WikiWalks – Elaboration Phase Status Assessment

# Assessment Against Objectives

## Has Production Level Support for the CCRD Been Achieved?

Yes, we have achieved this objective. This is demonstrated in a short video walkthrough which may be accessed from here: <https://drive.google.com/open?id=14lCHANSxl-6uBpEjniDzBpxdxFcWXtgk>

During the inception phase, we identified the ability to record and retrieve paths from a user’s area as the critical core use case. This is because:

* the purpose of the project is to allow users to find and walk paths in their area and
* in order to find paths, paths must be submitted by other users

We identified a four-tier architectural style as a feasible approach to addressing the requirements of the projects as outlined in the updated and continuing architectural notebook, which may be accessed here: <https://github.com/GoJoeyGo/WikiWalks/blob/master/LCAM_Deliverables/Revised%20Architecture.docx>

The main architectural elements which are demonstrated by the executable architecture are:

* a database to store information related to paths
* a server to handle submission to and returning of data from the database
* an Android app to pull from and submit path information to the serve

Those aspects of the architecture not addressed include:

* extra functionality for the Android app, as it currently only supports paths and not extra data like reviews, images, points of interest, and group walks

Correct support for the CCRD use case by the executable architecture was achieved as demonstrated and documented in the following user acceptance tests.

* Recording paths
* Upload paths with title
* Upload paths without title
* Resuming recordings
* Displaying map
* Displaying map (no connection)
* Uploading path (no connection)
* Generate creation time
* Create path (no location services)
* Resume recording far away

Actual test results can be accessed from here: <https://github.com/GoJoeyGo/WikiWalks/blob/master/LCAM_Deliverables/User%20Acceptance%20Testing.docx>

## Have Critical and Significant Project Risks Been Addressed?

The following list identifies the most critical and significant product, technical and project management risks to the project. Mitigation strategies identified and applied and the current status of the risk are also listed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Priority | Description | Mitigation Strategy | Current Status |
| Individual task in the project takes longer than expected to complete. | 1 | Parts of the project are not completed when they are expected to be. | Regular discussions between team members so that if a member is having difficulty with an assigned task, the group is aware of the problem before it is able to cause further delays in the development of the project. A strike against the team member will be enforced if the task was not completed due to poor planning and lack of communication with the group from the member. | Open – until the project has been completed, this will always be a risk. This problem has come up a few times when it came to documentation for the inception phase, and for a number of programming tasks including allowing the server to accept reviews and pictures, and for submitting paths to the server from the app. |
| Application front-end is unable to be implemented on older Android versions. | 2 | WikiWalks application does not work on Android devices running older versions. | Regular testing on devices running old versions, avoiding unnecessary new API features. | Closed – the basics of the app have been completed utilising only APIs and UI features that are compatible back to Android Jelly Bean. Future development will only require using these same features. |
| Google Maps API is too expensive to use for the WikiWalks project. | 3 | Google Maps API is too expensive to use for the WikiWalks project. | Code review is done before the testing stage is started to ensure that the app accesses Google Maps as little as possible while retaining functionality. | Closed – all functionality requiring use of Google Maps has been implemented and does not require spending money on the API. |
| Loss of data on the WikiWalks server. | 4 | Unable to access data stored on the WikiWalks server, making the app not work at all. | Regularly checking the status of the VM, and ensuring it has all the resources it needs to run effectively. | Open – this is an outside risk that can never be solved. |
| WikiWalks server goes down. | 5 | Data stored on the WikiWalks server is lost or corrupted, making the server inaccessible and effectively useless. | Regular backups of the database. | Open – this is an outside risk that can never be solved. |
| A member of the WikiWalks development team leaves the project. | 6 | A member of the WikiWalks development team is no longer able to work and contribute to the project. | Team members should try to remain in good spirits and redistribute tasks previously assigned to the leaving member. | Open – although extremely unlikely, there is never a guarantee that a circumstance will come up where another member leaves. |
| Team is affected by COVID-19 or other illness. | 7 | Members may be unable to work on and complete tasks on time. | Team members should take precaution to avoid illness, inform the team ASAP if possible, and try to work on what they can. | Open – although the pandemic has subsided for the most part in Australia, a second wave or even just another illness still has a decent chance of lowering productivity for the project. |
| WikiWalks will be unable to complete the project initiation stage. | 8 | WikiWalks will be unable to complete the project initiation or inception stages. | Regular meetings and re-assignment of uncompleted tasks was used to ensure this was not a problem. | Closed – WikiWalks is now well past the initiation stage. |

## Have the Vision, Requirements, or Architecture Changed?

During the elaboration phase, our understanding of the projects aims evolved as follows:

### Vision

* Clarified information saved as it was not specific before – now lists reviews, statistics, and points of interest
* Changed iteration lengths to two weeks instead of one or two as they are only two weeks
* Added points of interest into the needs and features as they are an important data category
* Clarified use of Nginx with Ubuntu, as a generic OS with Apache was listed before, and Apache ended up being more difficult to set up
* Removed specific roles for back/front-end as this didn't end up being the case and all team members are working on all parts

### Requirements (Functional)

Requirements have not been changed, however the document now includes a domain model updated to be more accurate to the class and function names that have been implemented.

### Requirements (Non-Functional)

Also unchanged as the requirements overall haven’t changed.

### Architecture

* Removed reference to waterfall methodology as that’s not what’s being used
* Changed phone in assumptions to device, as it will also work on tablets
* Added adding paths to the database to significant requirements as that is a base feature
* Removed setting point of interest as destination as this is not feasible to complete to a satisfactory standard in the time we have
* Changed MySQL to SQLite as we’re using SQLite for its simplicity
* Changed “navigate to” feature to be specific recorded routes rather than generated routes based on child paths as the latter is not feasible to complete in time
* Changed references to storing “PathMaps” to just storing paths, as PathMap is an internal class in the app, not something stored in the DB
* Changed front- and back-end definitions as they were inaccurate given the context
* Changed key labelling
* Reworded a number of sections for clarity

## Have the Project Plan or Master Test Plan Changed?

During the elaboration phase, our understanding of the best way to implement the project evolved as follows:

### Project Plan

* Added that Joey is team leader
* Changed roles to experience as specific roles for specific parts of the project (e.g. team member designation to just focus on the server or just focus on the app) have not been used
* Removed Isaac as he left the team
* Changed iteration lengths to two weeks instead of one or two as they are only two weeks
* Added phase status assessments to the iteration table
* Added walking UI to the iteration table
* Changed server dependencies to be more specific

### Master Test Plan

* Changed testing schedule for adding and getting paths to final iteration of LCAM
* Added additional path data tests to plan

# Deliverables

## Revised Project Vision

The project vision has been revised to be clearer and fit the project better, as outlined in section [1.3.1](#_Vision).

There were no issues encountered producing this document.

## Revised Requirement Model

The requirement model has been unchanged as the initial document still fits the requirements of the updated vision and plan.

There were no issues encountered producing this document.

## Final Architecture

The project architecture has been revised to be clearer and fit the project better, as outlined in section [1.3.4](#_Architecture).

There were no issues encountered producing this document.

## Revised Master Test Plan

The master test plan has been slightly changed to make testing more comprehensive, as outlined in section [1.4.2](#_Master_Test_Plan).

There were no issues encountered producing this document.

## Executable Architecture

A server for hosting the database and an Android app that covers the critical core use case has been developed. You currently can send paths to and retrieve paths from the server, displaying them on a map and giving you the option to walk these paths or record a new one.

The only issue this section had was falling behind a little in iteration 3, however this was resolved by the end of iteration 4 and everything was caught up.

## Evidence of Testing

### User Acceptance Tests

User acceptance testing has been completed, as mentioned in [1.1](#_Has_Production_Level), and all but 1 test passed.

There were no issues producing these tests.

### Unit and Integration Tests

Unit testing has been completed, with functions to run call individual functions within the server with test data for both submitting and retrieving paths. Additionally, unit testing has been done to ensure paths are created properly within the app.

Integration testing has been completed, with functions being developed to automatically pull data from the server and check that everything is within range, and to generate, submit, edit, and then finally delete a new path.

There were no issues producing these tests.

## Revised Project Plan

The project plan has been revised to be clearer and fit the project better, as outlined in section [1.4.1](#_Project_Plan).

There were no issues encountered producing this document.

# General Issues

There were no general issues during this phase. The only issue encountered was falling behind a bit in iteration 3, which was mentioned in section [2.5](#_Executable_Architecture).

# Summary – Overall Project Progress

The project has satisfactorily completed all it needs to for the elaboration phase, having fully developed the server and developing the parts of the app required to take care of the critical core use case. All risks left open in the project are risks outside of our control, and we have mitigation and contingency strategies should one of them become a real problem. There were no major changes to the project scope or project plan, most revision was simply clarifying vague statements, changing wording, or adding more detail that was decided on beforehand but just never added. There are no ongoing issues as of the conclusion of this phase.