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 disease for the James Hutton Institute’s Malawi potato project as part of the industrial team project module. The report focusses on the software development lifecycle undertaken by team 1 while developing the application. The 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 the 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 disease 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, 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 disease. 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 and could cost farmers money and their 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 team hope this application will also be successful.

**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 1 of the project. The additional methods for requirement gathering was the scout trip to the James Hutton institute and email communications with the client.

The initial client meeting was a presentation to the class. The client gave the teams background on the Malawi potato project, why the need the application and what the requirements are. After the meeting the team met and generated these requirements from notes.

Requirements:

1. As a stakeholder I want an image recognition system to identify potato diseases to aid the diagnosis of potato disease.
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 two 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 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.

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, were shown how to do the FTA and LFD test and could participate in this if they wanted too and the opportunity to ask professor Torrance any questions about the application.

Kari asked professor Torrance voiced her concerns 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 their requirement gathering which is a glossary that leads to a further information page where they can compare images of plants. Kari also asked professor Torrance about the expert system which she liked the idea of and gained a new requirement from professor Torrance which was the user should be able to update the application. During the trip some images were taken while recording the videos for the application and the plants these images can be seen in the appendix.

After the meeting the team wrote up a new set of requirements (user stories) document. 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]](#footnote-1):

Requirements listed in priority:

1. As a stakeholder I want a searchable glossary for leaf, nutrient pests and tubers symptoms which when a sympton 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.
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
4. As a stakeholder I want to be able to zoom into images to aid diagnosis
5. As a stakeholder I want like 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.

Technology:

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

The requirements are managed on Trello and can be found here:

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

The requirements are written as user stories as the team used the Agile approach to software development. The reasoning behind the Agile approach is because the project has a very short time constraint to produce a working solution to meet client’s expectations, test and evaluate the application as well as the necessary documentation to go with it. The agile approach allows the team to take the highest priority requirements they believe they can get done in a time frame of usually a week. This means the client should get the aspects of the application they really want and the developers are being realistic with the client and not over estimating or claiming to get all the requirements done and hence disappointing the client.

The team told professor Torrance we would be taking the Agile approach and we planned on achieving the top two requirements done 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 are 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 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.

**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 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 has any experience with and is said to be quite a steep learning curve. With around 2 and a 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 all 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 did research into phone usage in Malawi and found that Airtel Malawi a telecommunications company the market leader in Malawi 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(Provide by Airtel Malawi) |
| 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 iPhone’s so if a farmer or the institute was to purchase a phone to be used primarily for this application. An Android device would be more cost effective for this usage hence the team felt it would be better to develop for Android for the project and extend it the iPhone in the future instead of the other way around. 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 at the moment who said the most popular phone in use is Samsung Galaxy which reassured our team it would be suitable to produce an application for Android as they were available at reasonable price 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 allow 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 get 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 type of 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 so the team kept it this way and decided the two IDE’s we would use were eclipse or intellji depending on team members preference.

In terms of versioning control we decided to use Github as all of the team apart from Liu all had some experience with git and already had in installed so we felt it would be best to stick with a versioning control system that the team had installed and were familiar with. Kari also has 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 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 while developing aspects of the project and Robert and Ewan both have an Android phones so the team will use these for usability testing, checking features look the same on the phone as well as device and using the phone 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 choose to use the Android Junit tests as it’s built in with Android development kit and are specifically 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 a webspace and asked what type of website we could put within this webspace. He gave us two choices ASP.net/C# or PHP. Mark is unable to install 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 has much experience with PHP so were aware this may 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 is known to be time-effective when needing to mock up a website within tight time constraints.

**Database Design**

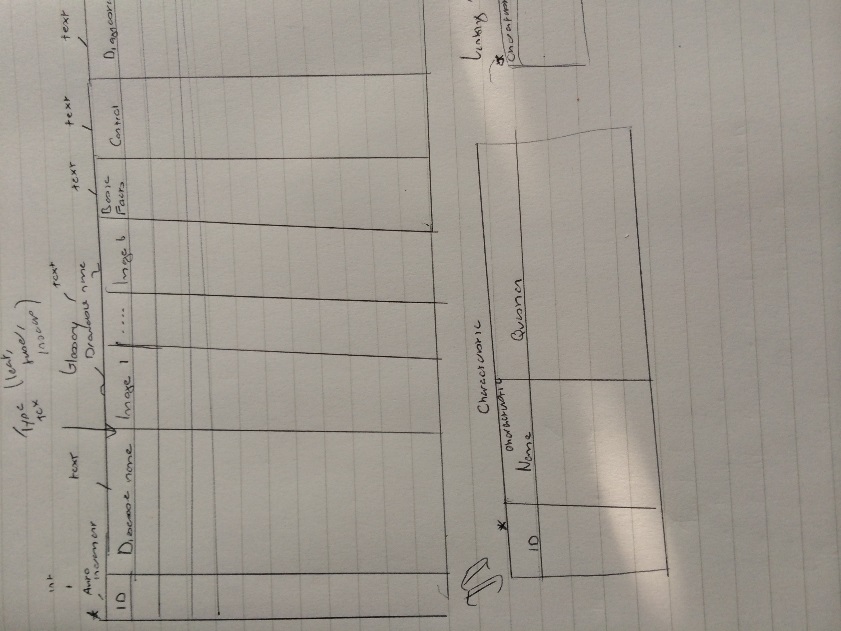
After the teams decision of developing on Android we needed to decide on the type of database the device would use to store the potato disease information. The client made clear she would like the application to work offline but also to be scalable.

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.

Since there was also a requirement to be able to make the application scalable and updatable. The team designed a solution for this which would be implemented if the team had time in the project 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 it is the database Android uses to store information within an application enabling the application to work offline. A SQLite database uses an application file format so it more easily accessible and cross platform which is useful if the feature to implement syncing between the server and in phone databases. It also has a very small code footprint and makes efficient use of memory, disk space and disk bandwith which is useful to be used in a phone application as they cannot cope with databases which take up a 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.). The only issue with the choice of using SQLite as out database is only two members of the team have experience with this database so we were aware there may be a learning curve for some members of the team.

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.

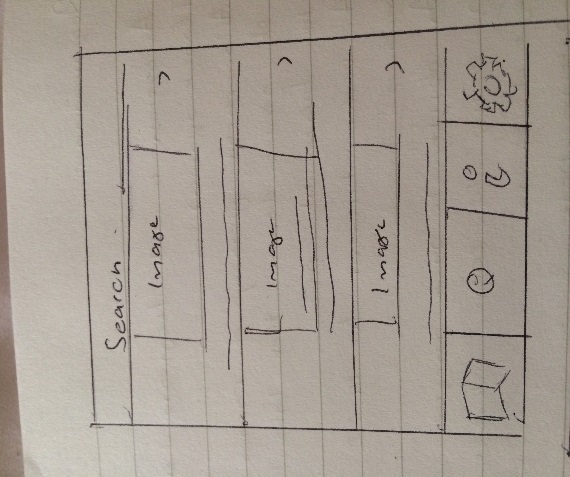


**Sketches**

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.

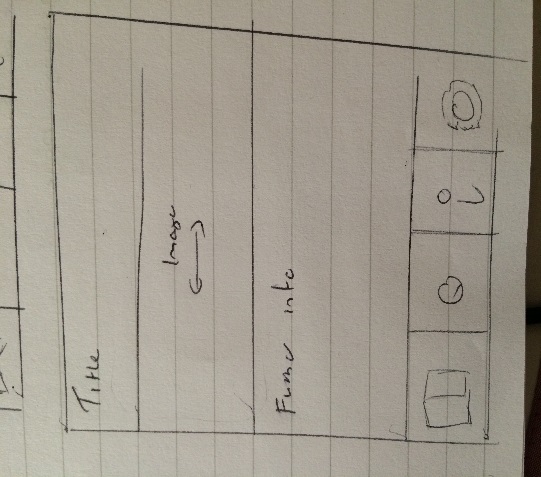
Glossary Page Sketch

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.



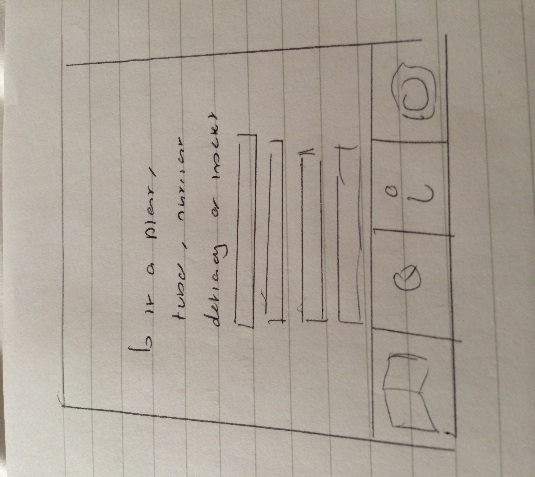
Further Information Page Sketch

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



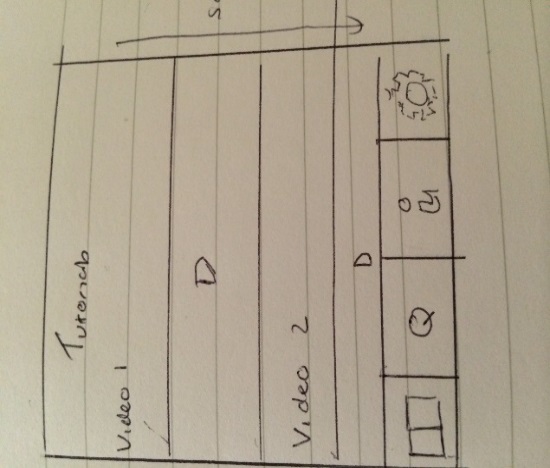
Expert System Page Sketch

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.



**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 plan for the daily meetings were 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 10 minute SCRUM where each member would say what they did the day before, what they were going to do and any obstacles. 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 2 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 finds 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, SUS usability form and the basic task sheet.

**Implementation**

As stated earlier the teams 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 sympton 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

At the start of sprint 1 implementation was quite slow as it took the team a while to pick developing on Android and working with java again. Over the course of the sprint the team achieved both the requirements set with the client and made a start on the expert system (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 there was a sketch of a flickable 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 style the viewpager in the style we wanted. Then when the image is from the slideshow 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 change size based on the image’s size. In the end we solved the issue by using some simple mathematics. Developing a custom list view also was 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 image, name and an arrow on each 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 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 in the database the images names were stored so we could access them from the folder for the relevant piece of data. 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 that were over 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.

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 compression needs to be saved as H.264 baseline. 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 a SQLite database to store the majority of information for the application. A 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 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 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 a 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 res folder and then have code to copy the file in the res 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 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 readability in 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 nice 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 on 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 with 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 and we will achieve for next sprint. At the meeting we recognised she struggled clicking the close button on fullsize images which needs 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. We 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 it who doesn’t own an Android phone and could only test the pinch zoom on an Android device hence only being able to test when around other team members with Android devices. We 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.

Expert system

Syncing

Challenges and successes

Unit tests and refactoring

Meeting and retrospectives. Burndown charts ?

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, who is doing them and how long it would take to get done. 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.

**Evaluation**

* **Client evaluation**
* **Group testing**
* **User evaluation**
* **Evaluative study**
* **Changes from sketches**
* **Rapid prototypes**

**Final product**

* **Final product**
* **Description**
* **Look**

**Appraisal**

* **Project management challenges**
* **Client expectations**
* **Successes**
* **Issues**
* **Technique issues**
* **Lessons learnt**
* **How to improve in the future**
* **Personal development training**
* **Team experience + technological improvement**
* **James Hutton Institute experience**

**Summary & Conclusion**

**Future**

**References**

**Appendix**

* **Trello link**
* **Backlog**
* **Risk assessment**
* **Sketches – Database + design**
* **Requirement sketches**
* **Minutes**
* **Emails and meetings with client**
* **Pictures from James Hutton Institute**
* **Picture from personal development**
* **Log book**
* **User guide - Video**

**Binded + on disk**

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 [↑](#footnote-ref-1)