feel as though we all benefitted from these techniques. During the project we had some project management issues often with team members sometimes running late, not communicating issues they were having in development or not turning up when needed. We tried to resolve these issues as much as possible through sprint retrospectives and at group meetings or through messages on our Facebook group. We tried to handle these issues as much as possible and voice concerns at managerial meetings if there was any.

A challenge within the product was trying to create a fully agile project where we had lots of documentation to develop and in an agile method there really should be little documentation. In Agile you also should have access to the client quite frequently and arrange meetings around your sprints this was difficult because we had pre-set times with the client which did not always work with the way we organised the sprint. The time for the project did not work well with the way agile projects should be organised which meant the style of doing Agile had to be changed slightly.

As a team we learnt the difficulties in a project when you do not always have the client close by and you need to produce something within a short time period especially if you are using the agile method. Although this has been a challenge it has given us industrial experience of what it might be like within an industry environment.

Due to the short time constraints in the project you do not have much time to plan the project before you need to start implementing. Over the course of the project the team came up with other solutions for the application in terms of database design or image storing which we didn't have time to implement but would have been interesting to explore. Although we have created a solution, which, we believe, works well, works offline and is scalable. In some aspects of the application there could be possible improvement or re-design due to more knowledge gained throughout the project. For example it was challenging to designing a database for the project when we didn't understand much about the topic and didn't have the best understanding of how the application would all work together especially with the website. When it got to the final week and we had spoken to the client, we realised we could have probably designed the database better. Perhaps if we had more time with the client and more time on the project we would have recognised the solutions earlier. Although, if the team continued to develop the project these solutions could also be implemented.

Summary & Conclusion

Overall we believe the project to have been a success we gained a lot of new skills, met the client's requirements set for both sprints and gained strong feedback on the current application. The project has given us a realistic understanding of what it is like to work in an industry environment where you may not have access to a client and when your team is random with a mix of different abilities. The lessons learnt and skills taken away from this project will benefit the team members greatly for future team projects. We believe participating in a project has given us invaluable experience of what it is like work in an industry environment and has benefitted the team significantly.

Future

On a whole we believe the application is a strong product which meets the client expectations but there are several aspects of the project that if we had time we would like to improve on. For example the website we built for the application was built as a mock website to show the client how they would update/insert to the application. We would like to improve the design of the website and possibly be able to delete information from the application through the website depending on if you have the correct admin rights to do so although this would need to be discussed with the client. We feel the website could be built upon quite a bit. Another aspect which could be improved on is the expert system currently the expert system shows images with the question which does not always match we would like to improve this but it would involve a change in database design which we did not have time for in the project. We would like to work with the client to build a better expert system as we did not have much time with the client and we built it just based on the information given. In the website/application you can insert text/images from the website to the application but in the update you can only update the text. In the future we would like to be able to update images in the application as well as view the images in the webpage. Within the application there is quite a few aspects we would like to improve on due to the short time constraints we just didn't get everything we wanted implemented in the project. As well as improvement we also believe there is a lot to build on in the application with the aid on Professor Torrance and her team which would be a possibility for the future of the application.

References:

Airtel Malawi, n.d. Airtel Malawi Facebook. [Online]
Available at:
https://www.facebook.com/pages/Airtel-Malawi/146777668735018
[Accessed 21 September 2014].

Airtel Malawi, n.d. Devices. [Online]
Available at:
http://africa.airtel.com/wps/wcm/connect/africarevamp/Malawi/home/personal/devices
[Accessed 21 September 2014].

Cellular Abroad, n.d. SIM & Cell Phone Packages For Malawi. [Online]
Available at:
http://www.cellularabroad.com/packages-malawi.html
[Accessed 21 September 2014].

James Hutton Institute, 2014. Specification For Project, s.l.: s.n.

James Hutton Institute, n.d. Staff - Lesley Torrance. [Online]
[Accessed 28 September 2014].

SQLite , n.d. SQLite Features. [Online]
Available at: http://www.sqlite.org/features.html
[Accessed 21 September 2014].