Table of Contents

Table of Figures

1. Project Statement

In both the public and private sector, sensitive public information is often freely accessible to members of staff and securely stored. This is a common problem. Users are often careless with passwords sending them across insecure external email systems or casually leaving them as written notes on a colleague's desk or monitor.

Obtaining these login credentials can allow for malicious attackers and unauthorized users to gain access to sensitive personal information. (Li, Wang, & Sun, 2017)

Access control is rarely implemented and there are often no restrictions on access to files and data, with staff members being capable of accessing whatever information on the system regardless of their trust factor.

These are serious security violations and have strong implications, particularly in recent years with the introduction of the General Data Protection Regulation in European Union law which protects the data protection and privacy all individuals within the European Union. All private and public companies dealing with sensitive data must adhere to strict provisions and requirements pertaining to the processing of personal data.

This project aims to resolve the problem of insecure login protocols and lack of access control by providing a multi-factor authentication system that can be used for controlling access further into the system.

2. Research

There are many problems associated with passwords which are discussed below along with a brief introduction to the domain and the evolution of security authentication and Access Control.

2.1 Introduction:

With most organisations and business opting to take their systems online in recent years, cyber-crime is on the rise. Malicious attacks against, government organisations, private companies and individuals are more common than ever with hacks, exploits and data breaches making daily headlines. For cyber criminals seeking an entry point to these organisation's systems, user accounts are often targeted to gain access.

As cybercrime evolves to become even more sophisticated than it already is, legacy systems without solid security will be open to modern threats and attacks. Simple human error, software exploits and much more can be used to execute attacks which can lead to severe financial and reputational loss for organisations. Despite Password and Username combinations being extremely insecure, they remain the most common form of user authentication today and although using a password is better than having no protection at all, they are based upon human intelligence. (Li et al., 2017)

In 2014 security firm 4iQ discovered a collection of databases on the dark web that contained 1.4 billion usernames and passwords in plain text. The databases are a cluster of various leaks found on the dark web. These leaks included plain text credentials from Bitcoin, LinkedIn, NetFlix, Zoosk and video games such as Minecraft and Runescape. A simple search of the password's "admin", "administrator" and "root" returned 226,631 records from administrator accounts. With the most common user passwords being "123456", "qwerty", "password" and "111111". ("Collection of 1.4 Billion Plain-Text Leaked Passwords Found Circulating Online," 2014)

This discovery shows the problem with passwords that users are using and how easy hackers could "crack" or guess login combinations.

With password recycling being another major problem with users, if a hacker was to gain access to an account with login credentials, these can often be used to gain access to multiple other accounts often exposing lucrative and sensitive data. In fact, during an analysis in 2013 of 6 million accounts it was found that 10,000 commonly used passwords could be used to brute force access to 99.8% of those accounts. It was also discovered that 73% of these accounts were sharing their online banking password with one or more non-financial sites, meaning if their passwords were leaked their bank account would be under threat. Without more and more companies deciding to take their businesses online everyday this issue is forever expanding as more user accounts are created. (Lindemann, 2013)

In the 1960s, passwords were for access to time-shared mainframe computers, the main purpose for the introduction of passwords was to stop researchers from using more resources and time on computers than they had been authorized and allocated. The Compatible Time-Sharing system was deployed at MIT in 1961, being one of the first to deploy passwords for resource control. However, with this new mechanism came security issues. The main one being that users could simply guess another user's password allowing them to use resources that had been allocated to the other user. (Bonneau, Herley, van Oorschot, & Stajano, 2015)

Further access control was developed during the 1970s, however this time passwords were adapted not only to protect computational resources but also sensitive data. MULTICS passwords, developed by Roger Needham and Mike Guy at the University of Cambridge, were implemented. These involved storing passwords in hashed form. (Bonneau et al., 2015)

On November 2nd 1988 the Morris worm was launched from the computer systems of the Massachusetts Institute of Technology, being one of the first worms to be distributed via the Internet. The worm exploited vulnerabilities in Unix services. The worm spreads by locating target accounts on the host account and infects them by copying itself to the remote machine if there is not already a copy there. According to its creator Robert Morris, the worm was written as an intellectual exercise to gauge the size of the internet. However, a critical error transformed the worm into a denial of service attack. To compensate for the possibility of administrators not allowing the worm to be copied to the system by running a process that answered “yes” when prompted whether there was a copy of the worm on the system, the worm was directed to copy itself to the system one out of seven times regardless of the systems response. This minor detail proved excessive and the level of replication caused the worm to spread rapidly. Once a machine was infected the uncontrollable replication would cause the machine to allocate processing memory to the worm which resulted in a degrade in machine performance and sometimes an entire crash.

As a result of the worm, administrators adopted to store password hashes in more heavily protected shadow password files. (Jajoo, 2018)

With the boom of the World Wide Web and e-commerce in the 1990s, attempts were made to replace passwords with public-key cryptography via SSL client certificates. This however proved too heavy a burden and the market never developed any further. Instead connections on the Web reply on one-way authenticated SSL for security. Servers are authenticated by a certificate and users prove their identity in different ways. Text-based passwords entered in HTML forms are exchanged for HTTP cookies, for user authentication. This has become the dominant form of user authentication and remains today.

Attempts have been made by multiple businesses to provide different forms of authentication for users known as Multi Factor Authentication. (Bonneau et al., 2015)

Multi Factor Authentication is adding multiple layers of security to an account. Usually users enter their username and password, but rather than gaining instant access, they must provide multiple other pieces of information. This way organisations can ensure that users who are trying to gain access to their accounts, are who they say they are.

These multi factors usually consist of:

- something you know – Password, PIN code, security questions
- something you have – NFC card, smartphone, authentication token
- something you are – Iris scanner, fingerprint scanner

When organisations adopt a multi factor authentication approach for logging into their systems it adds multiple levels of security to their user's accounts. A potential compromise of login credentials or a single factor won't allow hackers to gain access to the account. (Mohammed, Ramkumar, & Rajasekar, 2017)

As organisations decide to further deploy onto computer and internet-based systems more and more information is being exchanged and shared among different departments within but also outside organisations data protection becomes an increasingly difficult problem. Access Control allows an organisation to regulate and protect their systems. It is one of the main technologies for ensuring system security. By restricting permissions for accessing information stored on the system another essential layer of security is added. (Lindemann, 2013)

It is clear today that security requirements have evolved creating the need for an increasingly heterogeneous authentication mechanism. There is no one method or approach that can meet these diverse requirements. However, by providing multiple authentication methods along with access control, the reliance on insecure passwords may be reduced or mitigated.

2.2 Existing Multifactor Authentication & Access Control Products

There are many current existing multifactor authentication solutions which are discussed below, each having advantages and disadvantages concerning them. Reviewing these will give an understanding of what needs to be implemented into the project in order to create a successful authentication system.

2.2.1 RSA SecurID:

RSA SecurID is a mechanism developed by American computer and network security company, RSA Security LLC. RSA SecurID is the de facto way of deploying two-factor authentication in corporations and today it commands over 70% of the two-factor authentication market.

The mechanism consists of a login process in which user accounts are linked to a Secure Token which is usually issued by the company. When a user attempts to login to their corporate system they must supply a special onetime pass phrase or key which is displayed on the token. The key changes periodically, lasting for a maximum of 60 seconds time span and is directly linked to the company's authentication server. Periodically changing keys are essential to ensure it is a valid user who is requesting access.

The supplied key is validated against the authentication server and if it matches a valid key on the server the user is granted access.

Codes may also be sent to users via email or text SMS, eliminating the need for hardware token devices.

Advantages:

- Easy to implement.
- Provides many benefits for a small cost making it very cost-effective.
- RSA provides stronger computer and network security.
- Protects from network replay attacks.

Disadvantages:

- Susceptible to man-in-the-middle attacks, if the attacker can block user's authentication to the server during periodical key change, they could potentially access systems.
- Older RSA SecurID tokens have a built-in clock which can sometimes become of out sync with the authentication server's clock. This can make it difficult to use the login mechanism as it will cause key changes to also be out of sync.
- In general, if a user's device becomes missing or stolen they will wait at least one day before reporting it, giving a potential attacker plenty of time to breach the system. (Kaur & Devgan, 2015)

2.2.2 Google Authenticator:

In September 2010 Google introduced a two-factor authentication mechanism for users using Google applications. The mechanism consists of information you know, which is generally a user's login credentials to their accounts, and information you have.

The information you have part of the authentication process requires users to have a smart phone for receiving a code through text SMS or a voice-over-text message, or a preinstalled Google application for generating a one-time code.

This application known as Google Authenticator uses Time-Based One-Time Passwords known as TOTP along with a software-based OTP generation scheme.

In this mechanism rather than generating a one-time pass code from an authentication server the user's device generates a six to eight-digit pass code.

User's log into the system by providing this passcode as well as their login credentials.

Advantages:

- Easy to implement with multiple applications provided by Google and is supported by other companies.
- Provides an added layer of security to users free of charge.
- Can generate tokens offline.
- Provides a robust login to ensure only account owners are given access to their accounts..

Disadvantages:

- The transmission of the seeds to the user's mobile device which are used to generate one-time passcodes are sent in plaintext making them susceptible to passive 'eavesdropping' attacks .
- The seeds and login credentials stored on user's devices are stored in plaintext making it possible for attackers to enrol the same seed on multiple devices. Through a simple SQL query on the correct database attackers could potentially gain access to the same seed online from a stolen device. (Kaur & Devgan, 2015)

2.2.3 TOAST Smartphone App:

TOAST is a smartphone app that retrieves a seed from a server using a secure tunnel, TLS/SSL. This key is stored on the smartphone using a keystroke which is protected by a password. This seed is used to generate all access codes on the device.

TOAST consist of three entities:

1. A client who wishes to login to a system.
2. A server listening for authentication requests.
3. A smartphone app that the client has, which generates unique one-time access codes.

The login process therefore consists of the user's login credentials as well as a onetime six-digit passcode which is generated on the device. TOAST also implements encryption standard algorithms such as Blowfish and SHA-1.

Advantages:

- Easy to implement requiring users to only have access to a smartphone.
- Provides an added layer of security free of charge.
- The authentication system can work offline.
- The seed used to generate passcodes is transmitted through a TSL/SSL secure tunnel and stored on the device encrypted.

Disadvantages:

- The main drawback of TOAST is that it provides no initial authentication setup during the initial transfer of the seed.

2.2.4 OAuth 2.0

OAuth2.0 is an industry-standard authorization protocol and framework for delegating access. It supersedes the original OAuth protocol. It is used as a way for users to give organisations access to their information without giving them their passwords.

There are four roles used to define separate entities in the request:

1. The Client - This is the application or web site that is trying to access the users account or information.
2. The Resource Server – This is where the user's information is stored.
3. The Resource Owner – This is the user who is authorizing access to their data.
4. Authorization Server – This server receives the Resource Owners permission to send access tokens to the client.

When the Client requests access to the user's information the Authorization Server requests authorization from the Resource Owner for access. If the owner authorizes the access a token is sent from the authorization server to the Client. This token is used by the client to access the user's information on the Resource Server. (Darwish & Ouda, 2015)

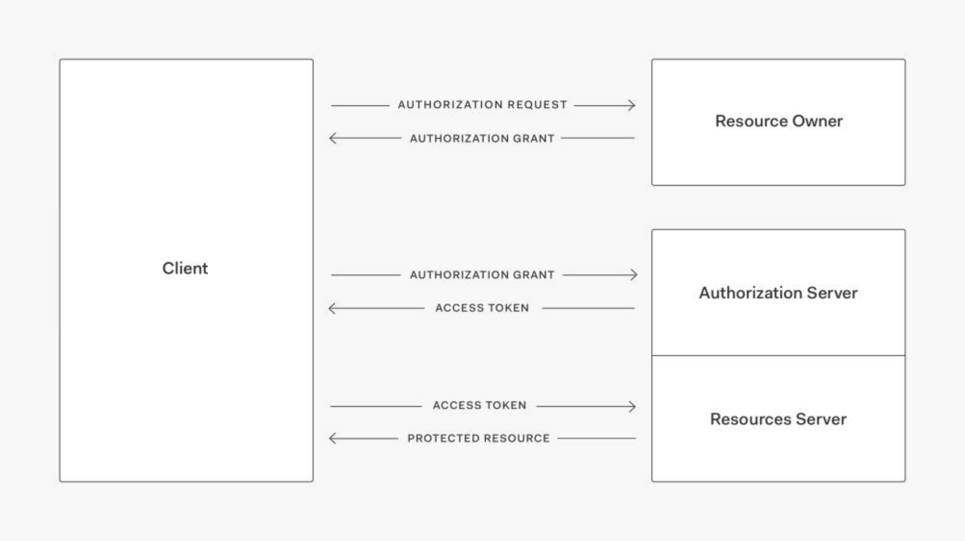


Figure 1 - OAuth Basic Flow("OAuth 2.0 The Complete Guide," 2018.)

Advantages:

- If implemented correctly, it provides a secure way for users to allow multiple organisations access to their information.
- Very user friendly.
- Can be implemented on an array of different systems and technologies.

Disadvantages:

- Can produce security beaches if the protocol is not properly implemented.
- Because web servers store refresh tokens their device's storage capacity may be exhausted if there are too many requests for tokens to be stored.

2.2.5 Image Authentication:

Another way of providing two-factor authentication is to use image authentication to generate one-time passcodes. In this authentication mechanism users are provided with a 'shared secret' key. This shared secret key is a piece of data known only to users trying access the system. The key may be shared with users before they login to the system, a pre-shared key, or it can be created when user's attempt to login.

An Image Authentication mechanism may work as follows:

1. Upon initially registering for the system, users are prompted to select a set of images which can be easily memorized. Examples of image sets are cars, landscapes, scenery or something related to the system's owner i.e. a company secret.
2. Every time the user attempts to login to the system they are provided with a set of randomly generate images.
3. Successful authentication occurs when the user correctly identifies the 'secret' images.

Advantages:

- The human mind is more adept in recalling previously seen images rather than other 'shared-secret' keys such as a text-based key.
- University of California Berkeley conducted a study which found that Image-based authentication systems are more user friendly than text-password systems.
- Prevents social engineering attacks as describing image-based passwords is more difficult than regular password such as text passwords.
- Makes it difficult for attacks to gain access to systems using brute force attacks.

Disadvantages:

- Harder to implement and deploy as it requires stronger computer performance on systems and running environments.
- Some images may have a higher probability of being picked by users upon registering. These are known as hotspots. If an attacker could accurately predict these hotspots based on human nature, they could potentially gain unauthorized access to systems, undetected.

There are many different approaches to providing a multifactor authentication solution. While RSA offers a server-based two-factor authentication solution for providing TOTPs, Google Authenticator and TOAST are providing a solution in which the client generates the TOTPs.

Other organisations are opting to provide alternative approaches to authentication such as the use of Access Tokens with OAuth or an alternate secret key approach as we see with Image Authentication.

There are benefits and drawbacks to all authentication methods which are outlined above. However, the key component in any authentication mechanism is to strike a balance between offering a high layer of security and a fluid user experience.

The user authentication method dominates the users experience while trying to access a system. Introducing quirky password combinations of uppercase letters, digits and symbols leads to a bad user experience. This may lead to poor system security as many users opt for convenience rather than security. Therefore, an authentication method should be as non-intrusive as possible on the user. the introduction of simple authentication methods such as the addition of smart cards to an existing two factor authentication solution can provide an essential added layer of security while maintaining a fluid user experience.

2.3 Technologies Researched & Selection:

Multiple possible technologies may be considered for the development of the application. These include mobile operating systems, databases, scripting language and front/back-end frameworks.
A selection of technologies will be made based on the requirements of the application.

2.3.1 Operating Systems:

2.3.1.1 Android:

Android is a mobile operating system developed by google and initially released on 23 September 2008. It is based on a modified version of the Linux kernel with its design primarily focusing on touchscreen use on mobile devices.

Applications are primarily written in the Java programming language using the Android software development kit to further extend the functionality of devices.

Android is incredibly easy to customize with most major mobile device manufacturers running their own customized version of the operating system on their devices.

Android Studio, an integrated development environment (IDE) was released by Google in December 2014 and is the primary IDE for Android Application development.

It is based on the ItelliJ IDEA, developed by JetBrains. It allows developers to benefit from the full power of ItelliJ IDE as well as many other unique features such as creating views and GUI's, using drag and drop interfaces all without the developer having to directly use XML. (Schmidt, 2016)

2.3.1.2 iOS:

iOS is a mobile operating system developed by Apple Inc and initially released in June 2007. It runs exclusively on their devices allowing users to interact with their device. After Android, it is the second most popular mobile operating system globally.

Applications in iOS can be written in several languages however nowadays Apple are pushing developers to use Swift and Objective-C, Apple's programming languages.

Xcode, an IDE for macOS is the primary IDE for iOS development. (Currents, Examiner, Spruce, & Publications, 2014)

2.3.1.3 Android Vs iOS:

This project's mobile application will be developed as an Android application for several reasons.

The first deciding factor for choosing to develop an Android application is the market share of the mobile OS. Android currently share 85% of that market worldwide, while Apple currently share 15%. Android's market share will grow in the coming years whereas Apple's is predicted to decline. This will mean that the mobile application will be compatible with much more devices. ("IDC - Smartphone Market Share - OS," 2016)

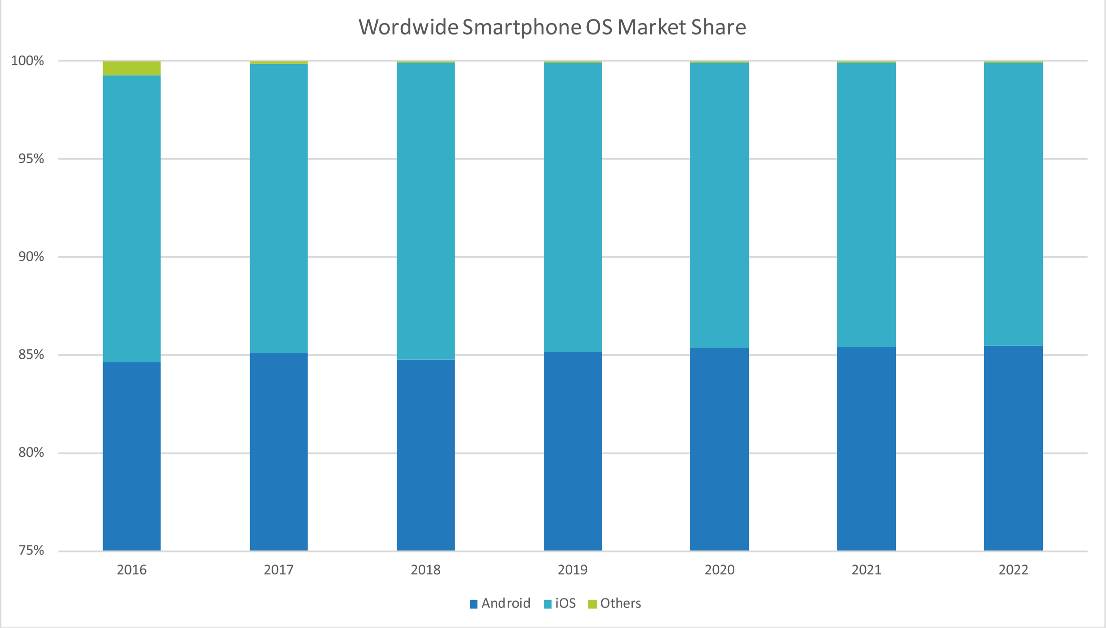


Figure 2 - Smartphone OS Market Share

Applications developed through Android Studio can be deployed on an array of virtual Android smartphones for testing. Xcode also has a testing platform for Apple devices but it is poor in comparison to what Android Studio has to offer.

Android applications are primarily developed in Java and languages related to it. Java is a very common and popular programming language, and it is preferred by most application developers, including myself. ("Android Vs. iOS," 2018)

Although one might argue that Android applications can be more complex to develop than iOS application as they need to cater for a larger target platform with multiple screen sizes and Android versions, for the purpose of this project, focus will be on developing the app for popular current devices which share a common or relative screen size and run on newer versions of Android. This will allow the creation of a more visually appealing application and allow added functionality which may not have been available or supported in older versions of Android. ("Android Vs. iOS," 2018)

Apple's development platform can be rather exclusive, with it being used with specific tools and limitations. However, Android allows for dynamic app development for allowing applications to be developed for almost any purpose.

While Apple have only in recent years introduced NFC to their mobile devices, Android devices have featured it for years. Android Studio provides many built in APIs and libraries for implementing NFC reading. This is essential for the project and it is planned to implement NFC smart cards as an authentication factor in the application.

2.3.2 Databases:

2.3.2.1 Firebase Realtime Database:

The Realtime Database provided by Google's Firebase, is a NoSQL cloud hosted database. It uses JSON to store data which is directly accessible from client-side code.

Firebase data is persisted locally, even when offline. When connection is gained the Realtime Database synchronizes local changes with updates that occurred when the client was offline.

Firebase also provides rules called Firebase Realtime Database Security Rules. These rules define how data should be stored and when data can be read or written. Developers have the option to integrate Firebase Authentication to restrict data access to users. (Esplin, 2016)

2.3.2.2 Oracle:

Oracle DB is a relational database management system provided and developed by the Oracle Corporation. Globally, Oracle DB is one of the most trusted and widely used relational database engines.

The system is built on a relation database framework. In this framework objects may be directly accessed by users on the client or server side through structured query language (SQL).

Oracle provides a fully scalable relation database architecture and robust security. Splitting architecture between logical and physical, physical structure may be added without affecting data or users. Using this architecture also means that no single point of failure can cause the database to crash as the networked storage scheme means that any failure is localized. Therefore, it is often employed by large global enterprises to manage data across networks.

Oracle runs on most platforms including Windows, Unix and Mac OS and databases can be deployed on the Oracle Database Cloud Service. ("What is Oracle Database (Oracle DB)? - Definition from Techopedia," n.d.)

2.3.2.3 MariaDB:

MariaDB is an open source relation database management system that can be used as a compatible replacement for MySQL database technology. It was developed by some of the original developers of MySQL which is widely used. Therefore, MariaDB is one of the most popular database solutions in the world.

MariaDB is based on SQL and supports many features such as ACID data processing, JSON APIs and multiple storage engines.

MariaDB runs on Windows, Linux and MacOS, while supported programming languages include C++, Java, Python and others.

MariaDB may be deployed on the cloud and is supported in Amazon RDS and Microsoft Azure. (“What is MariaDB?,” 2018)

2.3.2.4 PostgreSQL:

PostgreSQL is an open source relation database management system developed by a worldwide open source community of volunteers.

PostgreSQL has a strong reputation due to its reliability, architecture and data integrity.

It is ACID compliant and runs on all major operating systems. It allows uses to define data types and build custom functions with code written in different languages without the need to recompile the database.

While PostgreSQL attempts to conform with standard SQL, many of the features required by the SQL standard are supported so long as the conformance does not lead to poor architectural decisions.

PostgreSQL is completely free and open source. (“PostgreSQL: About,” 2018)

Amazon Web Services support many cloud deployments of PostgreSQL.

2.3.2.5 Selection (Firebase):

The chosen database for the project is Firebase. Firebase has multiple components that are essential in helping to create a strong security application. As it is a real-time database Firebase will provide the application with data persistence across all clients. Maintaining database persistence while being deployed on the cloud is essential for any applications security.

Although most of the authentication methods will be developed, Firebase includes many Libraries and SDKs for authenticating client users. This eases the process of integrating authentication mechanisms into the system.

The application is going to deal with access control and therefore the chosen database needs to support storage of an array of different file types to demonstrate the access control implemented on the system. Firebase allows for multiple file types to be stored such as images and videos.

2.3.3 Dynamic Web Applications:

2.3.3.1 Python:

Python is an interpreted, interactive object-oriented programme for languages. It incorporates modules, exceptions, high level data types and classes. It provides powerful functionality with a clear syntax and has interfaces to many systems calls and libraries. It may also be used as an extension language for applications needing a programmable interface.

Python libraries cover areas such as string processing, internet protocols and operating system interfaces.

Python runs on many Unix variants, MacOS and Windows. ("General Python FAQ — Python 3.7.1 documentation," 2018)

2.3.3.2 JavaScript:

Java is an interpreted programming language that allows users to implement complex features to webpages such as interactive maps, animated graphics and video boxes.

There are many API's built on top of JavaScript which allow users to add further functionality.

The browser's JavaScript engine is used to executed JavaScript, after the HTML and CSS of the web page have been assembled and put together. JavaScript dynamically generates this new content inside the browser and displays it to the client. This makes JavaScript a client-side scripting language.

JavaScript runs on many different operating systems, Windows, Unix, MacOS and more. ("What Is Java, and Why Is It So Great?," 2016)

2.3.3.3 PHP:

PHP is a general-purpose scripting language that is primarily used for web development.

PHP code can be embedded into HTML. PHP code is executed on the server generating HTML which is then sent to the client.
PHP supports a wide range of databases and makes writing database-enable web pages incredibly easy.

PHP is extremely versatile and offers many features to programmers such as server-side scripting, command line scripting and can also be used to write desktop applications. It may also be combined with procedural and object-oriented programming. Unlike HTML PHP can output images, PDF files and flash movies and can output any text such as XHTML, XML and others.

PHP can be used on all major operating systems including Linux, Windows, MacOS and more. ("PHP: What can PHP do? - Manual," 2018)

2.3.3.4 .NET:

.NET is a cross-platform open source development platform for building web, mobile, and desktop applications.

.NET apps can be written in C#, F#, or visual Basic.

- C# is a modern object-oriented programming, with its roots being in the C family it is like C and C++, and also Java and JavaScript.
- F# is an open-source functional programming language deployable on multiple platforms. It includes imperative and object-oriented programming.
- Visual Basic is a language with a simple syntax for developing type-safe object-oriented applications. (."NET Programming Languages," 2018)

2.3.3.5 Selection (PHP):

The chosen scripting language for developing the back-end is PHP. Using PHP results in faster loading speeds for websites. This is essential as the aim of the project is to provide legacy systems with a security solution. As legacy systems do not have the performance power of a modern-day system, having the fastest loading time is essential. PHP is also flexible for database connectivity, using it as the scripting language will allow connection to different database technologies during the life cycle of the project if required.

2.3.4 Front-End Framework:

2.3.4.1 AngularJS:

AngularJS is a front-end JavaScript framework for dynamic web apps. It is developed by Google with the overall goal of simplification.

Single page dynamic web apps can be developed in a Model View Controller (MVC) code structure with static HTML code being used to build dynamic HTML with the use of JavaScript.

AngularJS provides many libraries that developers can use to create interactive websites without needing to directly touch HTML or CSS code. ("What is AngularJS," 2018)

2.3.4.2 Ember.js:

Ember.js is a front-end JavaScript framework used to build rich complex websites. It provides a solution for developers that contains data management and application flow.

Ember.js uses HTML and CSS at the core of its development model.

Ember uses built in template libraries and rich feature sets allowing users to create robust and complex web apps while writing very little code. (tutorialspoint.com, 2018)

2.3.4.3 Bootstrap:

Bootstrap is a front-end framework created by a team at Twitter. It is significant for faster development of responsive websites and mobile applications.

It is an open source toolkit with a combination of JavaScript, HTML and CSS code to allow users to build a range of interface components. ("About - Bootstrap", 2018)

2.3.4.4 Selection (Bootstrap):

The decision to use Bootstrap as a front-end framework for this project is based on its ease of use and deployment on mobile applications. Due to the persistent growth of mobile devices and their screen sizes, it becomes increasingly difficult for developers to develop applications suitable for all devices. Using Bootstrap allows developers to automatically scale their webpages down to fit on multiple different devices. This is essential if the application is to be deployed on an array of devices.

There are many built in libraries for customizing Bootstrap websites. This will simplify the process of producing a visually appealing user application and fluid user experience.

Bootstrap incorporates many JavaScript components to add further functionality to a website.

Bootstrap may also be easily integrated with multiple frameworks and platforms.

2.3.5 Back-End Frameworks:

2.3.5.1 Zend Framework 3:

The Zend framework is a back-end framework developed by Zend Technologies. It is primarily used to develop web applications and services. It is a collection of professionally contributed PHP packages which provide object-oriented code, enabling users to implement many features.

The framework combines the Model View Controller in combination with the Front Controller Solution.

It works by using a router and dispatcher to decide which controller needs to be run depending on data it receives by URL. The controller then works in conjunction with model and view to create and display the webpage.

Applications built using the Zend Framework can run on any PHP stack, providing they meet specific requirements. The Zend Server is a stack provided by Zend which has been optimized for running Zend Framework applications. ("About - Zend Framework," 2018)

2.3.5.2 Symphony:

Symphony is an open source PHP framework originally developed by SensioLabs and backed by a community of contributors. It was originally inspired by the Spring Framework. Its primary use is for developing web applications.

Symphony focuses on optimizing its performance and today is one of the fasted PHP frameworks available.

It is a highly customizable framework and developers have control over the configuration of almost everything. ("Symfony at a Glance," 2018)

2.3.5.3 Laravel:

Laravel is an open source framework based on symphony and used to develop web applications. The framework aims to ease the development process by creating a fluid web development experience by using elegant and demonstrative syntax.

It eases web project tasks such as routing, sessions and caching. Laravel follows a Model View controller architectural pattern. (Solution, 2018)

2.3.5.4 Selection (Laravel):

The chosen framework for the application is Laravel. Laravel provides system modularity allowing flexibility and variety of use. This allows developers to split different parts of an application into different parts which are relevant to each other.

Although Authentication is the basis of the application and will be developed independently, Laravel simplifies the implementation of authentication on web applications by providing a range of libraries and functions.

Using controllers or direct route declarations, Laravel may also be implemented within many other applications. This will allow the implementation of Laravel into other aspects of the project if it is decided to do so.

The use of caching that Laravel provides will boost the performance of the web application by reducing the time needed to access data.

One of the main advantages Laravel provides is a multiple file system. As a key component of the application is Access Control the ability for the chosen framework to support multiple file formats is essential for testing purposes.

2.3.6 Conclusion:

Technologies have been chosen based on the reasons outlined above. This project will use Firebase to host the database, PHP, along with the Laravel framework as the backend of the system, and Bootstrap and Android Studio for front-end development of the web and mobile application.

2.4 Compliance Issues

The European Union and United States Congress have both introduced laws which directly affect organizations that deal with sensitive public data. These are outlined below.

2.4.1 EU Law:

2.4.1.1 GDPR:

The General Data Protection Regulation is an EU law which came into effect on the 25th of May 2018, replacing the existing data protection framework under the EU Data Protection Directive.

The GDPR increases the obligations and responsibilities for organisation and businesses who control data. The collection and use of personal data must be made fully transparent by the organisation and they must have the ability to demonstrate strong security measures for data processing activities.
The regulation addresses organisations directly in terms of the obligations it imposes and does not require transposition into the laws of a given country. With the introduction of this law, European citizens now have control over their personal information which creates a high-level requirement for privacy protection in Europe. This shows that high level security and access control solutions are essential for organisations who want to adhere to the new laws set out in the GDPR. (Seo, Kim, Park, Park, & Lee, 2017)

2.4.2 American Law:

2.4.2.1 Sarbanes-Oxley Act:

The Sarbanes-Oxley Act, otherwise known as SarbOx is an act to protect investors by improving the accuracy and reliability of corporate disclosures in accordance with the securities laws. The bill was enacted as a reaction to major corporate and accounting scandals. It adds criminal penalties for certain misconduct such as the wilful destruction of evidence to impede a federal investigation. The SarbOx Act also requires technical safeguards and IT controls to protect and guarantee the safety of public information by introducing Privacy Rules. All sensitive data must be protected with strong security protocols and privacy rules that organizations must be compliant with, limiting the use and disclosure of a person's information. They also must provide an audit trail of who has accessed what data and when, and then finally they must dispose of data properly once its retention period is up. For SarbOx data permanence is particularly important, but it still falls under the data privacy umbrella. All data must be proven that it has not been altered from the time it was stored to the time it was retrieved or used. By implementing access control into systems this can be ensured. SOX violations have fines and jailtime associated with them and therefore it is essential again that organisations implement strong security protocols such as multifactor authentication and access control.

As stated in the laws above, organizations that handle public sensitive data and private information are required to implement strong security and data processing measures and protocol on their systems.

These two laws demonstrate why using strong authentication mechanisms along with access control is an essential requirement for any organisation in today's world.

2.5 Studies Relating to Project Domain

There have been studies done relating to the domain of this project. Here is a review of some of them.

2.5.1 How Privacy Flaws Affect Consumer Perception

In this paper the authors examine how publicized instances of privacy flaws and data breaches of three real-world incidents are perceived by consumers. They investigate how it affected the consumer's future purchasing behaviour and how much trust they had in the company.

In recent years the increase in the level of personal information stored by organisations has drastically increased, this has led to privacy breaches becoming more common. When a breach occurs millions of records can be lost leading to identity theft and related crimes.

Both the United States and EU have responded with stricter laws that require organisations to notify individuals when their information has been compromised. This has increased the public awareness of data breaches, an issue that in the past could have been hidden from public knowledge. While the introduction of these laws reduces the harm after the incidence of a breach, allowing individuals to take precautions to reduce damage, there is a reputational loss for organisations involved.

The authors examined the consumer perceptions of Apple, Facebook and Sony after each company experienced privacy breaches. They surveyed 600 individuals in 2011 and again in 2012, before and after the breaches.

Apple iOS Location Data Storage: When Apple released its IOS4 operating system a privacy vulnerability was discovered. User locations were logged in an unencrypted file regardless of whether location services were enabled.

Facebook's Device Privacy Settings: Facebook has suffered many privacy flaws since its popularity began to soar. This has been widely publicized by the media in recent years with the company being constantly under fire for the mishandling of user data. One of the first publicized major privacy breaches was in 2009 when Facebook changed their user privacy policy. The new policy required users to utilize a privacy transmission tool. By default, this tool exposed previously protected profile information and made it available to the public for consumption.

Sony's PlayStation Network Breach: In April 2011, Sony announced that 77 million accounts on the PlayStation Network had been breached, after which followed another breach in which over 100 million accounts across Sony systems were compromised. This led private user information to be leaked and although it was speculated that credit card information had also been leaked there was no evidence to prove so. Sony employees were directly responsible for the breach after posting on an open forum that Sony systems were running an out-of-date version of the Apache Web Server which had not been patched for some time. This meant that aspects of the systems were running without any firewall.

In the days following each of the data breaches above, each company's stock had a significant decrease in value.

The results of the survey conducted following the breach by the authors were as follows:

- 59% of the individuals perceived Apple as less trustworthy with 57% stating they are less like to purchase an Apple product in the future
- 67% of the individuals perceived Sony as less trustworthy with 46% stating they are less likely to subscribe to the PlayStation Network in the future
- Facebook seen smaller decline in perceived trustworthiness by their users. However, 9% of individuals refused to use Facebook after learning about the privacy flaw.

The general findings are that consumers trust companies less following privacy breaches. The strongest effect of a privacy breach is on potential customers.
Both the individuals surveyed, and the authors agree that all three companies need to implement stronger security measures to regain trust.

It is evident that the use of a multifactor authentication system along with access control can drastically reduce the risk of a data breach. (Afroz, Islam, Santell, Chapin, & Greenstadt, 2013)

2.5.2 Personal Information in Passwords and Its Security Implications

In this paper the authors explore the security implications involved in using personal information in passwords. By analysing several leaked datasets, the authors investigated the extent to which user's use personal information while creating passwords. After analysing the dataset, they examine and list practices for users to mitigate the unwanted link between personal information and passwords.

It is clear today that text-based passwords remain the most common form of authentication, one which is not believed to be replaced in the foreseeable future. For years researchers and organisations have attempted to create a better solution for authentication, however the unwanted burden that they introduce to users with their authentication methods outweighs the benefits they bring.

The human mind has memory limitations making passwords created far from random system generated passwords. Humans choose simpler passwords because they are easier to remember. These range from simple dictionary words to keyboard strings such as "qwerty2 and "123456". This leaves them vulnerable to brute force attacks.

The researches of this paper chose a Chinese railway dataset which was leaked in 2014 to perform their study of password semantics. This dataset included the passwords and personal information of approximately 131,389 users.
Researchers found that the average length of a password was 8.44 characters with the dominant passwords being trivial number strings, keyboard strings and phrases such as "I love you". Furthermore, they found that 60% of passwords in the dataset contained personal information registered by the user, such as birthplace, last name and date of birth.

The authors of this paper propose the Distortion Function to increase password security without compromising its memorability. The purpose of using the distortion function is to increase password security by removing the ability to use password semantics to crack the password. Users need only remember an original password along with a chosen distortion function.

The authors conducted a proof-of-concept study to show the effectiveness of using a distortion function. The first function they implemented replaced each character in the password with another character like the Caesar Cipher. The second function added extra characters between any pair of characters in the password. These functions were able to mitigate the use of semantic patterns and personal information to crack passwords, without significantly sacrificing password usability.

In viewing this paper, it concluded that password-based authentication systems needed an added layer of security. This project will resolve the problem by using multifactor authentication methods along with passwords however, it must be ensured that the multifactor authentication solutions introduced don't affect the user experience and become a burden rather than a benefit. (Li et al., 2017)

2.5.3 Authentications and Delegation with Smartcards

In this paper the authors recognised the problem that distributed systems face with regards to authentication and access control as they appeared. These systems have many problems associated with them, from users being unable to encrypt and decrypt on client-side applications, to the lack of ability to control access to nodes after mutual connection.

The authors recognise the benefits of having a smartcard for authentication purposes and the functionality it can provide. The authors discuss a range of smart-cart protocols and analyse the assumptions and guarantees they offer.

They propose a simple approach to smart card authentication and access control. The weakness of a user's password may be eliminated with the introduction of simple smart cards with a small amount of read only memory along with a personal identification number (PIN). Much like an ATM when a user attempts to authenticate themselves to access a system they must provide their smart card along with a PIN. Using a PIN mitigates the user's account succumbing to a security breach if the card becomes lost or stolen.

For the use of a smart card to become a valid method of authentication the authentication server which they are interacting with must be secure. The authors discuss using encryption methods when transmitting data to and from the server along with public-key cryptography.

Once authenticated a simple approach to access control may be implemented. If a smart card had the ability to sign timestamp certificates the system could delegate the user's authority for a limited time, for example only allowing access to remote files for the next hour. This timestamp would provide critical protection against replay attacks.

While currently there are no cost-efficient smartcards available with the functionality of a clock and battery, the proposed approach for this project which involves the use of Near Field Communication (NFC) smart cards along with a mobile device could provide a means to for the user to sign timestamp certificates allowing for this access control method to be implemented.

The authors conclude that the use of smart cards offering a trade-off of some kind is inevitable. This could be an added cost for an organisation, a burden on the user experience, or gaining the trust the user needs to place in the system. However, they agree that the use of smart cards as a form of authentication is viable and majorly improves the security of a system. (Abadi, Burrows, Kaufman, & Lampson, 1993)

2.6 Resultant Findings and Requirements

The technologies outlined above will be used for the development of the system.

Through research and stakeholder engagements, the best authentication mechanisms to use that provide a layer of security without compromising the user experience were discovered. This is an essential part of the project as for a multifactor authentication mechanism to be successful it cannot be a burden on the user experience.

Through research and engagement with project stakeholders, the best approach to access control was found. A detailed description is outlined in the *Approach and Methodology and Design* sections.

3. Approach and Methodology

In order to choose a software methodology which is most suitable for this project, multiple methodologies must be reviewed. The chosen approach for this project is also outlined.

3.1 Methodologies Researched

3.1.1 Feature Driven Development:

Feature driven development is a client-centric software process, with client being used to represent what Agile Modelling refers to as project stakeholders.
FDD features are an important aspect of the development process. A feature is a small function that preforms an action on an object to get a result.
Feature Driven Development can be broken into four stages:

- **Develop an Overall Model -** The first part of FDD is to develop the overall model. This involves identifying the domain and technologies that the system will use throughout the project.
- **Build a Features List -** The second part of FDD is to build a Features List and group them based on their subject area and function.
- **Plan by Feature -** Next each feature is planned out before development starts.
- **Design by Feature/Build by Feature -** Once development starts features are designed and built. These two steps involve modelling, programming and testing. (“Feature Driven Development (FDD) and Agile Modelling,” 2014)

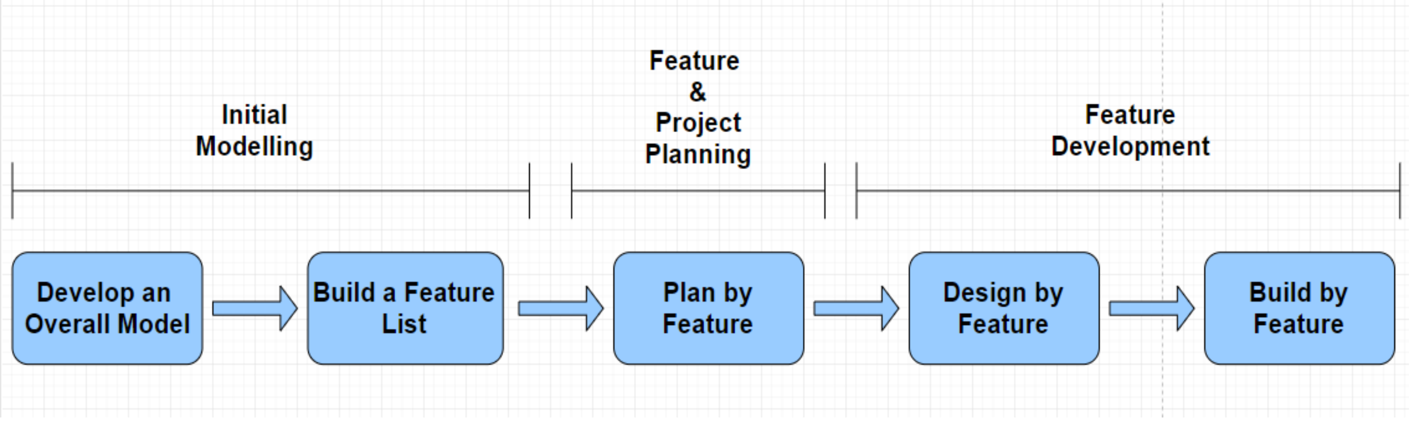


Figure 3 - Feature Driven Development

3.1.2 Waterfall Model:

The Waterfall Model is a software development process which emphasizes a logical progression of steps which are taken throughout the software development cycle down cascading steps, much like an incremental waterfall.

The Model is broken into six stages:

- **Requirements** – The potential requirements of the application are analysed and documented. This initial stage serves as the basis for all future development.
- **Analysis** – During this stage the system is analysed to generate models based on business logic. These models will be used in the application.
- **Design** – This stage covers the technical requirements of the project such as what programming language will be used, what services are needed, what technologies and data layers will be used, etc.
- **Coding** – The source code is written implementing all model and business logic.
- **Testing** – Application is tested and issues that require attention or need to be resolved are reported and documented. Sometimes a repeat of the coding stage is necessary.
- **Acceptance** – The application is ready and is deployed to a live environment.

(Andrew Powell-Morse, 2016)

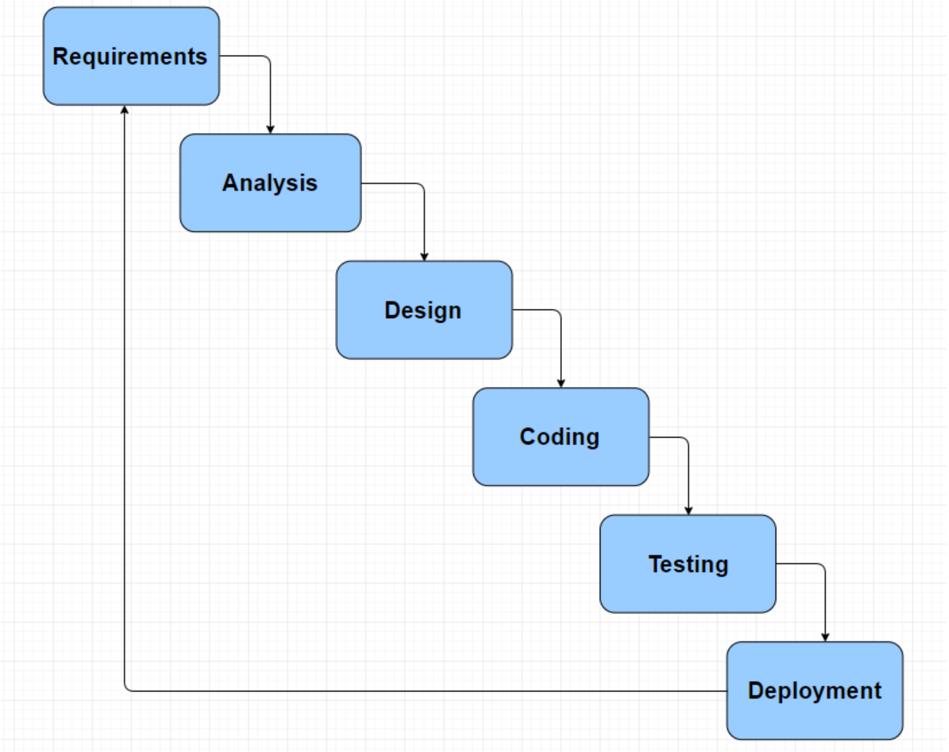


Figure 4 - Waterfall Model

3.1.3 Extreme Programming:

Extreme Programming (XP), is an agile software development methodology that supports frequent releases of application in short development cycles. It generally improves software quality and allows the development team to respond to changing requirements.

XP can be broken into five stages:

- **Planning** – Rather than a lengthy requirements document, requirements are defined based on the functionality needed, the value of that function in the business and the priority of the feature.
- **Managing** – Developers work collaboratively and effectively to avoid errors.
- **Designing** – Simple parts of application are designed first as they take less time than the complex solution after which, other parts are designed based on their level of difficulty.
- **Coding** - Coding of application. Functionality may be added, bugs may be fixed, and the application may be refactored if needed.
- **Testing** – Application is tested, and bugs are fixed before application goes live.

("What Is the Extreme Programming Methodology?," 2015)

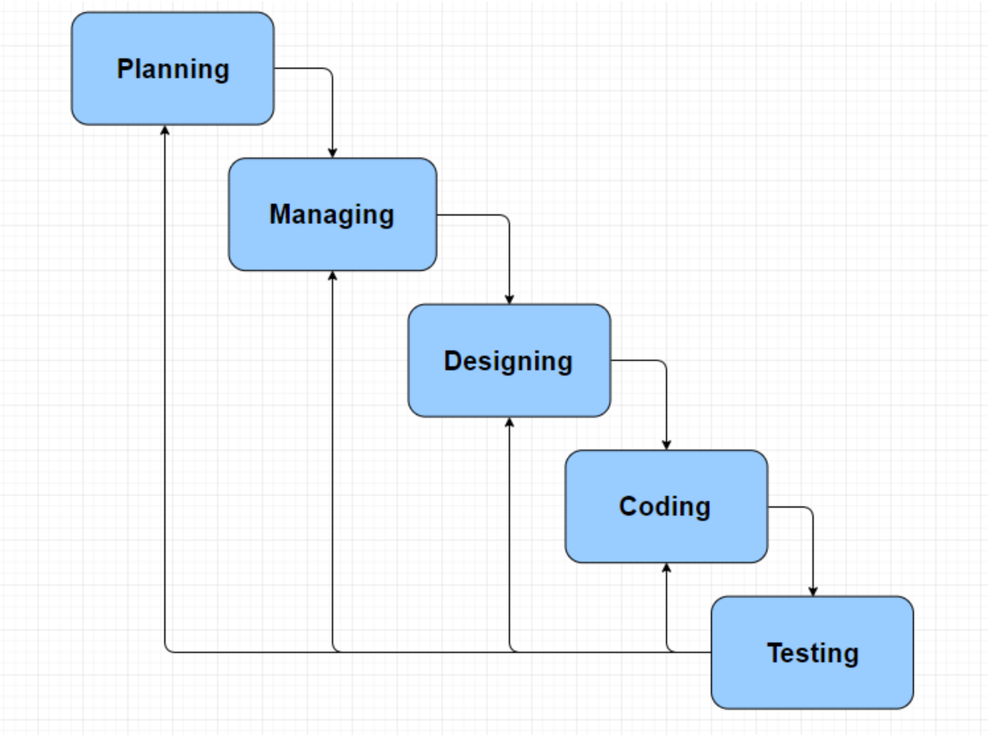


Figure 5 - Extreme Programming

3.1.4 Scrum Methodology:

The scrum methodology is an agile software development methodology with the goal of delivering new software features every 2-4 weeks. It generally produces higher productivity, better-quality products, improved customer satisfaction and better team dynamics.

Using a scrum methodology approach development can adapt to current conditions rather than predicted conditions. This allows developers to mitigate common pitfalls such as compromises on software quality, underestimation of time, resources of cost, and much more.

The scrum is broken down into five stages known as ceremonies:

- **The Sprint** – Is a period during which specific work is complete and made ready for a predefined deadline. Sprint usually lasts 2-4 weeks but can be shorter.
- **Sprint Planning Sprint** – Planning meetings are predefined lengths of time to discuss how black log parts of the application will be delivered and how the work will be achieved.

- **The Daily Stand-up** – Is a short meeting in which a team member quickly covers progress since last stand-up. All planned work which will be done, and any difficulties are also discussed.
- **The Sprint Review** – This is where the team demonstrates the work that has been completed during the sprint. The customer can check the work and give feedback.
- **The Retrospective** – This is the final team meeting to determine what went well and what didn't go well during the sprint and how it can be improved for the next sprint. It affords the team the opportunity to identify strategies so that the application may continuously improve.

These stages are repeated until software is delivered.

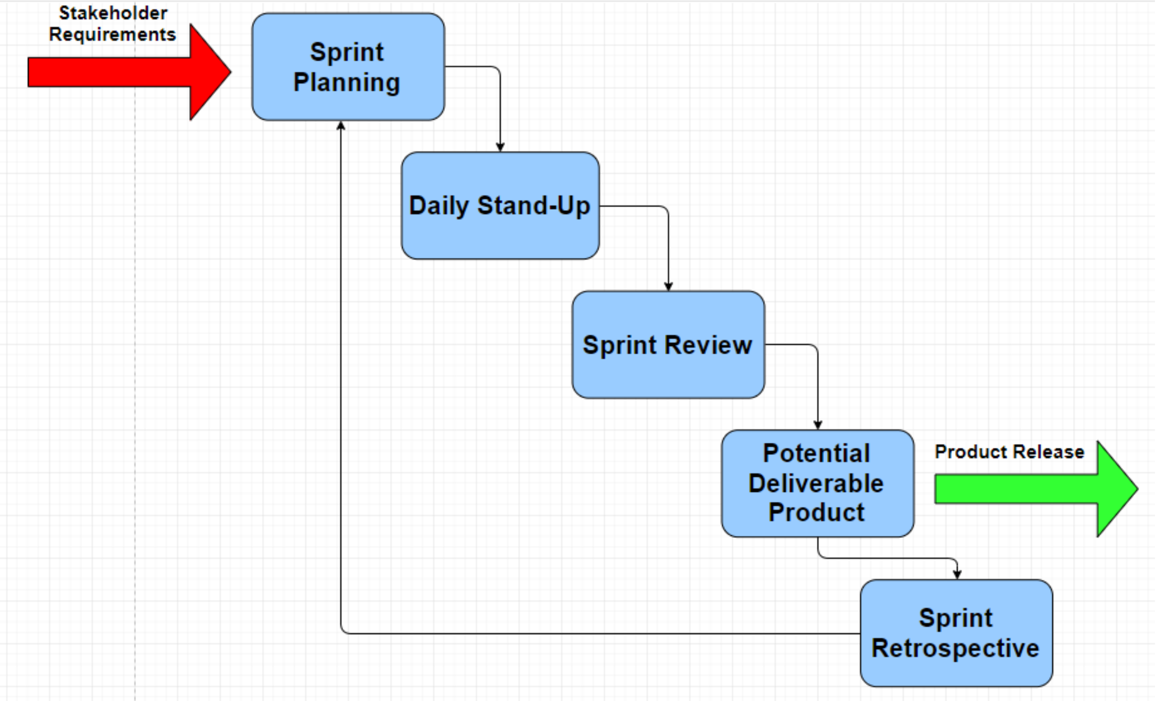


Figure 6 - Scrum Methodology

3.2 Chosen Methodology:

3.2.1 Feature Driven Development

Feature Driven Development was chosen as the software methodology to use for the project. The project consists of two main features the authentication mechanisms and access control. However, there are many sub features such as each authentication factor, client application etc.

Using FDD will allow the development of a feature rich application with features that rapidly evolve throughout the lifecycle of the project. The need for the features to be able to evolve is crucial as the security solution which will be developed is also going to evolve as new technologies are added and functions are linked together.

The feature list will be built with security taking top priority. Security related features will be developed first as they are the most important part of the project. The client-side and mobile application will be developed once the security parts of the application are completed to a high enough standard.

Design and coding are planned to be delivered in phases. The application is predicted to evolve greatly over the lifecycle of the project therefore it is not practical to have all the design completed upfront.

3.3 Chosen Approach

The chosen approach for the development of the project is to follow a Feature Driven Development approach. Using this approach will allow the engagement of stakeholders throughout each step of the project as features are built, developed, and tested.

The first step to the approach was to build an overall model for the project. This model consisted of a detailed description of the project and how it will work, technologies selected, issues and risks, project planning, and testing approaches.

The model was then presented to the stakeholders for review and feedback was received (see appendix A).

A detailed description on how the system will operate is given in the *Design* section below.

The next step was to build a features list. Through engagement with the stakeholders, some of who are experts in the area of cyber, a list of features was built. This list included the key features of the project, multiple authentication mechanisms, access control protocols, and client-side applications.

The authentication feature list was built by performing relevant research into the domain of security authentication in order to gain a deep understanding into what the project required and engaging with a stakeholder who is an expert in the field of security authentication (see appendix A).

Through engagement with another stakeholder who is the co-director of an international company who deals with private customer data, the access control features were built based on general administrator needs to eliminate privacy flaws. This stakeholder also gave an insight into what features are needed to create a fluid user experience on the client-side applications (see appendix B).

Next, began the development of a prototype which would be presented to the stakeholders. A detailed description of this prototype is given in the *Prototyping and Development* section below.

Once the prototype and an interim report of the project have been developed and submitted, design and development of features will continue as well as new features which need to be built.

As soon as features are built each one will undergo a testing process to eliminate security flaws and errors. The testing process that will be used is detailed in the *Testing* section below.

4.Design

4.1 Technical Architecture Diagram

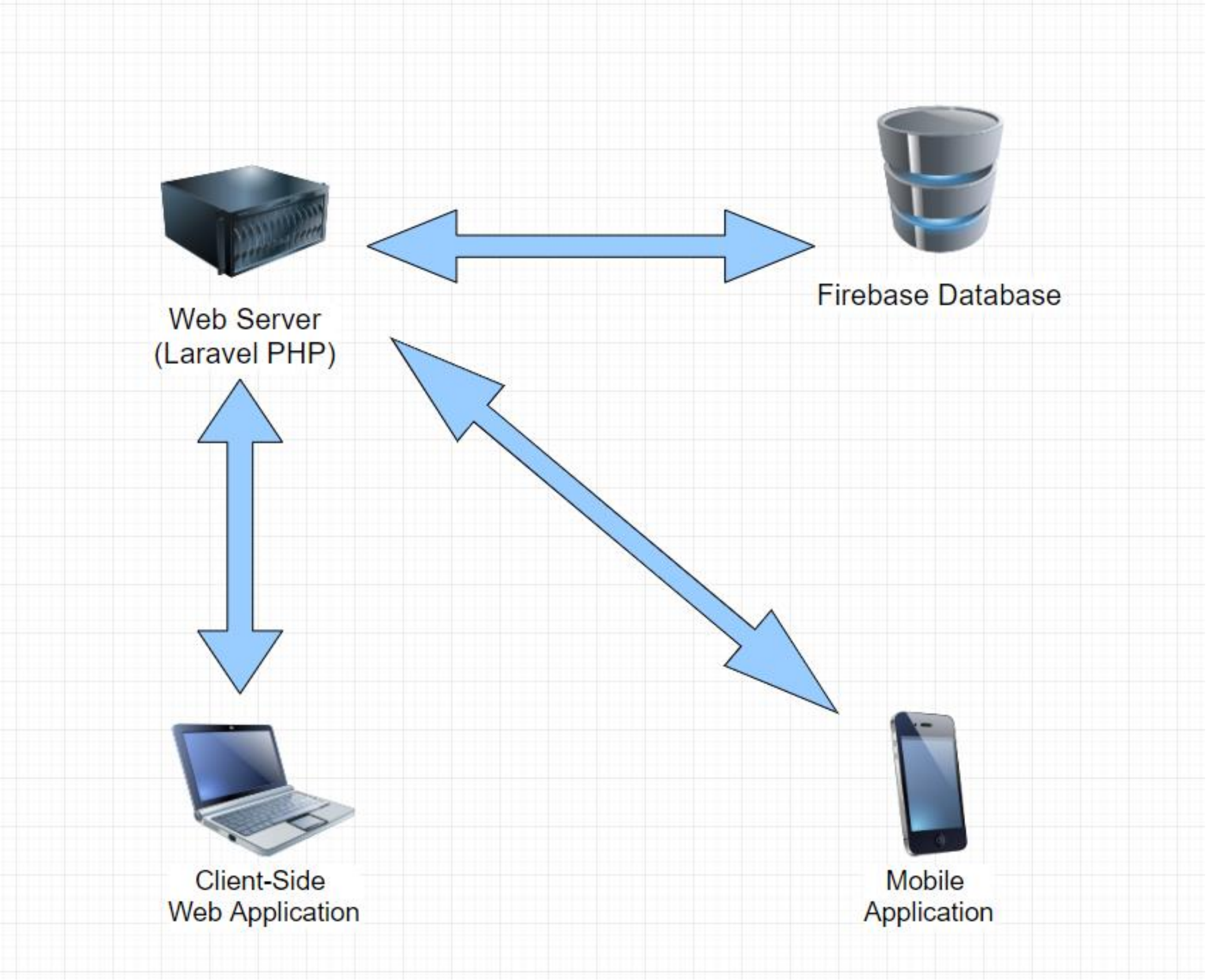


Figure 7 - Technical Architecture Diagram

4.2 Textual Description of System's Operation and its Benefits:

The complete version of the system which is intended to be developed will provide a secure solution for security authentication as well as access control without being a burden on the user experience.

When the user attempts to login to the system with their login credentials, they are required to provide a server generated Timed One Time Passcode (TOTP).
This TOTP is generated by the server application which is using PHP and sits on the Laravel framework.
In order to request the passcode, the user must scan their NFC card to their smart card. This the NFC card's ID number is compared with user data stored on a Firebase database and if it is registered to the user attempting to login the TOTP is displayed on the smart phone.
Once the TOTP is displayed the server initiates a countdown of a defined period, typically 60 seconds. If the user doesn't provide the generated TOTP to the web application in time the generate TOTP becomes invalid and a new one is issued or requested. If the user provides the correct password which is validated against the Firebase database, they are granted access to the system. This system is a web application developed using Bootstrap and will grant the user access to certain files and functions on the system based on their trust level which is predefined on the database and read when they login to the system. If a user does not have access to a system or a part of the system, they may request access from the administrator. This provides the user with a secure authentication mechanism to the server along with access control adding two layers of security.

For the access control part of the application there will be an admin user. This administrator will follow the same login process of the application. Instead of displaying the general user web application, they will be redirected to an admin dashboard, based on their trust factor. This dashboard will allow them to allocate and change a user's trust level, review a log of what users logged in, what files they viewed and also grant user access further into the system. Using the server clock, the administrator may restrict access to the system for users within a given time frame, for example 9.00-17.00 Monday – Friday. All access control logic will be written on and controlled by the server. This provides the system with another layer of security allowing the administrator to control further access to the system and access to the system outside working hours.

Data which is being read and written to the database will also be encrypted end-to-end to enhance the overall security of the system. This also provides another layer of security to the system as well as providing data privacy.

Depending on time constraints, it will be attempted to implement other authentication methods such as image authentication and "*known secret-key*" authentication. It will also be attempted to implement an access control function which will allow the administrator to choose what authentication methods the user needs to provide to log into the system using a simple mechanism on the admin dashboard.

4.3 Other Design Documents

4.3.1 ERD Diagram:

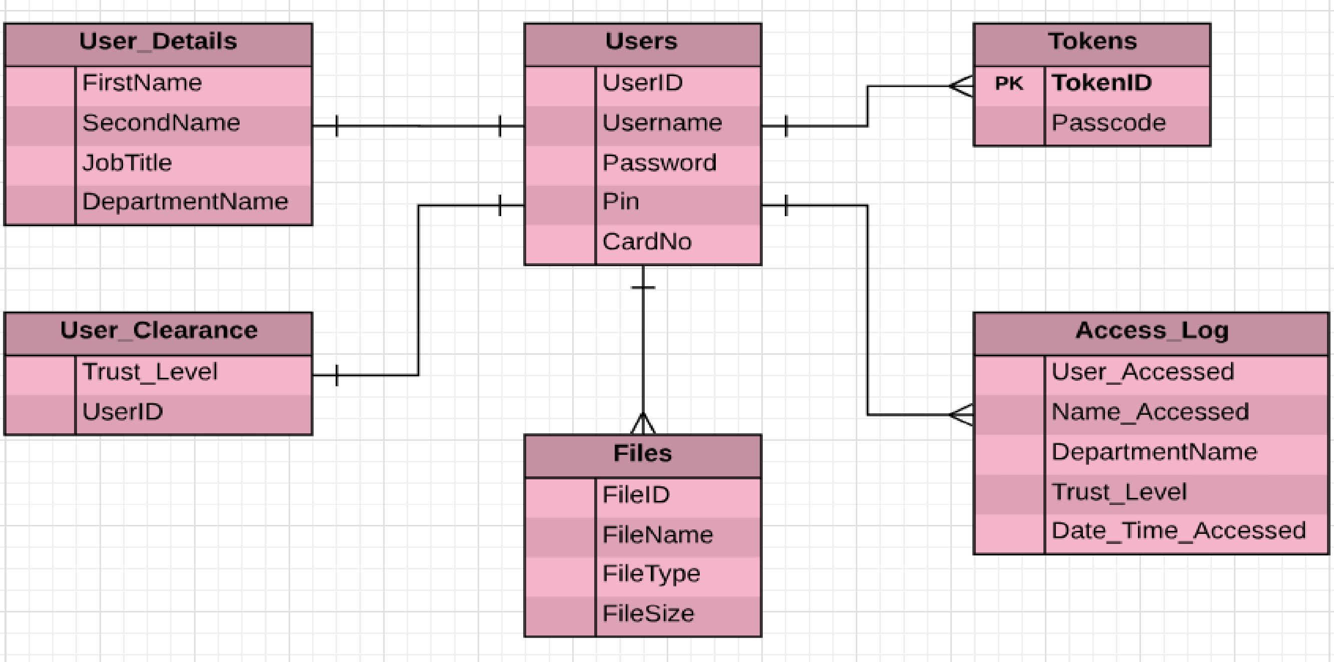


Figure 8 - ERD Diagram

4.3.2 Use Case Diagram:

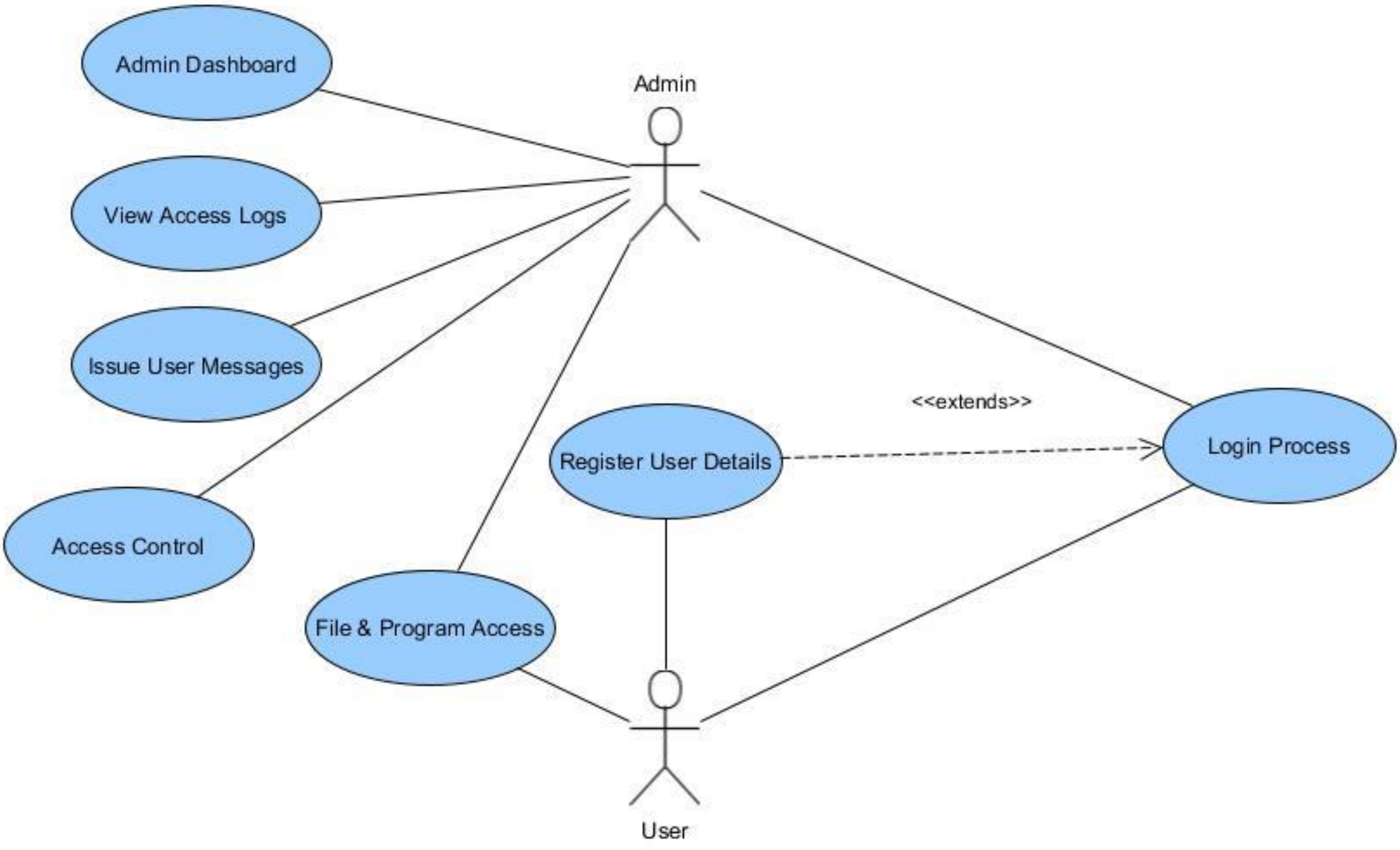
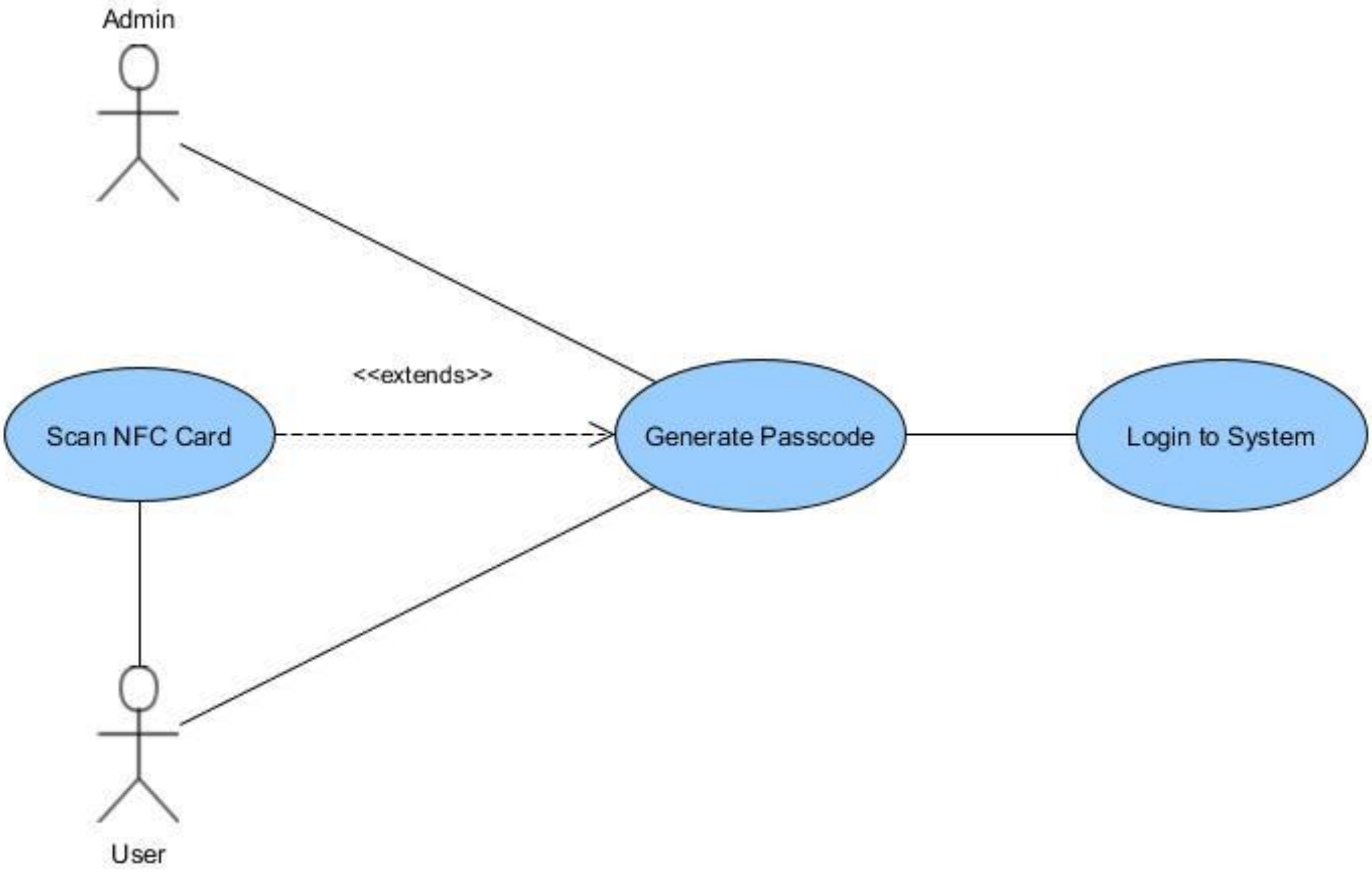


Figure 9 - Use Case Diagram

The User can Register to the system, Login to the system, and Access Files and Programs on the system.

The admin can Login to the system, View the Admin Dashboard, View Access Logs, Issue User Messages, and Control User Access to the system.

4.3.2.1 Authentication Process Use Case Diagram:



The authentication process consists of the user scanning their NFC card to a mobile application. This generates a Time-Base One-Time Passcode which is then used to login to the system along with login credentials.

4.3.3 Class Diagram:

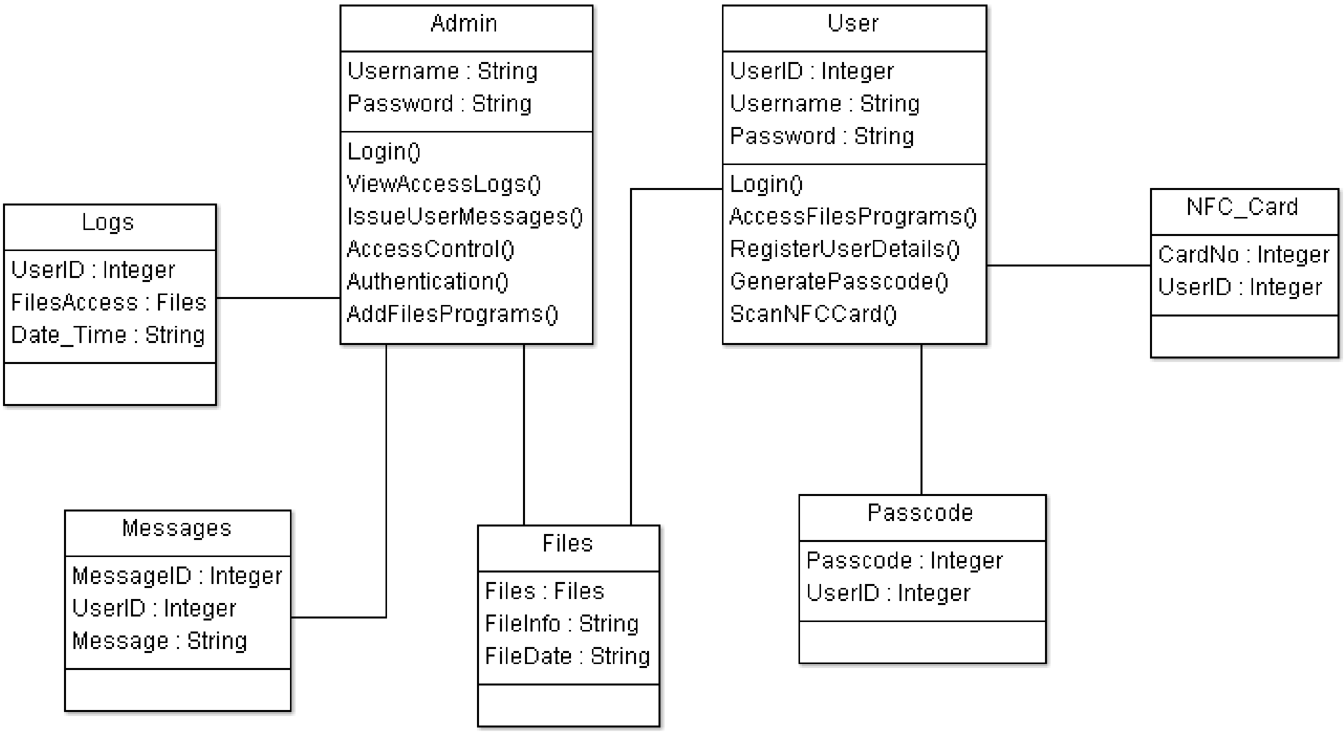


Figure 10 - Class Diagram

4.3.4 Sequence Diagram:

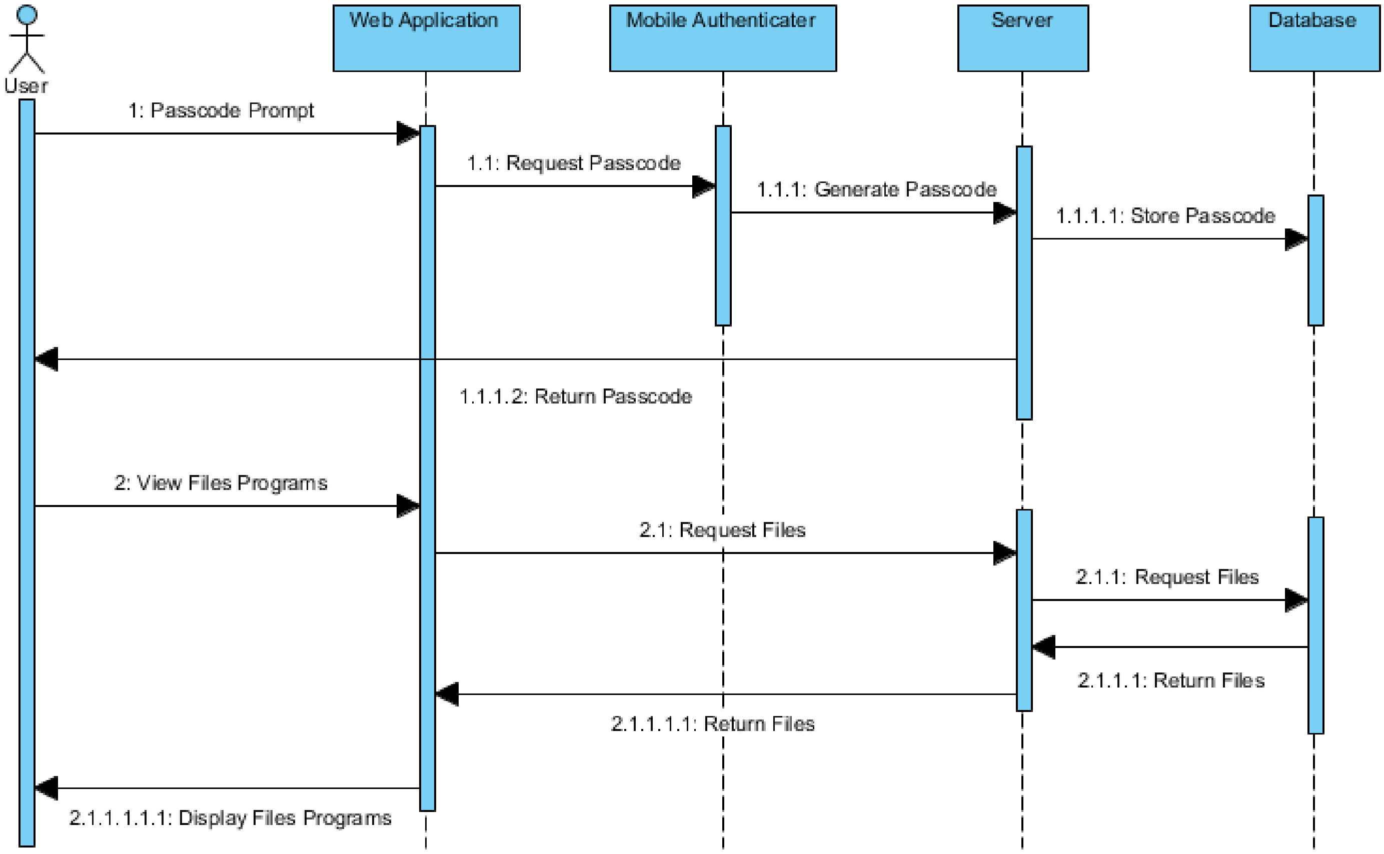


Figure 11 - Sequence Diagram

5. Prototyping and Development

So far, the bulk of the project has been the in-depth research into to the domain of the project in order to get a clear understanding of how the system will be designed and developed.

Focus has been on the development of a proof-of-concept prototype which will demonstrate some key features of the project.

5.1 Client-Side Web application:

A mock-up of the web application has been developed. Its interaction with the Firebase database will be demonstrated by implementing a simple login page as well as displaying some information to the user upon login.

5.2 Mobile Application:

A mock mobile application has been developed using Android Studio. The application currently includes some functionality. This mock app shows an example of the design which is hoped to be achieved for the mobile application as well as the flow of pages and how the functionality of the application will work. A login system and registration system which interacts with the Firebase database has been implemented.

5.3 NFC Card Interaction:

NFC card reading functionality has been implemented into the mobile application to demonstrate the use of the NFC card in authentication. The mobile application requests the user to scan their NFC card to the phone. Upon scanning the NFC card, the mobile apps compare the data on the card to data in the database. If the card ID is registered to the user, authentication is successful and some of the user's details in the database are displayed on screen to the mobile application. The user may then request a TOTP which will be needed to access the web application once the system is functional. This TOTP is currently being generated on the client side but will be implemented into server-side logic when the system is complete.

5.3 Generated One Time Passcode:

A simple program has been developed and implemented to generate a six-digit one-time passcode which is displayed to the user on their mobile application. When the system is completed, this will be implemented on the server-side of the system and as stated the user will need to provide the passcode within a given time frame to access the web application.

6. Testing

There are multiple parts of the system that need to be tested.

6.1 User Experience:

The user experience of the application must be fluid and not hinder the user as studies reviewed in the research section show that users will adopt bad security practices if a system is too difficult to access or has too many steps.
A testing environment will be setup where users may test the application and provide feedback based on their experience.

The use of this feedback will aim to improve the overall user experience of the system so that users don't make a security trade off in favour of simplicity.

6.2 Encryption Method:

The chosen encryption method implemented will be tested to ensure that it is working correctly by running the encrypted data through decrypting and brute force applications such Wfuzz and John the Ripper.

6.3 Secure coding practices:

Using the OWASP Top 10 along with source code analysing tools such as Graudit for PHP and PMD for Java code, the project will be tested to see whether the code has any vulnerabilities and if so, a more secure method will be implemented.

6.4 System Technology:

The application will be run to see if there are any errors or if the system cannot execute correctly. By reviewing error logs, the code will be debugged, and it will be ensured that all technologies are interacting with each other correctly and effectively.

7. Issues and Risks

7.1 Implementation of Authentication mechanisms:

As the domain of the project focuses on authentication, the implementation of various authentication mechanisms on the server-side of the application is essential to demonstrate a secure system. Deep research into the area of authentication will continue to be performed as well as server-side logic in order to ensure that the authentication mechanisms proposed are implemented.

7.2 Implementation of NFC cards:

The successful implementing of NFC cards into the project was an issue that was addressed from the start as it is a key feature of the system. Compatibility issues among others were all considered at the design phase of the project. This issue has been addressed and a prototype to demonstrate the NFC card authentication function of the application has been developed.

7.3 Time Constraints:

The system requires a lot of further research and development and as the project is due for submission around Easter 2019 there is an evident time constraint. The project plan outlined below will be adhered to in order to ensure that all aspects of the proposed system is completed.

7.4 Overlooked or Unknown Security Vulnerabilities in system:

As the domain of the project is security, it will be ensured there are no security flaws in the system, such as insecure server and database logic and bad coding practices which may be exploited. By analysing the OWASP Top 10 which provides a rich source of information about security risks in code and web-based applications, the system will be secured.

7.5 Security Vs Usability:

As discussed in the research section, one of the main reason's organisations don't adopt multifactor authentication solutions is because of the burden they have on the user experience. The system will provide a fluid user experience without a security trade-off. This will be achieved by researching popular existing authentication solutions as well as providing a simple user-friendly application. As stated in the *Testing* section testing on the user experience of the application will be done and the application will be improved with feedback received.

8. Plan and Future Work

8.1 Key Deliverables

Deliverable:	Date:
Proposal Submission	04th October 2018
Interim Submission	04th December 2018
Interim Project Presentation	6th December 2018
Project Dissertation Submission	Easter 2019
Software Submission	Easter 2019
Project Demonstration	Easter 2019

8.2 Project Plan

The GANTT chart below outlines the Project Plan for the duration of the project lifecycle.

8.2.1 GANTT Chart:

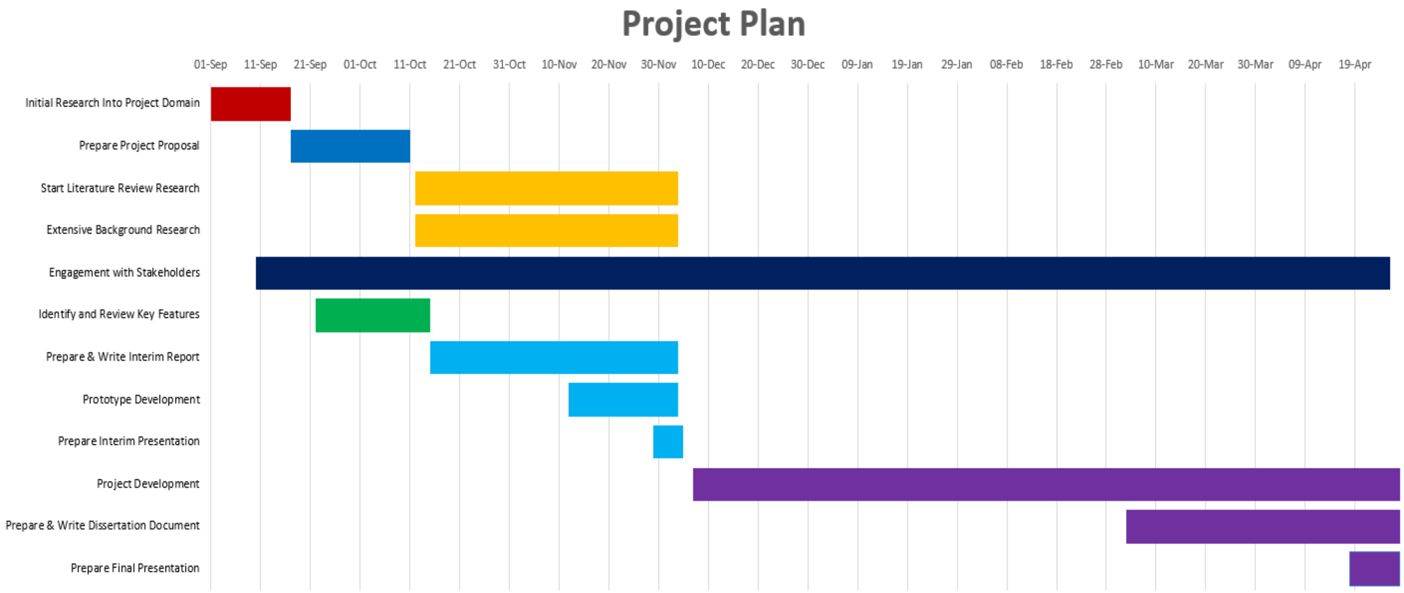


Figure 12 - GANTT Chart Project Plan

8.2.2 Engagement with Project Stakeholders:

Along with following the plan detailed in the GANTT chart, stakeholder engagement will continue throughout the lifecycle of the project. Notes will be taken of criticism and suggestions and then implemented into the project if needed. (see appendix A and B).

9. Bibliography

Abadi, M., Burrows, M., Kaufman, C., & Lampson, B. (1993). Authentication and delegation with smart-cards, 21.

About - Zend Framework. (n.d.). Retrieved from https://framework.zend.com/about

Afroz, S., Islam, A. C., Santell, J., Chapin, A., & Greenstadt, R. (2013). How Privacy Flaws Affect Consumer Perception. In *2013 Third Workshop on Socio-Technical Aspects in Security and Trust* (pp. 10–17). New Orleans, LA, USA: IEEE. https://doi.org/10.1109/STAST.2013.13

Andrew Powell-Morse. (2016, December 8). Waterfall Model: What Is It and When Should You Use It? Retrieved October 29, 2018, from https://airbrake.io/blog/sdlc/waterfall-model

Android Vs. iOS. (2018, March 7). Retrieved October 29, 2018, from https://blog.sagipl.com/android-vs-ios-development/

Bonneau, J., Herley, C., van Oorschot, P. C., & Stajano, F. (2015). Passwords and the evolution of imperfect authentication. *Communications of the ACM*, *58*(7), 78–87. https://doi.org/10.1145/2699390

Collection of 1.4 Billion Plain-Text Leaked Passwords Found Circulating Online. (n.d.). Retrieved October 22, 2018, from https://thehackernews.com/2017/12/data-breach-password-list.html

contributors, M. O., Jacob Thornton, and Bootstrap. (n.d.). Bootstrap. Retrieved November 15, 2018, from https://getbootstrap.com/

Currents, D. N. D. N. has been a tech journalist since 1994 H. work has appeared in C., Examiner, T., Spruce, T., & Publications, O. (n.d.). What Is iOS?

Retrieved October 29, 2018, from https://www.lifewire.com/what-is-ios-1994355

Darwish, M., & Ouda, A. (2015). Evaluation of an OAuth 2.0 protocol implementation for web server applications. In *2015 International Conference and Workshop on Computing and Communication (IEMCON)* (pp. 1–4). https://doi.org/10.1109/IEMCON.2015.7344461

Esplin, C. (2016, October 24). What is Firebase? Retrieved October 20, 2018, from https://howtofirebase.com/what-is-firebase-fcb8614ba442

Feature Driven Development (FDD) and Agile Modeling. (n.d.). Retrieved October 22, 2018, from http://agilemodeling.com/essays/fdd.htm

General Python FAQ — Python 3.7.1 documentation. (n.d.). Retrieved October 29, 2018, from https://docs.python.org/3/faq/general.html#what-is-python

IDC - Smartphone Market Share - OS. (n.d.). Retrieved October 29, 2018, from https://www.idc.com/promo/smartphone-market-share

Jajoo, A. (2018). A study on the Morris Worm, 18.

Kaur, N., & Devgan, M. (2015). A Comparative Analysis of Various Multistep Login Authentication Mechanisms. *International Journal of Computer Applications*, *127*(9), 20–26. https://doi.org/10.5120/ijca2015906472

Li, Y., Wang, H., & Sun, K. (2017). Personal Information in Passwords and Its Security Implications. *IEEE Transactions on Information Forensics and Security*, *12*(10), 2320–2333. https://doi.org/10.1109/TIFS.2017.2705627

Lindemann, R. (2013). The Evolution of Authentication. In H. Reimer, N. Pohlmann, & W. Schneider (Eds.), *ISSE 2013 Securing Electronic Business Processes* (pp. 11–19). Wiesbaden: Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-03371-2_2

Mohammed, S., Ramkumar, D. L., & Rajasekar, V. R. (2017). Password-based Authentication in Computer Security: Why is it still there?, *5*(2), 6.

.NET Programming Languages. (n.d.). Retrieved October 29, 2018, from https://www.microsoft.com/net/languages

OAuth 2.0: The Complete Guide. (n.d.). Retrieved November 8, 2018, from https://auth0.com/blog/oauth2-the-complete-guide/

PHP: What can PHP do? - Manual. (n.d.). Retrieved October 29, 2018, from http://php.net/manual/en/intro-whatcando.php

PostgreSQL: About. (n.d.). Retrieved October 29, 2018, from https://www.postgresql.org/about/

Schmidt, C. (2016, June 24). What is Android? Here is a complete guide for beginners | AndroidPIT. Retrieved October 18, 2018, from https://www.androidpit.com/what-is-android

Seo, J., Kim, K., Park, M., Park, M., & Lee, K. (2017). An analysis of economic impact on IoT under GDPR. In *2017 International Conference on Information and Communication Technology Convergence (ICTC)* (pp. 879–881). https://doi.org/10.1109/ICTC.2017.8190804

Solution, C. T. (2018, March 13). Django vs Laravel: Which Framework Should You Choose? Retrieved November 11, 2018, from https://hackernoon.com/django-vs-laravel-which-framework-should-you-choose-1d0f40049ec1

Symfony at a Glance. (n.d.). Retrieved November 15, 2018, from https://symfony.com/at-a-glance

tutorialspoint.com. (n.d.). EmberJS Overview. Retrieved October 29, 2018, from https://www.tutorialspoint.com/emberjs/emberjs_overview.htm

What is AngularJS. (n.d.). Retrieved October 20, 2018, from http://www.tutorialsteacher.com/angularjs/what-is-angularjs

What Is Java, and Why Is It So Great? (n.d.). Retrieved October 29, 2018, from https://www.dummies.com/programming/java/what-is-java-and-why-is-it-so-great/

What is MariaDB? - Definition from WhatIs.com. (n.d.). Retrieved October 29, 2018, from https://searchdatamanagement.techtarget.com/definition/MariaDB

What is Oracle Database (Oracle DB)? - Definition from Techopedia. (n.d.). Retrieved October 29, 2018, from https://www.techopedia.com/definition/8711/oracle-database

What Is the Extreme Programming Methodology? | Lucidchart Blog. (n.d.). Retrieved October 29, 2018, from /blog/what-is-extreme-programming

10. Appendix

Appendix A:

Stakeholder A Meeting - 1

Stakeholder A: Cyber Security Expert

Date and Time: 10th September 2018

Meeting Purpose:

I met with a Cyber Security Expert to discuss the idea for my project and the domain surrounding my project. I wanted to get an expert's opinion on the strength and weaknesses of my proposed project.
As this stakeholder has over 15 years' experience in the area of cyber security, he was able to give me recommendations as well as criticism on my project.

Items Discussed:

- My proposed project
- Existing Multifactor Authentication solutions
- Risk of text-based passwords
- Use of smart-cards in authentication

Recommendations:

- Implement a Time-Based One-Time Password system
- Implement strong encryption methods
- Add complexity to the project by adding additional system functionality, research common problems on IT systems within organisations

Criticism:

- Smart-cards may be stolen, not very secure alone
- System generated passwords can sometimes be more secure than user passwords and have risks associated with them such as users casually leaving them as written notes on a colleague's monitor or desk.

Personal Conclusion:

Because I want to use smarts-cards as an authentication step I will need to implement other authentication steps to eliminate the risk of stolen smart-cards.

I also plan to implement an access control system to add complexity to my project as well as security to the system I am designing.

Stakeholder A - Meeting 2

Stakeholder A: Cyber Security Expert

Date and Time: 08th October 2018

Meeting Purpose:

I met with the same Cyber Security Expert again to further discuss my project. Going into this meeting I had an idea of what I wanted to accomplish with my project and I wanted to present all my research and findings to an expert in the field.

Items Discussed:

- Project Design
- Background Research
- Resultant Findings
- Access Control
- Different encryption methods
- Proposed letting the user pick their own authentication methods

Recommendations:

- Implement an admin system for controlling user access
- Further design and develop an access control system to add complexity to project
- Research OWASP principles for securing application security

Criticism:

- Be careful not to let the user have too much control over their own security on the system

Personal Conclusion:

I have a better idea of what type of system I want to develop and its requirements. I thought about implementing multiple authentication methods and letting the user choose their preferred method. However, after recent discussions I have discovered that the system must handle most if not all the security, as letting a user choose their preferred authentication methods could compromise the system.

I will create a separate system on the web application to allow administrative controls for the access control system I want to implement.

Appendix B:

Stakeholder B - Meeting 1

Stakeholder A: Director of International Fashion Distribution Company

Date and Time: 12th October 2018

Meeting Purpose:

I met with the Director of an international company that deals with sensitive data such as personal information, credit card information, and customer lists daily. I wanted to gain an insight into the difficulty's that companies have securing the privacy of such data.

Items Discussed:

- Security Mechanisms in use
- Introduction of GDPR
- Financial and Reputational loss that can occur after privacy leaks
- Implementing Access Control for company systems

Recommendations:

- Ensure all data is stored in an encrypted format
- Only allow users access to information relating to them
- Ensure authentication is as user friendly as possible without compromising security

Personal Conclusion:

This meeting has underlined to me the importance of systems such as the one I am aiming to develop for implementation in organisations.

Privacy leaks can be catastrophic for an organisation, from both a financial and reputational standpoint. To mitigate privacy leaks, simple safeguards such as two factor authentication, access control and encryption should be implemented into IT systems.

I will implement all of these safeguards into my project and create a strong secure system.