Joshua Monaghan

www.joshmonaghan.com

Software Engineering

PWA Scope

Table of Contents

[**Client and Problem:** 4](#_Toc194674750)

[**Social, Legal and Ethical Implications:** 4](#_Toc194674751)

[Social Implications 4](#_Toc194674752)

[Customer Privacy 4](#_Toc194674753)

[Trust and Transparency 4](#_Toc194674754)

[Accessibility 5](#_Toc194674755)

[Legal Implications 5](#_Toc194674756)

[Data Protection Laws 5](#_Toc194674757)

[Customer Consent & Data Usage 5](#_Toc194674758)

[Intellectual Property Concerns 5](#_Toc194674759)

[Ethical Implications 6](#_Toc194674760)

[Data Retention and Deletion 6](#_Toc194674761)

[Marketing Practices 6](#_Toc194674762)

[Privacy Invasion 6](#_Toc194674763)

[**Why is the project a PWA:** 7](#_Toc194674764)

[Benefits 7](#_Toc194674765)

[Cross-Platform Compatibility 7](#_Toc194674766)

[No App Store Dependency 7](#_Toc194674767)

[Automatic Updates 7](#_Toc194674768)

[Security 7](#_Toc194674769)

[Disadvantages 8](#_Toc194674770)

[Limited Device Features 8](#_Toc194674771)

[iOS Limitations 8](#_Toc194674772)

[Performance Constraints 8](#_Toc194674773)

[Limited Multi-Tasking 8](#_Toc194674774)

[**Project Requirements:** 9](#_Toc194674775)

[Client Interactions 9](#_Toc194674776)

[First Interaction with Client 9](#_Toc194674777)

[First Requirements Interaction 10](#_Toc194674778)

[Second Requirements Interaction 11](#_Toc194674779)

[Signed Requirements 11](#_Toc194674780)

[Application Requirements 12](#_Toc194674781)

[Measurable Requirements 14](#_Toc194674782)

[**Security:** 14](#_Toc194674783)

[Common Web Vulnerabilities 14](#_Toc194674784)

[Security Principles 15](#_Toc194674785)

[PWA Security Implemented 16](#_Toc194674786)

[**Storyboards:** 17](#_Toc194674787)

[Client Interaction 17](#_Toc194674788)

[Initial Interaction 17](#_Toc194674789)

[Client Feedback 17](#_Toc194674790)

[Signup 18](#_Toc194674791)

[Login 19](#_Toc194674792)

[Create Firm 20](#_Toc194674793)

[Dashboard 21](#_Toc194674794)

[Company View 22](#_Toc194674795)

[Add Company 23](#_Toc194674796)

[Leads 23](#_Toc194674797)

[Add Lead 24](#_Toc194674798)

[Quotes 24](#_Toc194674799)

[Add Quote 25](#_Toc194674800)

[Invoices 25](#_Toc194674801)

[Add Invoice 26](#_Toc194674802)

[Contacts 26](#_Toc194674803)

[Add Contact 27](#_Toc194674804)

[Reports 28](#_Toc194674805)

[Settings 29](#_Toc194674806)

[Page Links 0](#_Toc194674807)

[**Data Dictionaries:** 0](#_Toc194674808)

[SQL Tables 0](#_Toc194674809)

[Users 0](#_Toc194674810)

[Firm 1](#_Toc194674811)

[Company 1](#_Toc194674812)

[Quote 2](#_Toc194674813)

[Invoice 3](#_Toc194674814)

[Lead 3](#_Toc194674815)

[Contact 4](#_Toc194674816)

[Python Data 5](#_Toc194674817)

[Users 5](#_Toc194674818)

[Firm 6](#_Toc194674819)

[Company 6](#_Toc194674820)

[Quote 7](#_Toc194674821)

[Invoice 7](#_Toc194674822)

[Lead 8](#_Toc194674823)

[Contact 9](#_Toc194674824)

[**Gantt Chart** 10](#_Toc194674825)

[**Assess feasibility** 10](#_Toc194674826)

[Projects Feasibility 10](#_Toc194674827)

[Challenges with Development 10](#_Toc194674828)

[**Research and Planning:** 11](#_Toc194674829)

[Web Frameworks 11](#_Toc194674830)

[Django 11](#_Toc194674831)

[ASP.Net Core 11](#_Toc194674832)

[React 12](#_Toc194674833)

[Chosen Framework 12](#_Toc194674834)

[Tools 13](#_Toc194674835)

[Programming Languages 13](#_Toc194674836)

[Tools 13](#_Toc194674837)

[Development Approach 14](#_Toc194674838)

[**Logbook** 15](#_Toc194674839)

# **Client and Problem:**

**Client:** Trevor Monaghan, middle aged man who has expertise’s in designing and building Progressive Web App and accounting. He understands what a PWA is and what is needed to design and build one which will make the requirements straightforward and detailed.

He is the owner of **Climax Valuations and Forensics**, a business located in Belmont NSW, specialising in business valuation and forensic accounting reports, predominantly for court matters such as divorce, shareholder disputes and economic damages. He has no accessibility requirements, but they will be included because the project is SaaS allowing multiple users to use it.

Trevor is seeking a PWA solution to manage all his client’s details, files, valuation reports and communication. His current Client Relationship Manager (CRM) is increasing their pricing which is why he needs a free solution to ensure his overheads are kept to a minimum. This will help him have easier access to his client's needs and details while also making him more productive and efficient.

# **Social, Legal and Ethical Implications:**

## Social Implications

### Customer Privacy

Customer privacy is an essential requirement for all projects, ensuring that all user’s information and data is secure and only accessible to authorised individuals. Australia’s Privacy Act 1988 defines how user’s information is handled and stored, ensuring that users can feel safe and secure on the web. In the CRM project, customer privacy is the biggest concern because of sensitive data for both the user and company.

### Trust and Transparency

Trust and transparency are very important in web development, being transparent about how and why their data is stored and handles allows a trusted connection between the user and service. Trust can be gained by being upfront about the secure data practices and having consent forms so the user can decide whether they want to agree. Having users trust the CRM application allows the application to include customer feedback and support.

### Accessibility

Accessibility in web development is a crucial aspect for user with disabilities. A PWA should meet the accessibility standards such as the WCAG, this ensures that every user can navigate and use the application without any issues. This includes:

* Easy navigation
* Layout is accessible for screen readers
* Indexing for keyboard only users
* Colour contrasts for visual impaired users
* Font sizes are readable
* Alternative text for images

Ensuring accessibility in a CRM allows a wider range of users, allowing businesses to not have to worry about accessibility for disability employees.

## Legal Implications

### Data Protection Laws

Data Protection laws like Privacy Act 1988 or APPs are extremely important for both businesses and users. The data protection laws state that all sensitive information or data should be stored securely and have security protocols in place to protect it. This is both important for legal reasons and trust with users. Ensuring that passwords and sensitive data are encrypted and limiting the data a user can see both revolve around the privacy laws. Data should only be stored if strictly necessary and users should have the right to remove/delete their own data. In the CRM application, only limited data about the user and company are stored to ensure security. Fines and reputation damage can occur if these laws aren’t met.

### Customer Consent & Data Usage

Customer consent for data usage allows for both ethical and legal advantages. Customers should understand what data is being collected, why it’s being collected and how it will be used. This will create customer trust and transparency ensuring that they consent to all data being collected and stored. The CRM will allow users to consent to the data being collected and explain why and how it’s being handled/stored.

### Intellectual Property Concerns

Intellectual Property or IP concerns around the ownership of user’s data and the CRM software must be clearly defined and accessible for the users. Being transparent about who owns the CRM and if third parties are involved ensure that it’s communicated to the users. Concerns over who owns the customers data is important to communicate to the user, in this PWA the user owns their own data. This ensures that their sensitive data is secured with them and can’t be sold off to other businesses or companies.

## Ethical Implications

### Data Retention and Deletion

Data Retention is an important aspect of projects that store sensitive information, it allows for data deletion after a certain period to enforce security and up-to-date data. An example of this would be if a user has been offline for over 5 years, there account could be deleted to ensure that their data is no long a security threat to them. The PWA application won’t delete data for the reason of company quotes and invoices may need to be traced back. Users should also have the right to delete their own data, in the CRM application the users have full control on deleting their own data.

### Marketing Practices

Marketing practices are typically aimed at users and advertising based off their information. Systems typically collect data like preferences, customer interactions and behaviours, this allows them to market the most appropriate advertising to the user. In the CRM application, there will be no marketing to users in the project. This will boost trust with the user and ensure that no legal issues with marketing will be addressed. Marketing can harm user’s relationships and gives a bad reputation on the business so removing marketing from the project will solve these problems

### Privacy Invasion

Privacy invasion in the context of a CRM occurs commonly because they are required to store sensitive information about businesses, clients and users. Excessive data collection plays a role in privacy invasion, if data is not needed then storing it is only a security risk. Sharing data with third parties invades both Legal and Privacy issues which can cause reputations damage and fines towards the company. To avoid privacy invasion the CRM will be transparent about all data being stored and handled while allowing the users to consent to this before any sensitive information is stored.

# **Why is the project a PWA:**

A PWA (Progressive Web App) is an efficient and easy using way of sourcing software on the internet. The following reasons why I’ve selected a PWA as the most suitable choice for the project:

|  |  |
| --- | --- |
| Benefits | Disadvantages |
| Cross-Platform Compatibility | Limited Device Features |
| No App Store Dependency | iOS Limitations |
| Automatic Updates | Performance Constraints |
| Security | Limited Multi-Tasking |

## Benefits

### Cross-Platform Compatibility

PWAs are web-based applications that combine websites and apps together on the internet, because they are on the web this allows anyone to access the application given that they have a web browser. This enables faster development and removes the need for multiple different versions of the application to be made to suit the devices requirements. This is crucial for the project; it removes the need for multiple versions of the application to be made which saves time.

### No App Store Dependency

Since PWAs are on the web this removes the need for the application to be downloaded and installed on the device. This streamlines user to access the application allowing for quick access given the user has an internet connection. In relation to the CRM, this allows users quick and easy access to the project without the need for downloads or updates.

### Automatic Updates

Unlike software installed on a device, PWAs can push updates to the application without the user even realising, allowing bug fixes or new features to be added to the PWA without user’s realising. This accelerates the user’s productivity and accessibility to the application while ensuring new features are automatically updated when a user revisits the site. This will help users have access to new features in the CRM without needing any downloads.

### Security

PWAs use security by design to ensure that user’s information is secure for unauthorised personals. Since the application is on the web, HTTPS encrypts all data that’s received and sent ensure man-in-the-middle attacks are avoided. Everything in the database is also stored on a secure server, not on the user’s device, which increases the security for all users. The PWA will be constructed with all common web security protocols, ensuring the project is secure for the users.

## Disadvantages

### Limited Device Features

PWAs have limitations when it comes to a full range of devices features. They are defined to what the browser allows, these results in limited features like Bluetooth and Advanced camera functionality which can’t be accessed. PWAs cannot access the devices system-level features unlike an application installed on a device, this prevents background processing for client side, advanced notification systems or managing files on the device. The CRM will not need any of these advanced device related features allowing for the PWA to be the appropriate choice for the application.

### iOS Limitations

iOS has more PWA restrictions then most other devices, limiting the features that are usable unlike applications installed on a device. Some PWA features that iOS limit are push notifications, home screen shortcuts and icon flexibility. Performance is also a critical aspect with iOS PWAs, unlike applications installed on the device a PWA will suffer slower performance. This could result in less responsive or smooth experiences for users. Hardware features like background location tracking and Bluetooth aren’t available using a PWA because it runs in a web browser. With all these features limiting PWAs it should not prevent any of the CRM’s features and capabilities.

### Performance Constraints

Performance constraints is a huge disadvantage compared to applications installed on the device. PWAs will typically run slower then native applications because of load times and the browsers limited resources. Applications that need to be fast and efficient are not commonly PWAs and are native apps. These performance constraints can be visible on lower specification devices and can make a user’s experience slow and laggy. In context of the CRM, the application is primarily targeted at computer users and the application is not resource heavy.

### Limited Multi-Tasking

Limited multi-tasking is a considerable downside to PWAs. Native applications on devices have access to run processes in the background and perform multiple tasks at once, while PWAs have access limited access to this if a user was to switch off the web page it may cause the application to stop background processing to save resources. This makes it difficult to perform tasks like location tracking or webpages that requires constant updating to render. PWAs have limited capacity to interact with other applications running on the device, making it difficult for multi-app communications. The CRM will not need any multi-tasking features to be build making this disadvantage redundant.

# **Project Requirements:**

The project is a CRM (Customer Relationship Manager) which allows users to track their quotes, leads, invoices and customer contact information. Trevor, who is the client, currently uses a CRM but needs another one because of the following reasons:

**Pricing**Trevor uses an application called Xero; with this they use to provide a free CRM that he has been using for years. Currently Xero have changed that and are charging a fee to use that feature which is why he needs this project.

**Features**Xero’s CRM doesn’t have all the features that would boost workflow, it does provide basic features but there’s no reporting. The PWA which I’m developing will have multiple features and will include reporting to boost production and experience.

## Client Interactions

### First Interaction with Client

The initial interaction with the client was on the 15/03, the phone call talked about expectations and requirements the project would achieve. After the brief call, I understood what the project would look like and what it should accomplish. A follow up email from the client should be sent in the upcoming days with a draft requirement attached.

### First Requirements Interaction

A screenshot of a computer

AI-generated content may be incorrect.

On the 18/03, the first requirements were received from the client with brief expectations and optional requirements that could be added to the project. After analysing the requirements requested, most requirements could be met in the timeframe assigned but some optional requirements won’t be feasible, like:

* **Stripe Integration**
* **Xero API**

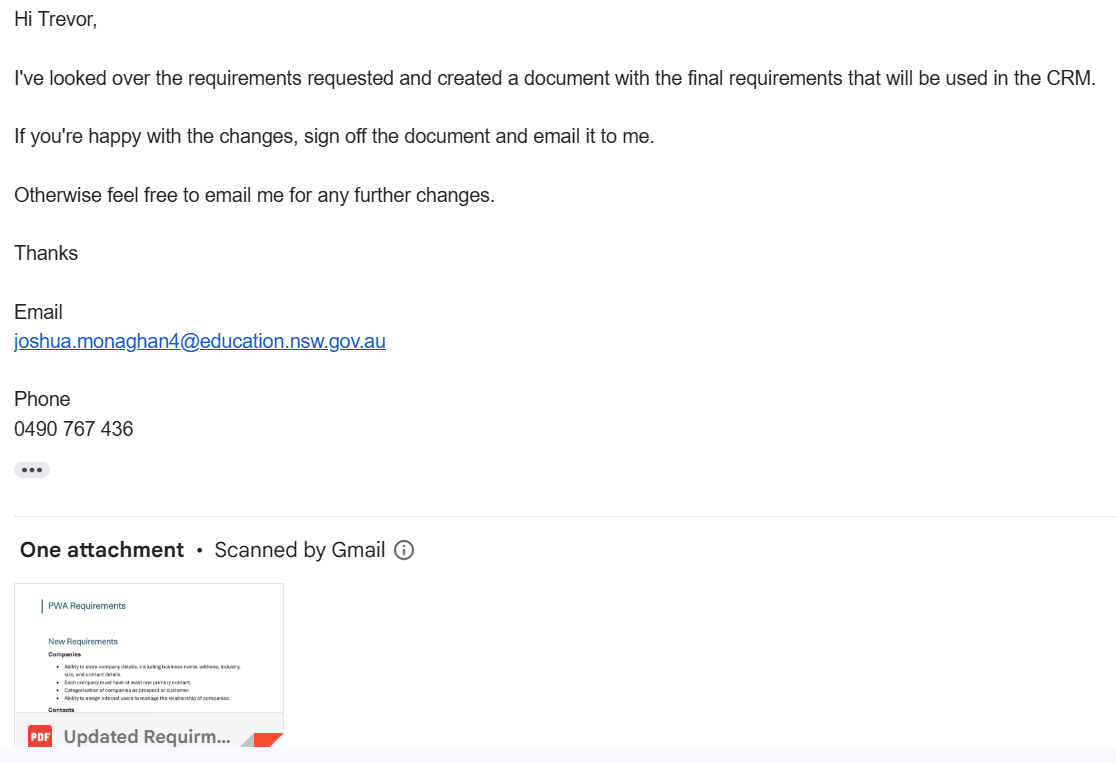
Every other requirement should be met and a follow up email revolving around the updated requirements will be sent in the upcoming days.

[*Initial Requirements for Software by Climax Valuations and Forensics Pty Ltd*](file:///C:\Users\joshm\Documents\SoftwareEngineering\Y12\Assessments\Major-Project\SoftwareEngineering-Major-Project\PWA%20Scope%20and%20Requirements\Initial%20Request%20for%20Software%20by%20Climax%20Valuations%20and%20Forensics%20Pty%20Ltd.pdf)

### Second Requirements Interaction

After reviewing the requirements requested, I will be using all requirements except: Stripe Integration and Xero API. The new requirements were sent through email to the client, this will allow him to review my requirements and give feedback.

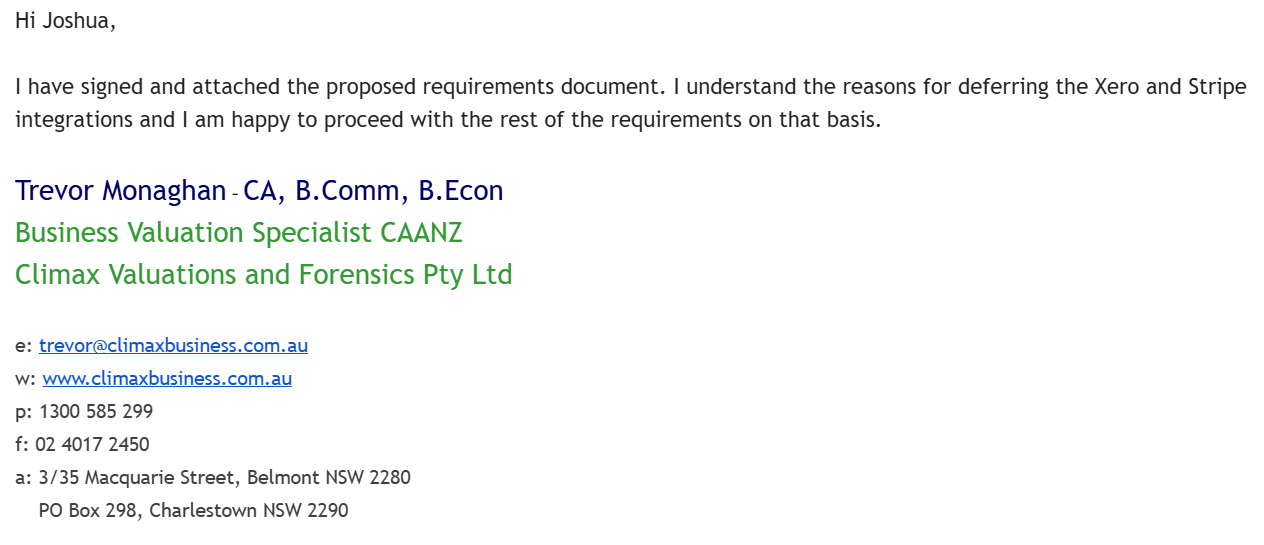
[Updated Requirments.pdf](Updated%20Requirments.pdf)



### Signed Requirements

The client responded with an email with the signed requirements attached. They had no feedback and are happy for me to proceed with the projects with these requirements.

[Requirements - Signed.pdf](Requriements%20-%20Signed.pdf)



## Application Requirements

The application has specific requirements that need to be met for the project to be considered ready for production. Here are the following requirements that need to be met:

**Companies**

* Ability to store company details, including business name, address, industry, size, and contact details.
* Each company must have at least one primary contact.
* Categorisation of companies as prospect or customer.
* Ability to assign internal users to manage the relationship of companies.

**Contacts**

* Ability to store individual contact details, including name, job title, email, phone number, and associated companies.
* A contact can be associated with one or more companies or exist independently without a link to any companies.
* Ability to mark a contact as a primary contact for a specific company.

**Leads**

* Leads are created when there may be a current or future opportunity to provide a quote to a company.
* Each lead must be associated with a company and may also be linked to a specific contact.
* Status tracking for leads (e.g., In Progress, Won, Lost).
* Ability to assign internal users to manage leads.

**Quotes**

* Quotes are generated for leads when a pricing estimate is provided.
* Standard fields for quotes, including quote number, date, associated company, contact, and lead etc.
* Ability to add line items to quotes, including description, quantity, price per unit, and tax code. Software needs to calculate and display the total and any tax.
* Ability to amend and resend quotes.
* Status tracking for quotes (e.g., Draft, Sent, Accepted, Rejected).

**Invoices**

* An invoice is created when a quote is won
* The invoice will initially contain the same line items as the corresponding quote but must be editable so that it can be different to the quote.
* Standard fields for invoices, including invoice number, issue date, due date, and payment terms.
* Ability to add and edit line items like quotes.
* Status tracking for invoices (e.g., Draft, Sent, Partly Paid, Fully Paid).

**Payments**

* Ability to record one or more payments against an invoice.
* Calculation of outstanding balance on each invoice.
* Automatically change the status to paid when payments are equal to the invoice total.

**Reporting**

* Users must be able to generate reports for Companies, Contacts, Leads, Quotes, and Invoices.
* Reports should be filterable by date, status, company, or other relevant fields.
* Ability to export reports to formats such as CSV or PDF.

**User Management & Security**

* Each business subscriber will have its own internal users.
* Role-based access control to restrict functionalities based on user roles.
* 2-factor authentication for all users.
* Secure login system with password recovery options.

**Additional Considerations**

* The CRM will be web-based and must support access from different devices.
* The user interface should be intuitive and easy to navigate.
* The system should be scalable to handle multiple businesses efficiently if we decide to make it a SaaS product.
* Data security and encryption should be implemented for sensitive information.

**Boundaries**

This project is to be built over a 10-week span, so time management is crucial so not all requirements asked by the client can be met. Boundaries are implemented to stop scope creep and to ensure projects stay on task with the assigned timeframe. Nothing outside of the scope will be integrated unless required for the project. No client suggestions will be implemented in the development phase unless strictly needed to stop scope creep.

## Measurable Requirements

The PWA will have measurable requirements to ensure that the application can be tested, ensuring that requirements are met. The following requirements will be necessary for the PWA to be considered finished:

* Pages must not take longer than 3 seconds on an average home Wi-Fi speed
* All pages have links to continuously navigate around the application
* Users cannot view unauthorised data
* All important functions should have tests written to ensure that there functioning properly before production
* Reports must not take longer than 5 seconds to generate on an average home Wi-Fi speed.

# **Security:**

## Common Web Vulnerabilities

PWAs are a hybrid between web and native applications, because of this they have many security vulnerabilities. The following are common web vulnerabilities:

**Cross-Site Scripting (XSS)**Cross-Site-Scripting is a web attack that injects malicious scripts into a web page to try and gain access to unauthorised data. These scripts may try to steal cookies, session tokens, or perform actions on behalf of other users. To prevent this attack your PWA must validate and sanitise every input to ensure no harmful characters or scripts are embedded.

**Cross-Site Request Forgery (CSRF)**Cross-Site request forgery is a vulnerability that uses an authenticated user to perform unintended actions without their consent. This can happen when the website does not verify whether a request was made by a user. Preventing this attack is as simple as using a CSRF token, a unique random token for each user’s session which is required for all form submissions.

**SQL Injection (SQLi)**SQL injection is a dangerous vulnerability, allowing users to query the database from forms on the website. This vulnerability occurs when a developer constructs SQL queries with raw user inputs. Preventing this vulnerability is simple and only requires the developer to validate and sanitise all inputs from forms to reassure that there is no raw SQL in the user’s inputs.

**Insecure Authentication**Insecure authentication happens when an application fails to verify the identity of a user, this allows unauthorised access to sensitive information. Common reasons this occurs would be weak or flawed implementation of authentication, leading to attackers to exploit the weaknesses to gain access to data. Implementing secure multi-factor authentication and using a hashing algorithm to encrypt information can prevent this vulnerability.

**Man-in-the-Middle (MITM)**Man-in-the-Middle attacks occur when an attacker intercepts data that’s communicating across the web. The attacker can listen on that channel without either parties knowing, this leads to sensitive information being leaked or modified messages. Prevention is to use HTTPS protocol which encrypts data while traveling, ensuring that Multi Factor Authentication is implemented ensure that if data is leaked the attacker can’t log in.

## Security Principles

PWAs should be developed with security by design, this meaning security is a constant requirement for all tasks. Developers use a list of established security principles, this helps guide the design, implementation and reduces vulnerabilities. The following are well known security principles:

**Principle of Least Privilege**Principle of least privilege is a principle that reduces all privileges to the very least the role needs. This minimises the potential attacks because of the minimum privileges every role has. An example of this would be a user of the PWA, they don’t need admin privileges, so it won’t be included in their privileges.

**Defence in Depth**Defence in depth is a principle that breaks the project up into multiple layers with different security. If one layer was to fail, then all other layers would still be able to prevent the attack. Commonly the layers are separated into an interface layer, logic layer and a database layer.

**Principle of Least Exposure**Principle of least exposure focuses on exposing minimum sensitive information to the user, this also means restricting which users can see what data. Data should be limited to what is needed on the screen, for example if a user needed to see their username, they shouldn’t also see their email and password.

**Secure by Design**Secure by design is a technique used where security is a constant consideration throughout the entire project. This helps reinforce security by not having to add it later in the project and instead start implementing from the beginning. An example would be using hashed passwords from the beginning or applying CRSF tokens to prevent CSRF.

**Fail Securely**Fail securely is a protocol that is to design a system that will fail securely, this means if something was to fail the system should not expose any sensitive data or allow any unauthorised access to users and attackers. An example of this is if a web application crashes, it should not display the detailed error message, revealing the internal structure of your application.

## PWA Security Implemented

The PWA will be developed and implemented to prevent common web vulnerabilities while using many security principles to prevent any attacks. The following security features will be added to the PWA project:

**HTTPS Protocol**The PWA will be served through over HTTPS to ensure all communication is encrypted and secure from MITM attacks. This will prevent sensitive data from the server or client being intercepted and stolen/manipulated.

**Authentication and Authorisation**Strong authentication and authorisation systems will be implemented to prevent sensitive data from being stolen. This is a big concern considering a CRM stores companies, clients and user’s sensitive data, so ensuring that proper authentication and authorisation is used is important.

**Data Encryption**Data encryption will be heavily used to prevent sensitive data from being leaked or stolen. Considering the CRM is storing very sensitive data, many encrypting techniques will be used to secure the data as best as possible.

**Session Management and Timeout**Session Management and timeout will be implemented to ensure that only authorised users have access to sensitive information. The timeout will prevent people from accessing sensitive information through your computer while the session management will use cookies to ensure no CRSF attacks occur.

**Cross-Site Request Forgery (CSRF) Tokens**Cross-Site request forgery tokens will be implemented to stop request attackers from forging request on behalf of users to manipulate or steal data. Using a CSRF token will ensure that only the user has access to requests.

**Security Testing**Security testing will be conducted throughout the development phase to ensure that all parts of the PWA is tested and secure. Regularly testing common web vulnerabilities like SQL injection and XSS will ensure that the PWA is secure.

**Logging and Monitoring**Logging and monitoring of all actions will be conducted to detect any suspicious request or activity. This could include log in attempt fails or brute force attempts which will be accessible in real-time.

# **Storyboards:**

## Client Interaction

### Initial Interaction

An email was sent to the client with a draft storyboard attached. If there was any feedback on the storyboards, an additional email regarding what changes should be made would occur.

A screenshot of a computer

AI-generated content may be incorrect.

### Client Feedback

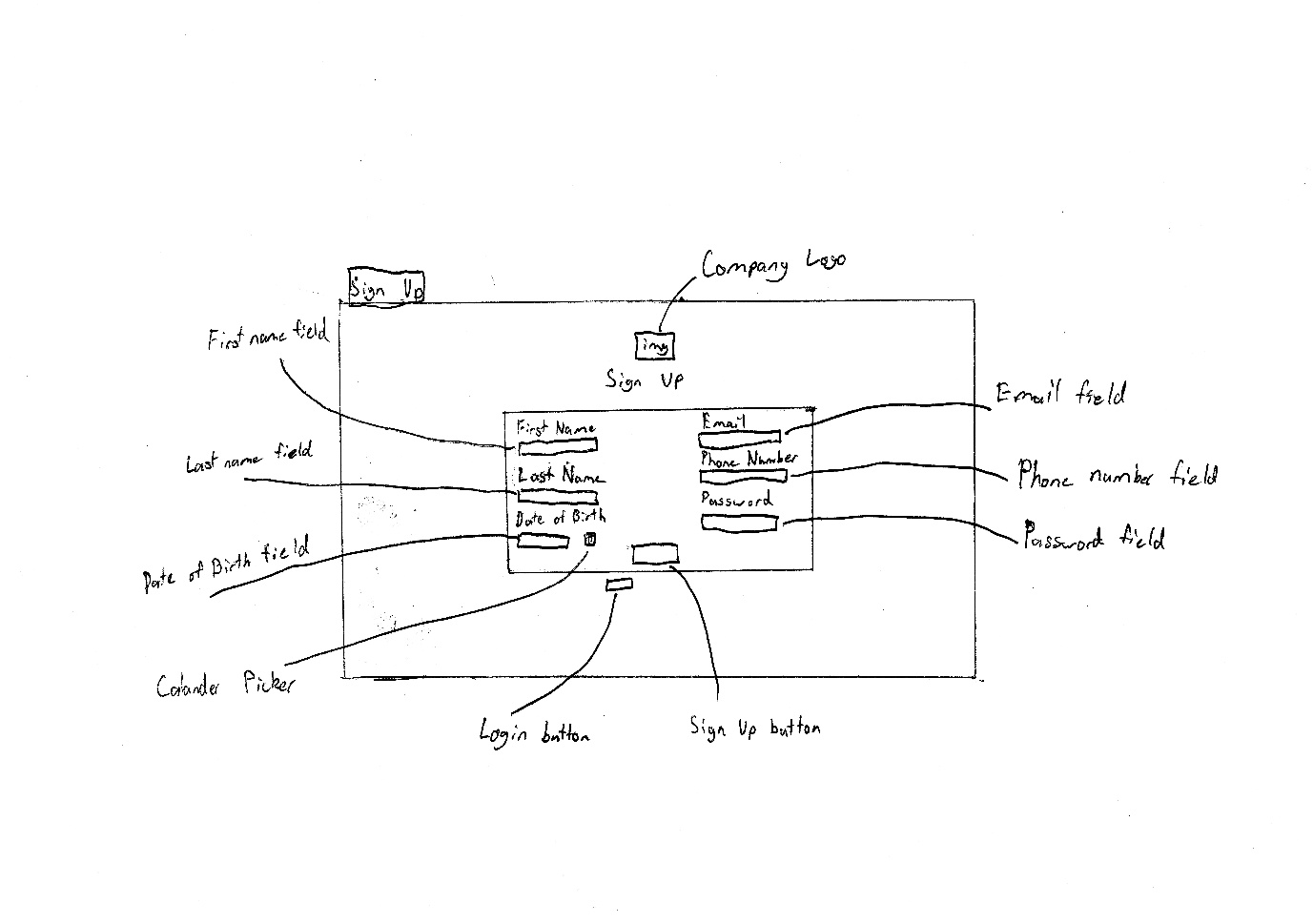
The client returned an email regarding feedback about the storyboards. The feedback was to add an address field for companies and add filters to the leads, quotes and invoice views. The client signed and approved the storyboard in advance because of the minor changes to be made.

A close-up of a computer screen

AI-generated content may be incorrect.

[Storyboard - signed.pdf](Storyboard%20-%20signed.pdf)

## Signup



## Login

A diagram of a electrical system

AI-generated content may be incorrect.

## Create Firm

A diagram of a computer program

AI-generated content may be incorrect.

## Dashboard

A diagram of a computer system

AI-generated content may be incorrect.

## Company View

A diagram of a computer system

AI-generated content may be incorrect.

## Add Company

A diagram of a computer system

AI-generated content may be incorrect.

## Leads

A close-up of a computer screen

AI-generated content may be incorrect.

## Add Lead

A diagram of a computer system

AI-generated content may be incorrect.

## Quotes

A close-up of a computer screen

AI-generated content may be incorrect.

## Add Quote

A diagram of a computer program

AI-generated content may be incorrect.

## Invoices

A diagram of a computer

AI-generated content may be incorrect.

## Add Invoice

A diagram of a computer program

AI-generated content may be incorrect.

## Contacts

A diagram of a computer

AI-generated content may be incorrect.

## Add Contact

A diagram of a computer system

AI-generated content may be incorrect.

## Reports

A diagram of electrical wiring

AI-generated content may be incorrect.

## Settings

A diagram of a web page

AI-generated content may be incorrect.

## Page Links

A diagram of a computer network

AI-generated content may be incorrect.A diagram of a electrical system

AI-generated content may be incorrect.

A close-up of a computer screen

AI-generated content may be incorrect.A diagram of a computer

AI-generated content may be incorrect.A diagram of a computer

AI-generated content may be incorrect.A diagram of electrical wiring

AI-generated content may be incorrect.

Only go to page if no firm is created

A diagram of a computer

AI-generated content may be incorrect.A diagram of a computer system

AI-generated content may be incorrect.A diagram of a computer program

AI-generated content may be incorrect.

A diagram of a web page

AI-generated content may be incorrect.A diagram of a computer system

AI-generated content may be incorrect.A diagram of a computer system

AI-generated content may be incorrect.A diagram of a computer program

AI-generated content may be incorrect.A diagram of a computer program

AI-generated content may be incorrect.A diagram of a computer system

AI-generated content may be incorrect.A diagram of a website

AI-generated content may be incorrect.

# **Data Dictionaries:**

## SQL Tables

### Users

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Variable | Data Type | Format for Display | Size in Bytes | Description | Example | Validation |
| UserID | Char (32) | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for every user. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| FName | Varchar (50) | Plain Text | Up to 50 Bytes | Stores the first name of the user. | Josh | Validates that the user’s input is only letters and is the correct length. |
| LName | Varchar (50) | Plain Text | Up to 50 Bytes | Stores the last name of the user. | Monaghan | Validates that the user’s input is only letters and is the correct length. |
| DOB | Date | DD MM YYYY | 3 Bytes | Stores the user’s date of birth. | 11-11-2008 | Validates that the year is appropriate, and the date and month are valid. |
| Email | Varchar (80) | All lowercase | Up to 80 Bytes | Stores the users email address. | joshmono2008@outlook.com | Validates the email address is valid. |
| PhoneNum | Varchar (15) | +CC-XXX-XXX-XXXX | Up to 15 Bytes | Stores the users Phone Number. | +61 490 767 436 | Validates that the phone number has a valid country code and local number. |
| Password | Varchar (250) | Alphanumeric with special characters allowed | Up to 250 Bytes | Stores a encrypted version of the users password. | $argon2i$v=19$m=16,t=2,p=1$WXZwbGpjUEdLUEZBeFlEUw$U/E9CHQXKaTReVVeZRvrmQ | Validates that the encrypted password is valid. |
| SecretKey | Varchar (16) | Alphanumeric and all upper case. | 16 Bytes | Stores a secret key used for Two Factor authentication. | QAAL5GI2H3XIRFHB | Validates that the secrecy key is 16 characters long and is valid. |
| FirmID | Char (32) | Alphanumeric and all lower case. | 32 Bytes | Many to many link to the users firms. | a81bc81bdead4e5dabff90865d1e13b1 | Validates that the link to the firms exists. |
| Admin | Bit | Yes/No | 1 Byte | Determines if the user is an Admin | False | Validate that the user is allowed to be an admin. |

### Firm

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Variable | Data Type | Format for Display | Size in Bytes | Description | Example | Validation |
| FirmID | Char (32) | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for every firm. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| Name | Varchar (50) | Plain Text | Up to 50 Bytes | Stores the firm’s name. | Apple | Validates that the user’s input is only letters and is the correct length. |
| Email | Varchar (80) | All lowercase | Up to 80 Bytes | Stores the firms email address. | business@business.com | Validates the email address is valid. |
| PhoneNum | Varchar (15) | +CC-XXX-XXX-XXXX | Up to 15 Bytes | Stores the firms Phone Number. | +61 490 767 436 | Validates that the phone number has a valid country code and local number. |

### Company

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Variable | Data Type | Format for Display | Size in Bytes | Description | Example | Validation |
| CompanyID | Char (32) | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for every user. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| Name | Varchar (50) | Plain Text | Up to 50 Bytes | Stores the firm’s name. | Apple | Validates that the user’s input is only letters and is the correct length. |
| Contact | Char (32) | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for every Contact. This field is a many to many so a company can have many contacts. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| Address | Varchar (100) | Plain Text | 100 Bytes | Stores the address of company | 1 Wood St Newcastle 2290 | Ensure the address is valid and below 100 bytes. |
| FirmID | Char (32) | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key linking to the lead. Optional Field. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |

### Quote

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Variable | Data Type | Format for Display | Size in Bytes | Description | Example | Validation |
| QuoteID | Char (32) | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for every quote. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| Pricing | Money | Currency | 8 Bytes | Stores multiple prices for the quote. | $12500 | Validates that the input is a valid currency. |
| Contact | Char (32) | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for every Contact. This field is a many to many so a quote can have many contacts. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| Notes | Varchar (300) | Plain Text | 300 Bytes | Store’s notes about the quote | Ring company. Email Joe. | Validates that the text is a valid format. |
| Status | ENUM | Text | 1 Byte | Stores multiple different enumerable for status. | “Pending Payment” | Ensures the Enum chosen is valid. |
| Item | Char (32) | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for multiple items related to the quote. One to Many. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| LeadID | Char (32) | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key linking to the lead. Optional Field. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |

### Invoice

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Variable | Data Type | Format for Display | Size in Bytes | Description | Example | Validation |
| InvoiceID | Char (32) | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for every invoice. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| Pricing | Money | Currency | 8 Bytes | Stores multiple prices for the invoice. | $12500 | Validates that the input is a valid currency. |
| Contact | Char (32) | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for every contact. This field is a many to many so a invoice can have many contacts. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| Notes | Varchar (300) | Plain Text | 300 Bytes | Store’s notes about the invoice. | Ring company. Email Joe. | Validates that the text is a valid format. |
| Item | Char (32) | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for multiple items related to the invoice. One to Many. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| Status | ENUM | Text | 1 Byte | Stores multiple different enumerable for status. | “Pending Payment” | Ensures the Enum chosen is valid. |
| QuoteID | Char (32) | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key linking to the quote. Optional Field. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |

### Lead

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Variable | Data Type | Format for Display | Size in Bytes | Description | Example | Validation |
| LeadID | Char (32) | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for every lead. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| Pricing | Money | Currency | 8 Bytes | Stores multiple prices for the lead. | $12500 | Validates that the input is a valid currency. |
| Contact | Char (32) | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for every contact. This field is a many to many so a lead can have many contacts. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| Status | ENUM | Text | 1 Byte | Stores multiple different enumerable for status. | “Pending Payment” | Ensures the Enum chosen is valid. |
| Notes | Varchar (300) | Plain Text | 300 Bytes | Store’s notes about the lead. | Ring company. Email Joe. | Validates that the text is a valid format. |

### Contact

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Variable | Data Type | Format for Display | Size in Bytes | Description | Example | Validation |
| ContactID | Char (32) | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for every contact. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| FName | Varchar (50) | Plain Text | Up to 50 Bytes | Stores the first name of the user. | Josh | Validates that the user’s input is only letters and is the correct length. |
| LName | Varchar (50) | Plain Text | Up to 50 Bytes | Stores the last name of the user. | Monaghan | Validates that the user’s input is only letters and is the correct length. |
| Email | Varchar (80) | All lowercase | Up to 80 Bytes | Stores the firms email address. | business@business.com | Validates the email address is valid. |
| PhoneNum | Varchar (15) | +CC-XXX-XXX-XXXX | Up to 15 Bytes | Stores the contact’s Phone Number. | +61 490 767 436 | Validates that the phone number has a valid country code and local number. |

## Python Data

### Users

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Variable | Data Type | Format for Display | Size in Bytes | Description | Example | Validation |
| User | Record |  | ~529 Bytes | A record that’s linked to the SQL database tables. |  | Ensures all attributes are valid. |
| User.UserID | UUID | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for every user. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| User.FName | String | Plain Text | Up to 50 Bytes | String of the users first name. | Josh | Validates that the user’s input is only letters and is the correct length. |
| User.LName | String | Plain Text | Up to 50 Bytes | String of last name of the user. | Monaghan | Validates that the user’s input is only letters and is the correct length. |
| User.DOB | Date | DD MM YYYY | 3 Bytes | Stores the user’s date of birth. | 11-11-2008 | Validates that the year is appropriate, and the date and month are valid. |
| User.Email | String | All lowercase | Up to 80 Bytes | String of the users email address. | joshmono2008@outlook.com | Validates the email address is valid. |
| User.PhoneNum | String | +CC-XXX-XXX-XXXX | Up to 15 Bytes | String of the users Phone Number. | +61 490 767 436 | Validates that the phone number has a valid country code and local number. |
| User.Password | String | Alphanumeric with special characters allowed | Up to 250 Bytes | String of encrypted version of the user’s password. | $argon2i$v=19$m=16,t=2,p=1$WXZwbGpjUEdLUEZBeFlEUw$U/E9CHQXKaTReVVeZRvrmQ | Validates that the encrypted password is valid. |
| User.SecretKey | String | Alphanumeric and all upper case. | 16 Bytes | String of secret key used for Two Factor authentication. | QAAL5GI2H3XIRFHB | Validates that the secrecy key is 16 characters long and is valid. |
| User.Firm | Record |  | ~177 Bytes | Record of the firm that links to user. |  | Validates that the firm exists. |
| User.Admin | Bool | True/False | 1 Byte | Determines if the user is an Admin | False | Validate that the user is allowed to be an admin. |

### Firm

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Variable | Data Type | Format for Display | Size in Bytes | Description | Example | Validation |
| Firm | Record |  | ~177 Bytes | A record that’s linked to the SQL database tables. |  | Ensures all attributes are valid. |
| Firm.FirmID | UUID | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for every firm. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| Firm.Name | String | Plain Text | Up to 50 Bytes | Stores the firm’s name. | Apple | Validates that the user’s input is only letters and is the correct length. |
| Firm.Email | String | All lowercase | Up to 80 Bytes | Stores the firms email address. | business@business.com | Validates the email address is valid. |
| Firm.PhoneNum | String | +CC-XXX-XXX-XXXX | Up to 15 Bytes | Stores the firms Phone Number. | +61 490 767 436 | Validates that the phone number has a valid country code and local number. |

### Company

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Variable | Data Type | Format for Display | Size in Bytes | Description | Example | Validation |
| Company | Record |  | ~214 Bytes | A record that’s linked to the SQL database tables. |  | Ensures all attributes are valid. |
| Company.CompanyID | UUID | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for every user. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| Company.Name | String | Plain Text | Up to 50 Bytes | Stores the firm’s name. | Apple | Validates that the user’s input is only letters and is the correct length. |
| Company.Contacts | List |  | ~227 Bytes | A list of records of the contacts that links to company. | [contact1, contact2] | Validates that the contacts exist. |
| Company.Address | String | Plain Text | 100 Bytes | Stores the address of company | 1 Wood St Newcastle 2290 | Ensure the address is valid and below 100 bytes. |
| Company.Firm | Record |  | ~177 Bytes | Record of the firm that links to company. |  | Validates that the firm exists. |

### Quote

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Variable | Data Type | Format for Display | Size in Bytes | Description | Example | Validation |
| Quote | Record |  | ~436 Bytes | A record that’s linked to the SQL database tables. |  | Ensures all attributes are valid. |
| Quote.QuoteID | UUID | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for every quote. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| Quote.Pricing | List | Currency | ~8 Bytes | List of multiple prices for the quote. | [12500, 200] | Validates that the input is a valid currency. |
| Quote.Contact | List |  | ~227 Bytes | A list of records of the contacts for the quote. | [contact1, contact2] | Validates that the contacts exist. |
| Quote.Notes | String | Plain Text | 300 Bytes | Store’s notes about the quote | Ring company. Email Joe. | Validates that the text is a valid format. |
| Quote.Item | List | Text | ~32 Bytes | A list of records of item for the quote. | [item1, item2] | Validates that the items exist. |
| Quote.Lead | Record |  | ~372 Bytes | Record of the lead that links to quote. |  | Validates that the lead exists. |

### Invoice

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Variable | Data Type | Format for Display | Size in Bytes | Description | Example | Validation |
| Invoice | Record |  | ~436 Bytes | A record that’s linked to the SQL database tables. |  | Ensures all attributes are valid. |
| Invoice.InvoiceID | UUID | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for every quote. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| Invoice.Pricing | List | Currency | ~8 Bytes | List of multiple prices for the invoice. | [12500, 200] | Validates that the input is a valid currency. |
| Invoice.Contact | List |  | ~227 Bytes | A list of records of the contacts for the invoice. | [contact1, contact2] | Validates that the contacts exist. |
| Invoice.Notes | String | Plain Text | 300 Bytes | Store’s notes about the invoice | Ring company. Email Joe. | Validates that the text is a valid format. |
| Invoice.Item | List | Text | ~32 Bytes | A list of records of item for the invoice. | [item1, item2] | Validates that the items exist. |
| Invoice.Quote | Record |  | ~436 Bytes | Record of the quote that links to invoice. |  | Validates that the quote exists. |

### Lead

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Variable | Data Type | Format for Display | Size in Bytes | Description | Example | Validation |
| Lead | Record |  | ~372 Bytes | A record that’s linked to the SQL database tables. |  | Ensures all attributes are valid |
| Lead.LeadID | UUID | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for every quote. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| Lead.Pricing | List | Currency | ~8 Bytes | List of multiple prices for the lead. | [12500, 200] | Validates that the input is a valid currency. |
| Lead.Contact | List |  | ~227 Bytes | A list of records of the contacts for the lead. | [contact1, contact2] | Validates that the contacts exist. |
| Lead.Notes | String | Plain Text | 300 Bytes | Store’s notes about the lead | Ring company. Email Joe. | Validates that the text is a valid format. |

### Contact

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Variable | Data Type | Format for Display | Size in Bytes | Description | Example | Validation |
| Contact | Record |  | ~227 Bytes | A record that’s linked to the SQL database tables. |  | Ensures all attributes are valid |
| Contact.ContactID | UUID | Alphanumeric and all lower case. | 32 Bytes | A UUID value that is used as a primary key for every contact. | a81bc81bdead4e5dabff90865d1e13b1 | Ensure that the UUID length and characters match the criteria. |
| Contact.FName | String | Plain Text | Up to 50 Bytes | Stores the first name of the user. | Josh | Validates that the user’s input is only letters and is the correct length. |
| Contact.LName | String | Plain Text | Up to 50 Bytes | Stores the last name of the user. | Monaghan | Validates that the user’s input is only letters and is the correct length. |
| Contact.Email | String | All lowercase | Up to 80 Bytes | Stores the firms email address. | business@business.com | Validates the email address is valid. |
| Contact.PhoneNum | String | +CC-XXX-XXX-XXXX | Up to 15 Bytes | Stores the contact’s Phone Number. | +61 490 767 436 | Validates that the phone number has a valid country code and local number. |

# [**Gantt Chart**](GanttChart.pdf)

A diagram with multiple colored lines

AI-generated content may be incorrect.

# **Assess feasibility**

## Projects Feasibility

Given the timeframe for development of the PWA is 10 weeks, all decisions must come down to is it achievable. The current PWA requirements will be achievable based on the Gantt chart created, this ensures that the project will be finished. The requirements were refined to remove unreasonable expectations, this was important to ensure that the user had not only a finished project but a polished and tested PWA. Since I’m a one-person team and all tools are open source or free there is no cost in developing this project. For production, an AWS server will cost money to constantly host while a domain name also needs to be bought. With these in mind a budget of $50 a month ,depending on how many server are running, will be enough to support the fees.

## Challenges with Development

Throughout development there will be many challenges and features that will require researching. Generating reports is something that I’m not familiar with but with the given time frame it should be possible to learn how to and implement it into the PWA. Reporting is an important part of a CRM so removing it from the requirements was not feasible. There should be no other challenges since I’ve completed most of the requirements in several other projects.

# **Research and Planning:**

## Web Frameworks

### Django

#### Overview

Django is a python web development framework used for rapid development. It is built on the model template views (MTV) architecture which allows for ease of use and scalability. Django is an easy framework to learn with class base database models allowing for simple and fast development.

#### Key Features

* Service Worker Integration
* Cache Management
* Security Features

#### Advantages

* Djangos security features are beneficial for handling sensitive data.
* Python and Django’s ease of use allows for fast and efficient development time.
* Django has a small learning curve

#### Disadvantages

* Django lacks the ability to create dynamic UI rending without using front end libraries

### ASP.Net Core

#### Overview

ASP.Net Core is a powerful and fast framework developed in C# allowing for high-performance web applications. This framework can handle both frontend and backend development allowing for easier interaction between the two layers.

#### Key Features

* Service Worker and Manifest Support
* Offline Capabilities
* Blazor (WebAssembly)
* Authentication and Security

#### Advantages

* ASP.Net tooling with Visual Studio support makes it a great choice for enterprise development.
* Blazor allows C# to be written for the front end, allowing full stack development across all layers
* ASP.net can be deployed on Linux, MacOS and Windows

#### Disadvantages

* ASP.Net has a large learning curve for developers who are more familiar with JavaScript based frameworks.
* Blazor cannot create dynamic webpages as well as other frameworks like React.

### React

#### Overview

React is a JavaScript based framework that is primarily used for single-page applications. It is currently maintained by Facebook making it reliable with constant security features and patches. If the developer has a strong JavaScript background the learning curve is very little and easy to pick up.

#### Key Features

* Service Worker Integration
* App Manifest Support
* Offline Support
* Performance Optimisations

#### Advantages

* React is optimised for dynamic interfaces
* It has several third-party libraries that can extend the PWA functionality
* React has a large community with many tutorials.

#### Disadvantages

* React is focused on front end development making the need for a backend such as Node.js to be required for sever side integration.

### Chosen Framework

The PWA being conducted will use Django for the framework for several factors. The ease of development will help reach deadlines for this project, ensuring that every requirement will be met. Django security features are beneficial when dealing with sensitive data which the PWA being a CRM will greatly increase security for sensitive information. The learning curve with Django will greatly increase production allowing for faster and more efficient development and delivery. With my familiarity with Django, it was the most appropriate choice for the framework of this PWA.

## Tools

These tools have been selected to suit the projects short deadlines, performance needs and for familiarity since I’ve constructed PWAs with these tools before.

### Programming Languages

**HTML**HTML is a widely used markup language used to layout web pages. The CRM webpages heavily rely on HTML to be display information and format the pages. I’m very familiar with HTML and can create webpages without any issues, allowing for faster and more efficient development time. Django’s uses HTML for their templating, based on this HTML will be required for this project.

**CSS**  
CSS is used to style HTML webpages, for example changing colour, adding margins, or changing fonts or text size. For the story boards to look accurate to the CRM, CSS is mandatory to ensure accessibility and navigation for users. Django allows simple integration for CSS, making it accessible for development.

**JavaScript**  
JavaScript is used to create dynamic web pages; it allows get and post request without leaving the page. The CRM will need JavaScript to allow dynamic web pages, for example a button could open a form on the same page. Django allows easy integration for JavaScript, making it easy to include in the project.

**Python**  
Python is a high-level programming language which is commonly used widely. I’m confident with programming in Python which will allow for faster development time. Django uses python as the logic layer language, this means python is mandatory for this project.

**SQL**  
SQL is a standard query language used for interacting with database systems. The CRM project needs SQL to store user information and all companies, leads, quotes, invoices, contacts and firms. Django uses SQL to queries the assigned database for the project, allowing for easy development.

### Tools

**GitHub**  
GitHub is a version control management system that allows all history for a repository to be tracked. Features like pull and push request allows for teams of developers to work on the same project without any issues. GitHub will be used to allow development across multiple computers, while tracking all history of pushes.

**Visual Studio Code**  
VSCode is a free open source IDE that helps developers with programming. An IDE has multiple features like debugging, running code, autocomplete code and bug fixing tools. The CRM will be built using VSCode because it’s the IDE I’m most comfortable using,

**PostgreSQL**  
PostgreSQL is a database system that is secure and reliable. It is a common system that is paired with Django projects, because of this it has many guides and tools to help with integration. The CRM’s database will be using PostgreSQL because of security and my experience with it.

**Docker**  
Docker is a software that lets you run applications in a virtual environment. It allows you to create images which will create an image with the specified requirements installed for example, python. This will allow the CRM to be easily added to production servers without the need for installing many requirments.

## Development Approach

The development approach chosen will be Waterfall because of the structed approach allowing for faster development times. This will be crucial for developing and testing the project before the deadline is reached. Since the requirements are clear and detailed, waterfall is an appropriate approach allowing for simple and fast development. Agile and WAgile were also looked at, agile would take too long making the project making it not feasible while WAgile would also do the same because of its agile like development phases. The Gantt chart was created to suit this development approach. The projects complexity has been narrowed down in detail with the requirements and data dictionary, this allows for a top-down approach which will lead to a fast and well laid out PWA.

# **Logbook**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date | Task Achieved | Difficulties and Solutions | Ideas and Thoughts | Reflection | References |
| 15/03 | Identify what my project would be and who my client is. | Trying to find a project that would cover all areas in web development. | Unsure if the project is too large to finish in the time frame provided | I have a client and project which is a good starting point. |  |
| 16/03 | Create GitHub Repository. | None | None | None |  |
| 16/03 | Wrote about the client and the project. | Only problem was I had not received any requirements. | I knew that I needed requirements before I started the other parts. | Good start but need requirements asap. | <https://www.climaxbusiness.com.au/> |
| 18/03 | Received Requirements | None | Now that I had the requirements, I was able to complete the rest of the report. | Need to start the rest of the report now. |  |
| 19/03 | Application Requirements and Boundaries | No difficulties | I understand the boundaries and requirements that the application requires. | Need to start story boards and data dictionary |  |
| 25/03 | Client Interactions | Waiting for client to respond. | I had sent the requirements over, just waiting for client to respond | While I’m waiting, I can start on other parts. |  |
| 26/03 | Start on Data Dictionary | Had difficulties understanding what the data dictionary needed and what tables I would need. | Creating a picture of what this reports database is going to have to look like. | Good start just needs to finish it off. |  |
| 26/03 | Gantt Chart | Difficulties logging in and using | Could make project an approach agile. | Looks good just need to touch up on a few things. | <https://www.atlassian.com/agile/project-management/gantt-chart> |
| 27/03 | SQL Data Dictionary | Difficulties making the tables match requirements | I have an idea now of what tables ill need and what links to what. | Good table design that will be used in the project. | <https://www.w3schools.com/sql/sql_datatypes.asp> |
| 28/03 | Edit Why the project is a PWA | No difficulties | Edited this section to suit the marking criteria more. | Looks a lot better than the previous version | <https://inoxoft.com/blog/benefits-of-progressive-web-apps-pwa-advantages-and-disadvantages/> |
| 29/03 | Start Story Boards | Struggled with ideas | Made a brief start on the story boards, lots more to do. | Didn’t give me 100% attention so not much was completed. |  |
| 30/03 | More Story Boards | Struggled with screens linking. | Made a lot better of an attempt, not finished yet. | Almost done them all, just need a few more minor screens | <https://www.dreamhost.com/blog/make-your-website-accessible/#:~:text=This%20can%20be%20achieved%20through,menus%20and%20other%20dynamic%20content>. |
| 31/03 | Finish Story Boards | No struggles | Finished off story boards and made diagram of all the page linking | Page linking is messy, but Mr Bartlett said this was the way to do it. |  |
| 01/04 | Client Signed Requirements | No issues | Client signed off the storyboards. | Story boards were detailed with complex layouts, should be fun building it. |  |
| 01/04 | Social, Ethical and Legal Implications | No issues | Lots of words and is detailed. | I made it extra detailed because I lost marks in the last assignment because of this section. | <https://www.koombea.com/blog/legal-requirements-for-app-development/> |
| 01/04 | Benefits and Disadvantages of a PWA | No issues | Not as detailed as I would like but running out of time. | Good effort for the time I spend doing it. | <https://median.co/blog/what-are-pwas-pros-cons-and-considerations> |
| 01/04 | Finish Page links map | No issues | Finished off the page links, the rest of the report is now writing. | Good effort, hard to read without zooming in. |  |
| 01/04 | Common Web Vulnerabilities | Issues finding information | Was a pain to find information on these. | Detailed with lots of vulnerabilities. | <https://www.toptal.com/cybersecurity/10-most-common-web-security-vulnerabilities>  <https://www.cloudflare.com/learning/security/threats/owasp-top-10/> |
| 01/04 | Security Protocols | Issues finding information | Was a pain to find information on these. | Detailed with lots of protocols. | <https://www.catonetworks.com/network-security/network-security-protocols/>  <https://www.6clicks.com/resources/answers/what-are-the-5-basic-security-principles#:~:text=The%20five%20basic%20security%20principles,foundation%20of%20effective%20cybersecurity%20strategies>.  <https://www.cyber.gov.au/resources-business-and-government/essential-cybersecurity/ism/cybersecurity-principles> |
| 01/04 | PWA Security Implemented | No issues | Talked about security would be implemented into the project. | Easy and quick to write. |  |
| 01/04 | Web frameworks | No issues | Talked about 3 frameworks and their strengths and weaknesses | Wasn’t as much detail as I would like but it covers the marking criteria | <https://careerfoundry.com/en/blog/web-development/django-framework-guide/>  <https://medium.com/@darshan.unadkat/pros-and-cons-of-net-core-37ec451edd0>  <https://www.naukri.com/code360/library/advantages-and-disadvantages-of-react-js> |
| 01/04 | Tools used | No issues | Gave me a chance to think about all the tools I would need. | Easy to write |  |
| 01/04 | Development Approach | No issues | Waterfall was the easiest approach to choose and my Gantt chart alright followed that structure. | If I had more time, I would have chosen agile. | <https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model>  <https://www.opentext.com/what-is/agile-development>  <https://www.codestringers.com/insights/wagile-project-management/> |
| 01/04 | Assess feasibility | No issues | Talked about cost and feasibility of the project. | Easy part to write |  |
| 02/04 | Python Data Dictionary | Issues understanding what it wanted | Was by far the weirdest part because I didn’t understand what it was after. | Once I understood it was easy. |  |