# Construction Phase Status Assessment

## 1. Assessment against Objectives of the Construction Phase

### 1.1 Does our tool cover all use cases?

We understand the main functional requirements of the project are as described in the requirements model, currently our tool covers all requirements.

The non-functional requirements are currently met and the tool meets usability, security and performance metrics.

This is demonstrated through the UAT testing as well as the unit testing.

### 1.2 Have we performed Alpha UAT and does functionality meet expectations?

Alpha UAT has been performed and links can be provided, these tests show that the functionality requested can be provided by the tool

### 1.3 have integration and unit tests been completed

Integration and unit testing has been completed as per our master test plan.

All testing is completed with code coverage of 73.9% and All successful tests.

## 2. Deliverables

### 2.1. Beta Ready Implementation of the final application

Our application is in production ready for consumption for beta UAT. <https://ms.havealookdemo.com.au>

Our source code can be found here: <https://github.com/MatthewJuliusScott/ResourceManager>

We have a demonstration video of our product in production here: <https://drive.google.com/file/d/1CVOqkvsqI4LusZD8pi3S5F_Pt13Fn7ne/view?usp=sharing>

### 2.2 Test model

We have provided our master test plan and have performed all current testing as required.

This includes unit tests, integration tests and UAT tests. Please see below link for test results:

<https://github.com/MatthewJuliusScott/ResourceManager/tree/master/Documentation/Test%20Results>

### 2.3 User Manual

The user manual can be found at the link provided and contains instructions on how to perform each function provided by the tool.

## <https://github.com/MatthewJuliusScott/ResourceManager/blob/master/Documentation/RM_UserGuide.docx>

## 3. General Issues

### 3.1 Communication within team suffered through the construction phase

The team had difficulties during the construction phase when tasks were completed or when tasks were available for other members to work on. This led to a heavy work load for the following iteration, and team members being disciplined. This issue has now been resolved and the team is communicating more frequently.

## 4. Risks

### 4.1 Poor Use of Version Control

This risk can cause project delays if the team does not appropriately use version control, causing the project to fail.

The mitigation strategy is to use a common version control system with available help online and from more experience team members helping via Teamviewer when issues arise.

This risk is resolved.

### 4.2 Team has Inadequate Skillset

This risk can cause project delays in the form of team members only learning requisite technologies when they need them which can make developing the application take longer than required.

The mitigation strategy for this is to use skills taught during the CSU Computer Science Bachelors and to occasionally have skill quizzes to ensure the teams skills are up to date.

This risk is still open but being managed.

### 4.3 Team Members Leave the Project

If any members of the team leave the project, due to the small project team, some use cases will have to be removed.

The mitigation strategy is to ensure that team members are all well communicated to and for the team to be open with each other to manage expectations.

This risk is still open and being managed through ensuring the team discusses issues when the occur.

### 4.4 Inadequate Security

If the program is not secure enough to store data, then the users will not use it.

We will ensure the security of the data is appropriate to the requirements.

This risk has been mitigated through HTTPS and implementing a standardized and well tested security implementation. We are using Spring Security, so each endpoint URL is secured by an authenticated session, and each service method is also secured by authenticated credentials.

User passwords are encrypted using BCrypt, and we only store the Hash and Salt of each password in the database so even a data breach does not reveal our users’ passwords.

### 4.5 Incapable of Providing Needed Functionality

The risk of not meeting the client’s needs can lead to refusal of product purchase thus a waste of time and money on part of the developers, alongside a potential loss of reputation.

The mitigation strategy is to establish the project requirements and selecting technologies that can deliver the services required by the client.

This risk is being mitigated by iteratively performing UAT with the client.

### 4.6 Team Unable to Identify or Procure Technologies

If team members are unable to determine the technologies that should be used, the project will cease to proceed past the planning phase.

The mitigation strategy is to investigate early into the technologies to be used.

This risk is resolved.

### 4.7 Team continues to communicate poorly.

During the Elaboration phase the team had communication issues which are now resolved.

If the team continues this trend the project will likely fail.

This risk is being managed by ensuring an iteration manager is responsible to ensure delivery of all artefacts. Our team charter dictates a three-strike termination policy, and any serious breaches in communication warrant a strike to dissuade not communicate major issues with the team.

## 5 Summary – Overall Project Progress

We have achieved construction phase deliverables; the application has been alpha tested and is ready and available with documentation for Beta testing.

There are no ongoing issues; however we have started using GitHub issues to track project defects. This allows us to track when a bug occurs, how to reproduce it, and what the current status of it is.

<https://github.com/MatthewJuliusScott/ResourceManager/issues>