# Title

# Abstract

# Table of contents

# Introduction

## Problem statement

The projects finished capability as described by the client is an application accessible on PED’s (personal electronic device) designed for co-ordinating/collaboration of teams, tasks and projects that is approved for security classification of up to official. Currently teams use outlook calendars, excel spreadsheets vera teams or a “defence diary” on the DRN (Defence restricted network) to organise their projects, the issue with the current methods are information is split between many different applications or by individuals which can cause team managers to lose track of ongoing work due to difficulty finding where information is stored. Documenting task completion can be difficult for personnel as DRN access is not always easy on a job and their competed task are document long after it has been done (end of the day for example). This can cause needless delays or even outright forgetting to document task completion. With the current system used for organising projects there are many points where confusion can be caused, and information can go missing or undocumented.

## Client Needs

The intent behind this project seems to be to create one easy to access application for both team managers and there personal to track day to day projects in real time as well as standardising a communication method for projects while keeping a detailed record of both communication and task progress (likely to aid with accountability and mistake finding). This will likely be most accessible as a downloadable application on an individual’s phone or app.

The main problem with current options in the market seem to be the security risk associated with using applications where direct control of information, servers and source code are not controlled by defence Australia. These risks include uncertainty of encryption levels of information being send over networks and server hosting location (application database access not controlled by defence). These seem to be the main reasons why a dedicated app made and controlled by defence is needed. Even through not directly stated being able to time maintenance and server down time to prevent needless confusion for personal is also a reason on why defence would prefer to use their own personal servers. Also, from my own personal experience with using the DRN (current solution used DRN access) I have found that having to access program through the DRN entail going through a virtual machine which causes all application usage to come with lag or a delay causing user experience to be slow when performing any actions. To have a highly responsive product for end users the application will need to be separate from the DRN to prevent the issues associated with virtual machines.

## High level objectives

The project can be defined by 2 high level objectives the product or application must be designed to meet. These being official+ level encryption and application capability.

* official+ level security (HL-1)

The application will need to be clear by defence to have access to information at official at minimum or higher securities levels. This will mean data send over any network will need to be encrypted, users will need to be authenticated and data stored will also need to be encrypted. Any other possible security vulnerability will also need to be addressed, eliminated, or minimised to a reasonable degree. These will include but not limited to cross website scripting, sql injection, PED’s loss (device password storage and loss of access), key loggers, remote access to device and shoulder surfing.

* Application Capability (HL-2)

The application must be capable of organising, sorting, and tracking on going work projects from wherever a user is.

## Low Levels objectives

The high-level objectives can be converted to the following low-level objectives.

* RSA encryption (HL-1)

RSA encryption is an asymmetric encryption method approved for information with the classification TOP SECRET. The RSA will need a 3072-bit key to be approved for this level of classification. RSA encryption stated will need to be used to exchange AES keys. (Australian Signals Directorate, 2021)

* AES encryption (HL-1)

AES will be used to send information across a network at minimum AES-256 will need to be used for information of up to TOP SECRET (Australian Signals Directorate, 2021). AES will be used for general information exchange as RSA will likely be far too slow to both encrypt and decrypt data causing end users to have an unresponsive user experience (Priyadarshini Patil, 2015).

* Password hashing (HL-1)

To prevent storing passwords in plain text in the database hashing will need to be used. SHA-2-384 will be used and is clear of TOP SECRET classified information. (Australian Signals Directorate, 2021)

* Ability to create team/project and create/track, update tasks (HL-2)

The application will need to capable of assisting team leaders and personal in managing tasks. Therefore, the application will need features and functionality to create team/project and create/track, update tasks.

* Ability to communicate to team or individuals (HL-2)

Project management no matter the job will require a way for individual assigned to teams to communicate in a quick and easy manner. This will entail the ability to send messages to groups and individuals in teams as well as push notifications to alert users of messages.

* Easy to understand UI design (HL-2)

Projects can become quickly overwhelming with the sheer number of tasks required to be done. The user interface for clients will need to be easy to navigate to important information as well as ways to view only important information to prevent screen cluttering as most users will be using the application on small screens.

## Project Constraints

The constraints on this project’s solution revolve around security.

The most impactful security issues are with iOS kill codes installed on all apple devices using iOS 9 onwards. Mobile device management (MDM) administrator can remotely enable lost mode and from there remotely lock and wipe devices. (2021 Apple Inc, 2021) It should be noted that this can be done without any agreement with user if desired from apple (from a technical perspective). This is an issue as if apple desired, they could wipe all apple devices used by Australian defence personnel. As the app aims to be used to coordinate defence personnel having a 3rd party with the power to disrupt communication would be a major security concern. To avoid this the app will not be designed for apple devices forcing users to have PED’s with operating systems that do not have iOS installed.

There are many security issues with using webs-apps that include but are not limited to cross website scripting, sql injection and key loggers for example. Due to the limited time and developers on the project it will not be feasible to develop an application that is able to handle all these security issues. To complete this project in the timeframe a mobile app will be developed as they are fully executable programs with far less security issues to handle which will therefore save time.

# Literature Review

The Australian Signals Directorate (ASD) has approved the cryptographic algorithms of RSA with 3072 bit key, AES-256 and SHA-2-384 for information classified up to TOP SECRET. (Australian Signals Directorate, 2021). Due to defences high security standards these algorithms will need to be implemented to allow for secure communication between user apps and defence servers in this project.

RSA is a widely used encryption and decryption algorithm but also an asymmetric algorithm meaning it has different keys for encryption and decryption allowing for safe key exchanges between parties. (A. A. Hasib and A. A. M. M. Haque, 2008)

An example of using RSA to secure connections for Secure e-learning web-based application has been implemented and shown capability to prevent data theft, data modification, data fabrication of an unauthorized user and prevents files from being readable both in storage and transmission through the encryption process. The RSA algorithm also had the benefit of authentication data to a specific user (Baihaqi, 2017). This example shows that the RSA algorithm is an effective algorithm to establish secure connection between parties and is an acceptable method to establish security keys and encrypt data over a network.

As stated, the RSA is an effective algorithm however there are disadvantages to using it. These being the time it takes to encrypt/decrypt data as well as the avalanche effect. The avalanche effect is simply how much the data will change once encrypted due to a small change in the original text. Out of the most used encryption algorithms (DES, 3DES, AES, Blowfish) RSA performed the worst having the least amount of change. Note that AES performed the best (Priyadarshini Patil, 2015). The RSA algorithm also is easily the worst algorithm to choose from when comparing encryption and decryption times as it will grow with the size of the data (over 2 seconds for 3MB of data) where other algorithms will not grow due to these circumstances such as AES for example (Priyadarshini Patil, 2015). This shows that using AES algorithm will allow for a far more responsive app as well as shown far greater changes from the original plaintext when encrypting data. For the project using AES will be a better choice over RSA however AES is a symmetrical algorithm creating a problem with secret key exchange between communicating parties.

To solve the issue with key exchange an implementation of using a combination of RSA and AES was used to secure electronic health record application. This method used the RSA algorithm to send the AES secret key over a network encrypted. This allowed a secure connection between party to be form using the AES algorithm without risk of the disclosure of the secret key (Wardhani, 2016). A similar method will be used in this project to create a connection between a server and client apps.

RSA and AES will be used to establish an encrypted communication between the server and user apps however authenticating users will be done using knowledge possessed by the user in the form of passwords. To prevent users’ passwords or other identifying knowledge being possessed by the server to be disclosed a hashing algorithm will needed to be used on user’s data. Approved hashing algorithms include SHA-2 with 384-to-512-bit outputs may be used for information with the classification of TOP SECRET (Australian Signals Directorate, 2021).

SHA-2 can protect user’s data as it transforms an input message into a 256 bits message. This transformation is one way, and the original message cannot be recreated from the resulting transformation (R. V. Mankar, 2013). The SHA-2 has shown a high level for randomness in tests being able to completely remove the original input and compress it down to the specified bits (Z. Al-Odat, 2019). These tests have shown that the SHA-2 hashing algorithm is a highly effective algorithm with no noticeable issues and is therefore the algorithm that will be used to secure passwords in the sever database for the project.

In the java standard libraries or API provided by Oracle for the Java platform includes built-in many of the most used cryptographic algorithms, including the RSA and AES encryption algorithm as well as the SHA message digest algorithm and key agreement algorithms (2021 Oracle, 2021). These libraries will allow the security required by defence to be implemented using the programming language java therefore java will be used to develop the project.

# Methodology

## Overarching approach

## Data flow description

## Software tools used

# Results and Discussion

# Conclusion

# Timeline

## Login functionality

The first thing after the programs were communicating fully encrypted was the ability to authenticate what the sever was communicating with, which was done in the form of a login page. To do this a rudimentary database was created to store username and passwords and sever connection to said database was tested as well as the ability to query the database (how to get and change information on the database from the sever program). A login page for the user app was also created to allow users to send inputted information over to the sever then check it against the record in its database and send a response back to the user app. After this login functionality was implemented. A visual of the login page can be seen in figure 1 below.

## User registration

A page to register new users to the database was created, this worked the same way as the login page except it will add information to the database instead of authenticating a user. If/when this app is implemented the server should probability have some way of authenticating new users such as checking pmkeys or needing a defence email to add a new user to simply prevent non-defence personnel from even accessing the network. Currently program just excepts all new entries if username is not already in database. A visual of the registration page can be seen in figure 2 below.

## SHA encryption

Storing passwords and other identifying information in plain text on severs would allow anyone with database access to looked at user’s information, to prevent this a one-way encryption is used on this sort of information. The encryption algorithm of SHA with 384-bit output with salting (a key is used during encryption) was used as recommended by defence for information up to top secret.

## Home UI skeleton

Once a user has been authenticated by the sever the user app will enter its main navigation area. It was decided to use a bottom navigation bar with four possible displays being a projects page, create project page, messages, and diary tabs to navigate to the future planed feature of the app. At this point each of the displays where empty/had placeholding images.

## Database design

The database in its current form would not be capable of storing the necessary information needed to represent a user’s work-project. At this point creating some functionality to end users was being planned out and the database needed to go through a redesign to represent the information needed for a user’s work project.

## Create project page.

Once the database was redesigned a way to allow users to store information was needed. Under the create new project tab from the bottom navigation option a create new projects page was created seen in figure 3. This page functioned like the registration page as it would send information to the sever to be stored.

## Project Selection page.

The project selection page is the tab of the bottom navigation options that a user will be sent to after login. This page displays each project the user is connected to (as indicated by the server) and can be scrolled through. The plan is to make each of these projects a button which will send the user to a projects individual information page however this is not yet implemented.

## Project Visualisation

Below in figure 1 the image on the right side shows the projects tab which displays in a scrollable list all the projects the currently log in user is associated with and the image on the right shows the page a user is redirected to when one of the projects are selected. This area that users are redirected to is made up of 4 tabs used to interacted with the selected project. The tabs being home, overview, add user and add task. Note that the home page hasn’t been implemented yet. This tab is intended to be used to display project updates and team leader messages to the team.

The first tab that was implemented was the add users tab. This tab can be seen in figure 2 below. This tab will display all user that are not currently associated with the project and when a user is selected that user will be given access to the project. There is also a search tab at the top of this page which can be used to filter out users. After a recent meeting with the client this tab will need to be updated to only be accessible to team leaders and users accounts with increase access.

The next tab that was implemented was the add task tab which can be seen below in figure 3 the left image shows the top of the page, the middle the bottom and the right shows how dates are selected. This tab is used to create a task for the project and upload the information to the server.

The next tab that was implemented was the project overview tab as seen below in figure 4. This tab is used to display task information of the selected project to users in the form of a Gantt chart. The page is split with tasks listed vertically on the left (can be scrolled through) and right of that a table that can be scrolled horizontal where each column will represent a time. At the top of the page there is a button that acts as a title and a way to change the columns between weeks and days. The columns represent either a day or a week the first being the project creation date or earliest task start date in project. The current day/week can be found by finding the columns that have a purple boarder around them (note that there are some missing graphics as seen in figure 4 as orange is yet to have a boarder around it and there isn’t a red block with a boarder either). It is planned that the top number column will be able to be click on and the date it represents will be displayed. Tasks are ordered from top to bottom first their status as tasks that are in progress will be first in the list, then followed by completed then cancelled. After task status tasks are ordered by their planned finished date. The colours that fill the chart are blue for in progress, orange for task running overtime, green for complete and red for cancelled.

Below in figure 5 is an example of the task info page which is accessible by clicking on the tasks in the left most column in figure 4. This page is used to display task specific information and the ability to update task information. Currently only the ability to update the status is available however it is planned to create a task specific log and the ability to change the assigned user to the project.

# References

# Appendices

Source code

Creation scripts