

Web Based Property Management Tool for Landlords & Tenants

Luke Alexander Dawes

15810469

This report is submitted in partial fulfilment of the requirement for the degree of BSc Computer Science

2017/2018

# Abstract

Eighteen years into the 21st Century, technology has rapidly improved the way we communicate and connect with each other. With over 10 million people privately renting in the UK today, landlords should embrace the power of technology to assist with the management of their properties and communication between their tenants (Author Online , 2018).

Although there are a variety of software available for landlords to manage their properties, the market lacks solution available for both landlords and tenants to use effectively. Many of these software are not available on all operating systems nor provide the full service needed to effectively perform all tasks carried out by landlords and tenants.

To tackle this issue in my project, I plan to develop a user-friendly web-based application that is open source, free and available on devices with a web browser and internet connection.

# Acknowledgements

After months of hard work, I have left this moment till last so I can truly thanks and acknowledge those who have helped me throughout the project.

Firstly, I would like to thanks my family and girlfriend for their constant support they have shown throughout the project.

Secondly, I would like to thank a Karen, a friend who currently work in the IT industry and has helped me to develop this project with a professional mind.

Thirdly, I would like to acknowledge my college teach Esam, who was my first teacher of web application development and provided me with the skills necessary to study it further.

Lastly, I would like to thank all that has participated in this project or helped in any way.

Contents

[Abstract 2](#_Toc514706764)

[Acknowledgements 3](#_Toc514706765)

[Chapter 1 – Introduction 7](#_Toc514706766)

[1.1. Problem Statement 7](#_Toc514706767)

[1.2. Aims & Objectives 7](#_Toc514706768)

[1.3. Minimum Requirements 8](#_Toc514706769)

[1.4. Testing 8](#_Toc514706770)

[1.5. Relevance to Degree 9](#_Toc514706771)

[Chapter 2 – Background Research 10](#_Toc514706772)

[2.1. Duties of a Landlord 10](#_Toc514706773)

[2.2. Duties of a Tenant 10](#_Toc514706774)

[2.3. Technologies Considered 11](#_Toc514706775)

[2.3.1. Base Programming Languages 11](#_Toc514706776)

[2.3.2. Structured Query Language (SQL) Database & PHP 12](#_Toc514706777)

[2.3.3. Firebase 13](#_Toc514706778)

[2.3.3.1. Realtime Database 13](#_Toc514706779)

[2.3.3.2. Authentication 13](#_Toc514706780)

[2.4. Existing Solutions 13](#_Toc514706781)

[2.4.1. Web-based Application Solutions 13](#_Toc514706782)

[2.4.2. Standalone Solutions 14](#_Toc514706783)

[Chapter 3 – Project Management 15](#_Toc514706784)

[3.1. Methodology 15](#_Toc514706787)

[3.1.1. Waterfall 15](#_Toc514706788)

[3.1.2. Agile 15](#_Toc514706789)

[3.2. Requirements Analysis 16](#_Toc514706790)

[3.3. Project Timetable 18](#_Toc514706791)

[3.3.1. Gantt Chart 18](#_Toc514706792)

[Chapter 4 – Development 19](#_Toc514706793)

[4.1. Wireframes & Prototype 19](#_Toc514706794)

[4.2. Implementation 20](#_Toc514706795)

[4.2.1. Index – Sign up Page 21](#_Toc514706796)

[4.2.2. Login Page 22](#_Toc514706797)

[4.2.3. Structuring & Implementing Firebase Real Time Database 23](#_Toc514706798)

[4.2.4. Home Page 24](#_Toc514706799)

[4.2.5. Menu Item Pages 27](#_Toc514706800)

[4.2.5.1. Profile Page 27](#_Toc514706801)

[4.2.5.2. Properties Page 28](#_Toc514706802)

[4.2.5.3. Transmen Page 29](#_Toc514706803)

[4.2.5.4. Maintenance Issues Page 29](#_Toc514706804)

[4.2.5.5. Tenants Page 30](#_Toc514706805)

[4.2.5.6. Documents 31](#_Toc514706806)

[4.2.5.7. Expenses 31](#_Toc514706807)

[4.2.5.8. Reminders 31](#_Toc514706808)

[4.2.5.9. Settings 32](#_Toc514706809)

[5. Chapter 5 – Testing 33](#_Toc514706810)

[5.1. User Testing 33](#_Toc514706811)

[6. Chapter 6 – Conclusion 35](#_Toc514706812)

[6.1. Project Conclusion 35](#_Toc514706813)

[6.2. Assessment of Succession of Project 35](#_Toc514706814)

[6.3. Personal Reflection 36](#_Toc514706815)

[Bibliography 37](#_Toc514706816)

[Appendix One 39](#_Toc514706817)

[A. Original Aims & Objectives 39](#_Toc514706818)

[B. Landlord Responsibilities 39](#_Toc514706819)

[C. Interview Transcript for Tenant Interview 39](#_Toc514706820)

[D. Transcript for Landlord Interview 41](#_Toc514706821)

[E. Functionality of Web-Application Including Changes 43](#_Toc514706822)

[Home page – Login 43](#_Toc514706823)

[Signing up as a Landlord 43](#_Toc514706824)

[Signing up as Tenant 45](#_Toc514706825)

[Personal Landlord User Home Page 46](#_Toc514706826)

[Personal Tenant Home Page 47](#_Toc514706827)

[Profile Page 48](#_Toc514706828)

[(Landlord) – Property Page 48](#_Toc514706829)

[(Landlord) – Tradesmen Page 49](#_Toc514706830)

[Reminders 49](#_Toc514706831)

[Maintenance issues 49](#_Toc514706832)

[Tenants Page 50](#_Toc514706833)

[Documents 50](#_Toc514706834)

[Expenses 51](#_Toc514706835)

[F. Standalone Comparison Table 51](#_Toc514706836)

[G. Requirements Table 51](#_Toc514706837)

[H. Original Gantt Chart 54](#_Toc514706838)

[I. Full Firebase Database Structure 55](#_Toc514706839)

# Chapter 1 – Introduction

In this chapter I will further discuss the problem I’m seeking to tackle with my project, outline the aims and objectives of the project, the requirements of the final deliverable, testing of the web application and finally the relevance my project has to my degree.

## Problem Statement

With soaring house prices in the UK, there has been an increase in demand for rental properties due to people not being able to own their own home. This opens a gap in the market for property owners to rent their home to people seeking a place to rent. With more people renting their homes and renting properties, the time-consuming problem of property management and overall communication between both parties becomes a bigger issue. There is an abundance of standalone and online applications for landlords to manage their properties but can become expensive when forced to buy multiple applications to perform different tasks. However, current solutions exist on the internet today, which means availability is not a big issue, they are high in price and require costly updates.

The proposed solution in this project is to create a web-based application. It will be written using the programming and mark-up languages; HTML5, JavaScript, CSS3 and external libraries such as jQuery to make development more efficient. The application is small in size which benefits users with limited hard-drive space, open source, and free, making it accessible for anyone to use.

## Aims & Objectives

Since proposing this project, some aims and objectives have changed to better meet the final deliverable of this project (*See* ***Appendix One*** *for original aim and objective plan*). In a technical sense, the aim of this project is to produce a web-based application that will be used by landlords and tenants.

The main aim of this project is to provide one environment that can be used by both landlords and tenants without the need to use external applications or software to fulfil their duties. The application should also provide an instant messaging feature to allow both parties to communicate instantly and directly. Although the aim of this project is to prevent the need of using multiple software, I understand that some users may already be using other software. With that in mind, I plan to design the application to work and function in a similar way as existing softwares so that its behaviour is predictable and familiar to users.

To meet theses aims, I have set the following objectives:

* Revise existing research around the topic of investigation.
* Collect information from stakeholders and understand their interest and requirements.
* Develop a clear understanding of the relationship between landlords and tenants and how they currently communicate.
* Research programming languages that will help to assist form my proposed solution.
* Information from stakeholders will be used to construct the functions and features of the website.
* Create a website where landlords and tenants can communicate and fulfil their individual duties.

## Minimum Requirements

To ensure that I provide the best possible solution to my proposed problem, I have created minimum requirements that I need to meet in the timeframe of the GRANTT chart before submitting the software deliverable. They are as follows:

* An interface that’s fully functional, visually appealing, and simple to understand.
* Ability to switch from one property to another.
* Account management system; including sign up and login functionality.
* Create and edit maintenance issues, used to inform users of maintenance issues within a property.
* Create reminders, used to inform users of notable events or tasks.
* A visual notification board which allows users to keep up to date with important notifications.
* Instant messaging function to allow users to communication with each other.
* Expense management that provides users with the ability to manage the expenses of the property.
* A place to view registered tenants and properties, and a profile page for each tenant.
* Ability to control the settings of contact information, notifications and account.
* A way to log out.

## Testing

Before submitting the final software deliverable, a couple of stakeholders were selected to perform task on the software to ensure it works in the way they expect it to. This method of testing is known as user testing and was chosen because the performance and interaction of the user can be analysed and direct feedback from the users can be received. The information gathered from the user testing can be used to source out what works well within the system and what doesn’t and apply the finding to make improvements.

## Relevance to Degree

The reasoning behind the choice of project is because it’s based around what I learned most during my time at University of Brighton. The modules studied that have contributed to the project include:

* **CI102 – Introduction to Database:** which educated me on how web-based application and databases are connected and used in today’s society. This module also helped to understand the structure of databases and how they store information.
* **CI135 – Introduction to Web Development:** in this module I recapped many subtopics in web development that I learnt prior to university, however, it also broadened my knowledge in JavaScript and showed a better practice when developing website.
* **CI141 – Human Computer Interaction:** this module showed many ways in which human interact with computers and technology. Not only that but the science and reasoning behind interface designs.
* **CI143 Introduction to Requirements Analysis:** because it opened a door to project management I didn’t know existed. This module helped me to understand the process and different methods of gather requirements to achieve a successful project.
* **CI222 – Project Planning and Control:** was an interesting module and taught me how to correctly plan and project in the most effect way.
* **CI347 – Web Network and Management:** Although I studied this module as I complete this project, it helped me along the way by pushing me to do research about tool that are useful to a ‘webmaster’ which I then later used in my project.

All the above modules have contributed to this project in some way however, this project has also helped me to widen and improved on the skill these modules have provided me with.

# Chapter 2 – Background Research

In this chapter, I will discuss the background research I perform in the early stages of this project, prior to the development stage. In addition, an introduction to the programming languages, Application Program Interface (API), and Software Development Kits (SDK) that was used or taken into consideration within the project. Finally, a quick look into current solutions available and the architecture behind them.

## Duties of a Landlord

To have a clear understanding of what potential users will expect from the web application, I researched into to roll of a landlord and the duties they carry. I gathered a selected couple of participants to take part in a one-to-one interview to have an insight in their role as a landlord, which I discuss this further in **Chapter Three**.

Additionally, I took it upon myself to investigate the role of a landlord using online resources and books provided in the university library. A property that’s fit to live in and the safety of tenants was something I came across often during my research. A list of the landlord’s responsibilities to make repairs to the property is listen in **Appendix One**. This information was used to structure the layout of the Maintenance Issues page.

If a tenant has contacted the landlord requesting repairs on the property, the landlord is expected to inform the tenant when the repairs will be done.

Evaluating the overall conditions of the property is another important duty a landlord has. Not only does it allow the landlord to view any problems with the property, but it gives them a chance to confirm if the tenant is keeping to their end of the agreement or not (Property Investment Project, 2013). Legally, the landlord must contact the tenant to inform they’ll be visiting the property with a minimum of 24 hours’ notice.

Cost and expenditure management can be a stressful responsibility for landlords. Tax payment, maintenance cost, insurance and rental management can be time-consuming and needs to be constantly updated. This is a landlord’s duty outside of any tenancy agreement however is still a significant role of a landlord (Rollingsons, n.d.).

## Duties of a Tenant

Tenants may not have the same amount of duties and responsibilities as a landlord, but they still legally have responsibilities to keep. One of a tenant’s main responsibilities is to pay rent on time. Rent is usually paid in advance every month and should be paid on time. If rent becomes backlogged, the landlord has the right to add additional charges or even take steps in evicting the tenant and claiming the money back (Shelter, 2017).

Although the landlord has the responsibility of repair and maintenance, it’s the tenants job to take care of the property and undertake minor maintenance jobs such as checking smoke alarms. If any damages are made to the property’s interior or exterior, the tenant will be liable for the cost of repairing the damages. If any maintenance issues occur, it’s the tenants job to immediately report it to the landlord so they can be dealt with promptly. However, no matter how serious the issue is, the tenant is still liable to continue to pay rent on time (Shelter, 2017).

Allowing access for repairs is another responsibility of a tenant. In the tenancy agreement, the law implies that the tenant must give access to the property for the landlord. The landlord has a right to see what repairs may need doing in the property, but unless it’s an emergency, the landlord must provide at least 24 hours’ notice (Citizens Advice, 2017).

My independent research into the duties of landlords and tenants has helped me to understand the legal and non-legal responsibilities of both parties. Although some of my findings didn’t include a variety of responsibilities that could be seen a task within an application, they can still be used to structure the proposed application and confirms that the biggest responsibility for either party is good communication.

## Technologies Considered

Before developing the web application, I wanted to have a clear understanding of what technologies will be used to build and structure the application. I looked into a variety of technologies that could potentially be use in the project, some of which I went on to use and some that I didn’t.

### Base Programming Languages

It was clear from the initial stages of the project that the web-application will be built using a web-based programming language, therefore HTML was one of the first languages to be considered. HTML5 is the latest specification of HTML which is the one I chose to use for this project because of its cross-browser support, mobile support, and ability to write clean and descriptive code for good practice.

Although HTML builds the foundations of a web-application, it’s limited to the way it can change the style and appearance of the web-page. CSS was the next language considered in the development of the application. Paired with HTML, CSS has the capability of changing the colour, shape, size, position and much more of HTML elements. For the application to be compatible, I followed the CSS3 specification as it’s currently the best to work with HTML5.

With the web-application only containing HTML and CSS, it would only display static information that does nothing. To make the application more interactive, which is required for my project, I considered using JavaScript (JS) to implement the needed interactivity on the application. It was a firm decision that JS will certainly be used in the backend development of the application alongside the jQuery library. The library is a cross platform JS library used to simplify the manipulation of HTML elements and provide the appearance of neater, easier to read code.

I used an online source that is very detailed to re-familiarise myself with the syntax and revise the basics of JS and jQuery. It wasn’t long before I started to remember how to use JS and jQuery to manipulate HTML (W3 Schools, 2018). In addition, I read books from the library about HTML5 and CSS3 to update my knowledge on the languages and find anything that I didn’t know that could be useful in the development of the application[[1]](#footnote-1).

The reasoning behind my choice of a web-based application over a standalone is because I have more personal experience with developing web applications and using web technologies. Furthermore, for the type of application I plan to build, required technologies such as API’s can be implemented to a web-based application easier compared to a standalone. If, however I was to choose a standalone, I would use the programming language C++ for it suitability for personal applications.

### Structured Query Language (SQL) Database & PHP

Data storage is an important aspect of any application and deciding the correct way to store data is important to get right before starting development. SQL was considered early in the project to store and manage data for the application due to studying it in my first year of university. Since then I haven’t studied or investigated SQL databases and decided to look further into it. What I found was, to connect an SQL database to a web-application requires the knowledge of a server-scripting language such as PHP. I previously studied this language, but it was a brief overview and wouldn’t be confident enough to create an application that strongly uses the language.

After speaking with my project supervisor, he informed me that Firebase offer an online, real-time database that is free to use and suitable for my project. I took it upon myself to do more research into Firebase which is discuss in more detail below.

### Firebase

Firebase is Google’s mobile and web application platform that help developers to create high-end application. It offers a range of services and products that come with documentation to help implementation. I the research, two services were discovered that amazed me and would be extremely useful in the development of the web application.

#### Realtime Database

Firebase Realtime Database (FRDB) is an online service that provides developers with an API to store and sync data between the applications users. Not only does this service allow the users to access their data on either mobile or web applications, but it also helps the users to collaborate with one another. The FRDB is optimised for offline use, meaning whenever a user loses their connection, the database SDK uses a local cash on the device to store changes. Therefore, when the user’s connection is restored, the changes made in the database while they were offline is automatically synchronised. FRDB can be seen as the complete opposite to an SQL database because it’s a non-relational database and data is stored as JSON (Firebase, 2018).

From my research, I decided that the FRDB was the method I will used to manage and store the data of the application because it simple to implement and is in real time.

#### Authentication

In the early stages of my project I knew the application need to have a way of identifying users to provide them with a unique experience on the application. I my research I came across many SDK’s and API that would help imbed authentication into my application, however after researching into FRDB, I discovered a service Firebase offered called ‘Authentication’.

Firebase Authentication (FA) provides SDK’s and open-source UI’s to authenticate users to an application. FA also supports authentication through familiar services such as Facebook, Twitter, GitHub and Google. When a user signs up, they’re provided with a unique identification key that will never change. This key is used to access all the user’s information and control what data they can access on the backend server (Firebase, 2018). Due to FA’s simplicity and connection to the FRDB, my consideration quickly turned into a decision to use FA as a way of authenticating users on to my application.

## Existing Solutions

### Web-based Application Solutions

Researching existing web-based application was an important part of my project because of their similarity to my proposed solution. To begin my research into this, I discovered an article by Zoe Hall how listed *“15 essential websites for landlords”* (Hall, 2017) which included web-based software related to my project.

RentCentral was the first software to be listed and provides a variety of service for landlords to manipulate and use to their advantage. It has a wide range of features such as managing tenants and expenses, preparation for self-assessment, and tracking of paperwork, rent and deposits. RentCentral offer different price brackets for the services they provide which range from free to £18.33 per month. Depending on what package the user buys, will determine what they are entitled to. Overall the website has a clean layout and style but what it lacked was the ability for tenants to interact with the system as well has the landlord.

### Standalone Solutions

Although my proposed solution is not a standalone software, I decided to research into standalone solutions to seek any insight that might help with the development of my application. In my research I discovered an ongoing survey on SoftwareAdvice where users can rate and review the software they use to fulfil their duties as a landlord. I picked the top three software to explore in more detail and displayed their pricing, space, operating system support and more in a table found in **Appendix One**.

# Chapter 3 – Project Management

In this chapter I will discuss the stages of the project management phases used to achieve a professional, high-level web-based application. This will include, the methodologies considered, requirements gathering and analysis, and project scheduling.

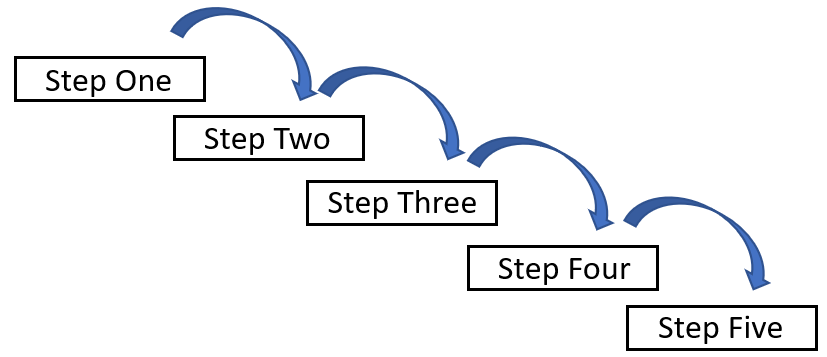


## Methodology

Using the correct methodology is important in project management and I intended to have a clear idea of the type of methodology I will use in the early stages of the project. The ‘CI222 – Project Planning and Control’ module had a big contribution to the decision of which methodology to choose, however I did some research online to seek better understanding of different methodologies to make a better decision.

### Waterfall

The waterfall methodology is considered the classic and tradition framework that has unique goals set for each phase of the development which needs to be completed before the next phase has started. Once the next phase has started there’s no turning back therefore, in theory, the project will be delivered on time due to the well planned and detailed phases.

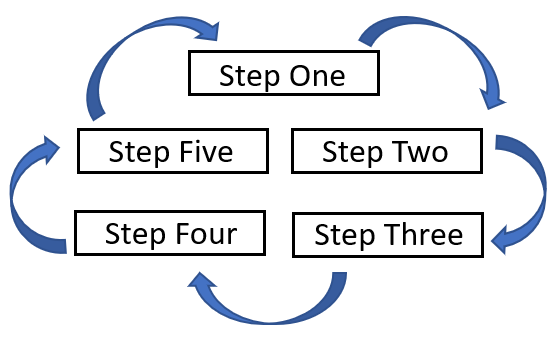


**Figure 1: Waterfall methodology.**

This method can’t accommodate sudden changes software development may have. With waterfall, it’s difficult to go back and make changes once the stage has been completed. I understood that this was not the right methodology for me to follow because at any time in this project, something can change such as a requirement or design feature. For these reasons, I didn’t use the waterfall methodology.

### Agile

Agile has a completely different approach to project management compared to waterfall. It’s designed to minimize risk of failure by each stage of the project broken down into small increments that provides the opportunity to go back and correct or make changes, even if the stage may have been completed.



**Figure 2: Agile methodology.**

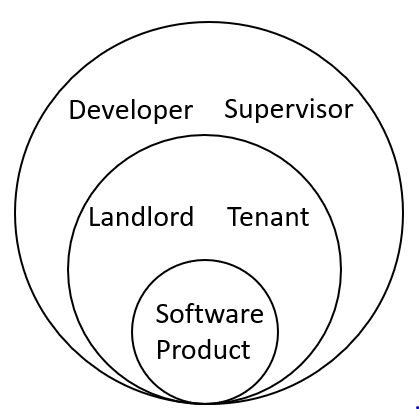
It also encourages frequent inspection and adaption to ensure that the project is moving in the direction it’s supposed to. Piece by piece, the small increments of the project come together to form the final software deliverable with minimal risk of failure.

From what I have learned about agile in my previous year at university and my findings from this research, the decision to use an agile approach to the software development project was certain because it accommodate changes throughout the project and phases that cross paths and executed simultaneously.

## Requirements Analysis

Gathering and analysing requirements is one, if not the most important phase of pre-software development. Ken Harkin, a representative of Sparx Systems explains why defining requirements in the early stages of a project is important. He goes on to say in an article on The New Economy that requirement gathering minimises risk of failure by define the purpose, expectation and user needs of the software before starting development (Harkin, 2013).

Stakeholders are the prime people to seek out when gathering requirements because they will either have an interest in the software product or going to be affected by it. Key stakeholders are usually the users; therefore, landlords and tenants are the people who meet the criteria of a suitable stakeholder to gather requirements from. To visually understand the stakeholder’s relationship to the project goal I created a stakeholder onion model containing all stakeholder’s.



**Figure 3: Stakeholder Onion Model**

Collecting requirements can be done in several ways such as on-to-one interviews, workshops and first-hand experience. I gathered a couple of tenants and landlords I initially planned to conduct a workshop with, however due to their differences in availability, it was difficult to gather both on the same day to conduct the workshop. Interviewing the stakeholders provides a first-hand insight to what they expect from the software product. It also allows the developer to understand the users and how they will interact with the software. For this reason, I decided to use interviews as a fall back from a workshop despite the possible risks using interviews hold such as; too much information to manage, time consumption, and only every getting one perspective.

Before I started planning for the interviews, I done some online research on different techniques to gather requirements through interviews. In an article on User Focus, Dr Philip Dodgson explains, before asking questions it’s important not to guess what the stakeholder wants or expects from the software product, and to get all the issues out. He also goes on to say that hesitation and uncertainty will produce weak requirements so avoid moving on if the stakeholder is uncertain (Hodgson, 2014).

I started the interviews with a set template of question but tended to sway from the template and come back to it. I had an associate record everything that was said during the interview, so the stakeholder’s responses could be analysed and used to generate requirements. The transcript of the interviews can be found in **Appendix One**.

Once both interviews had been executed, I read through their transcript and created the requirements from the answers the stakeholders provided which can be found in **Appendix One**.

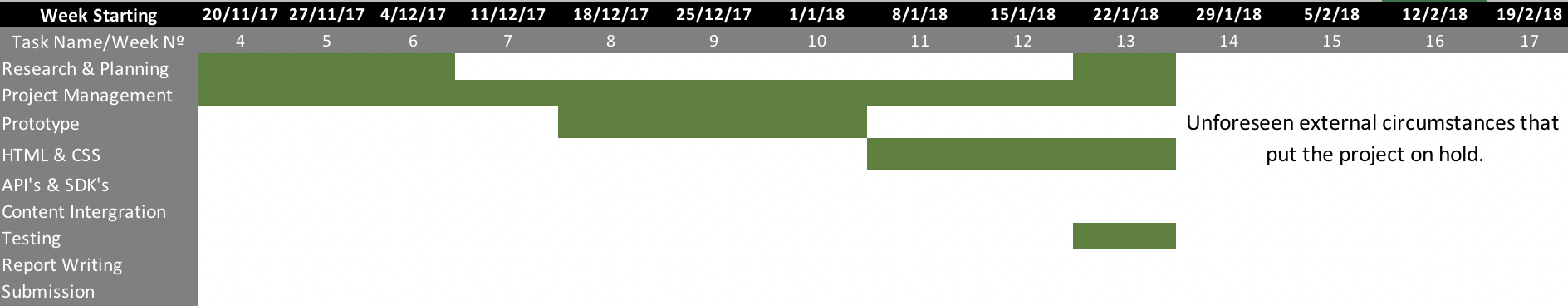
To further my understanding of the functionality of the application, I used the requirements gathered from the stakeholders and created a detailed discerption of tasks and actions users can perform on the application. I then emailed this document to the stakeholders, so they can read through it and make any changes they deem necessary. When I received their responses, I combined changes made by both parties and created a final draft which can been seen in the requirements table in **Appendix One**.

## Project Timetable

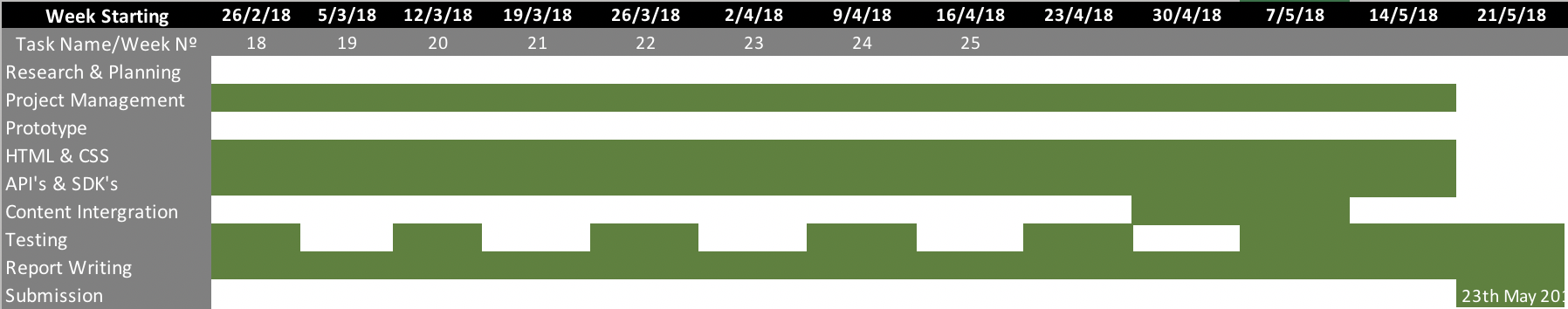
Planning and scheduling must be done correctly to maximise the probability of a successful project. There’re several ways to plan and schedule the phases and activities within a project to ensure that they are completed on time. If deadlines are missed and or a task is finished out of schedule, it can have knock-on effects on the rest of the project.

### Gantt Chart

A Gantt was used to display the stags of the project, so I could quickly glance at the chart and know exact timings of each stage. In the Gantt chart, the stages are detailed on the left with the week numbering and dates along the top. Cells that are filled with green on the chart correspond to what stage I should be working on and when. The Gantt chart seen in figure 4 and 5 is the most recent chart which has been modified to make accurate. The original version can be found in **Appendix One**.

****

**Figure 4: Current Gantt chart Pt.1**

****

**Figure 5: Current Gant chart Pt.2**

# Chapter 4 – Development

Details of the development stage of the project will be discussed in this chapter. The key stages of the development will be split into subchapters and discussed further.

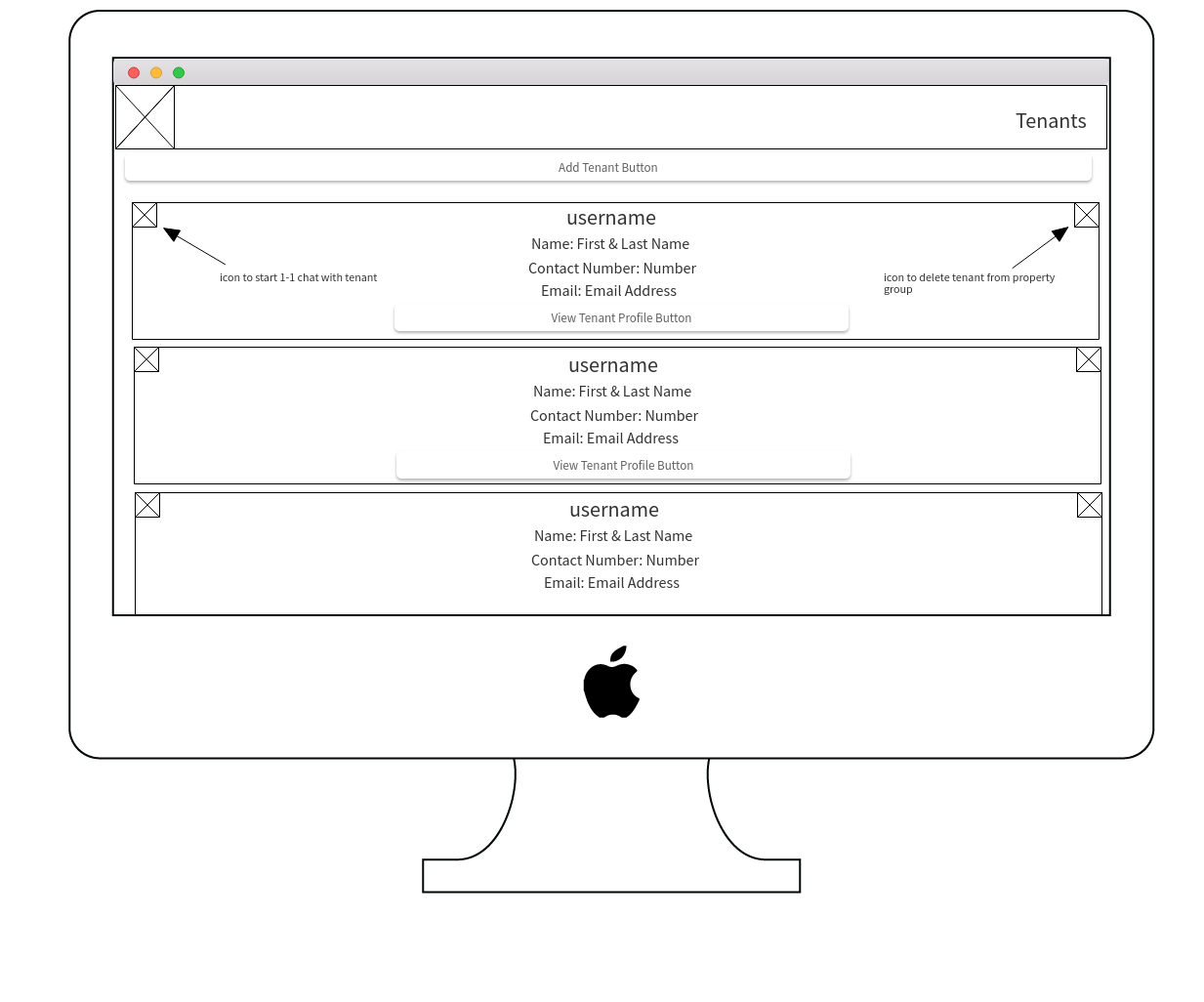
## Wireframes & Prototype

Designing the wireframes and creating a prototype is important because it provides the stakeholders with a visual representation of what the software product will look like. It also presents the opportunity for any changes stakeholders feel is needed within the design prior to development.



Figure 6: Home page wireframe.

The first wireframe I created was for the home page. I based the design from some of the requirements gathered from the interviews in the requirement analysis stage. The home page is required to have a way for landlord users to switch from one property to another. Therefore, I design a button to perform this function because it’s user friendly and provides a clear design. The home page also required to have a notification box, instant messenger for the property group, and a rental payment box (*see Figure 6*).



**Figure 7: Tenant page wireframes.**

The second wireframe I created was the page where the registered tenants of a property are listed. On this page the user will be able to view each tenants’ name and contact details with a button to view their full profile. In each corner will be a clickable icon to perform different task. If the user is a landlord, one icon will be to remove tenants from the property group and the other is to start a one-to-one messenger with a tenant available to all users (*see Figure 7*).

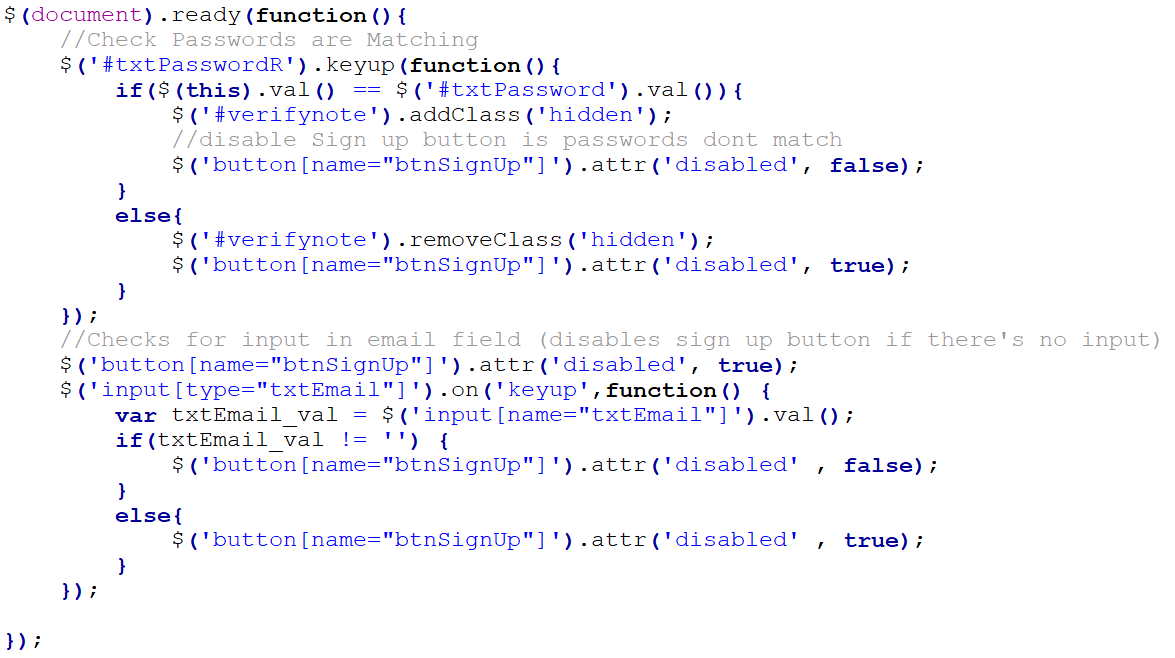
Once I completed the wireframes, my initial plan was to gather the close group of stakeholders I selected and meet up to go through the wireframes. However, due to different availability, I was forced to email each member of the close shareholders group a copy of both wireframe for them to analyse and report back their opinions and any changes they deem necessary. Both stakeholders got back to me and confirmed they’re happy with the designs and they meet all their expectations. With that being said, I then began to design and implement the website based from the wireframe designs.

## Implementation

With the importance of using the website anywhere anytime, I developed each web page with the mobile first responsive design approach. This was to ensure user have the best experience no matter what device they use.

### Index – Sign up Page

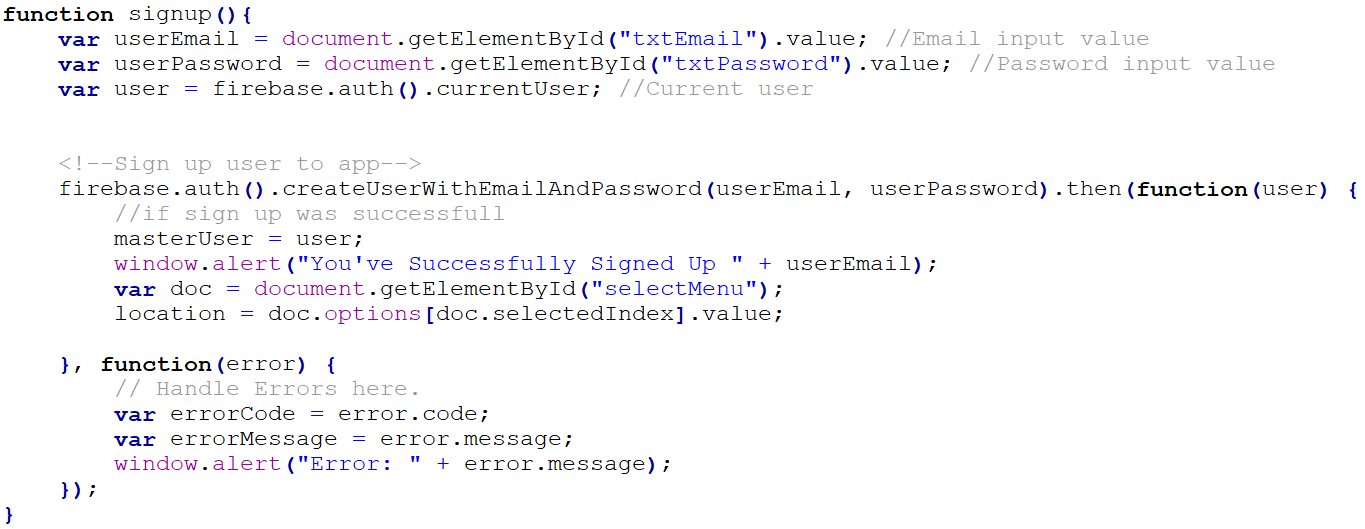
To start the development of the website, I decided to create the sign up and login function of the website first because it’s critical feature of the website and will grant users access. With the basic HTML structure in mind, HTML5 and CSS3 was used to design a web form for users to input data required to sign up such as email and password. The button was disabled with the use of JavaScript (JS) and jQuery if all fields are empty, email field is empty, or the passwords done match (*see Figure 8 for JS and jQuery code*).



**Figure 8: JS & jQuery code for disabling 'sign up' button.**

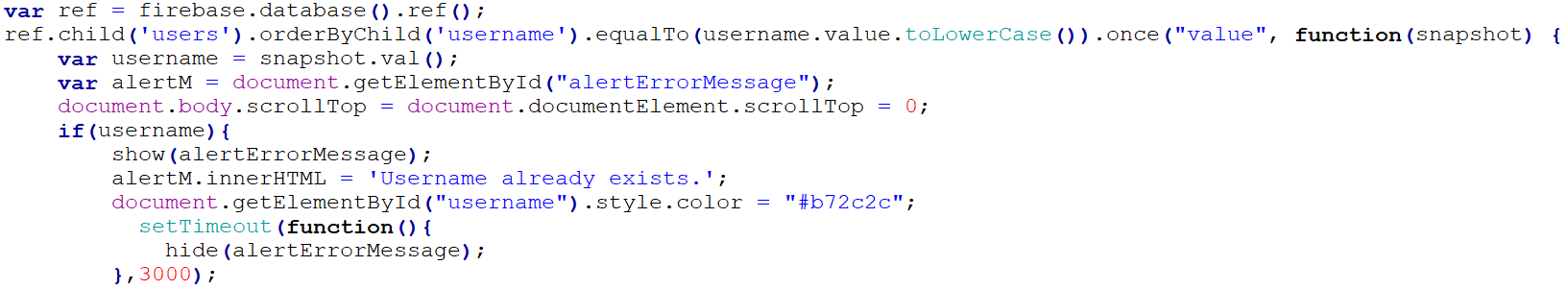
What come next was one of the biggest challenges faced during this project. Implementing a way to authenticate users to the web application was done using Firebase Authentication. Prior to this project I had no experience with JSON database. However, after reading through the documentation provided by Firebase, it wasn’t too hard to connect the web application to the Firebase Real Time Database and Authentication (Firebase, 2018).

First, I was required to copy a personalised snippet of code and paste it into the *<head>* of the webpage to initialise Firebase. In this code contains two script files, one with a Firebase JS file, and another contain a unique API key, domain name, database URL, and project ID. After adding the code to web page, I used the provide ‘create user with email and password’ method to create user by their email and password. This method takes two parameters, email and password; therefore, I stored the users input and used the values to fill the parameters for the method. This method was contained in a function called ‘signup’ which is called when the user clicks the sign-up button (*see Figure 9*).



**Figure 9: JS function containing firebase method to sign up user with email and password.**

The user is then redirected to another page which also contains another web form. This is where the user will enter the profile details such as name, contact information and the address they’re renting. Each user must have a unique username therefore, when the user attempts to submit their details, the username they’ve inputted is queried in the database to find a match (*see Figure 10*). If a match is found, then the users is prompted to try another username. Whereas if successful, then the user is taken to home page only when the required input fields are not empty. If the user is a landlord, then a pop-up is opened where they can register all properties they manage. They have the option to skip this process but will be prompted again to register their properties.

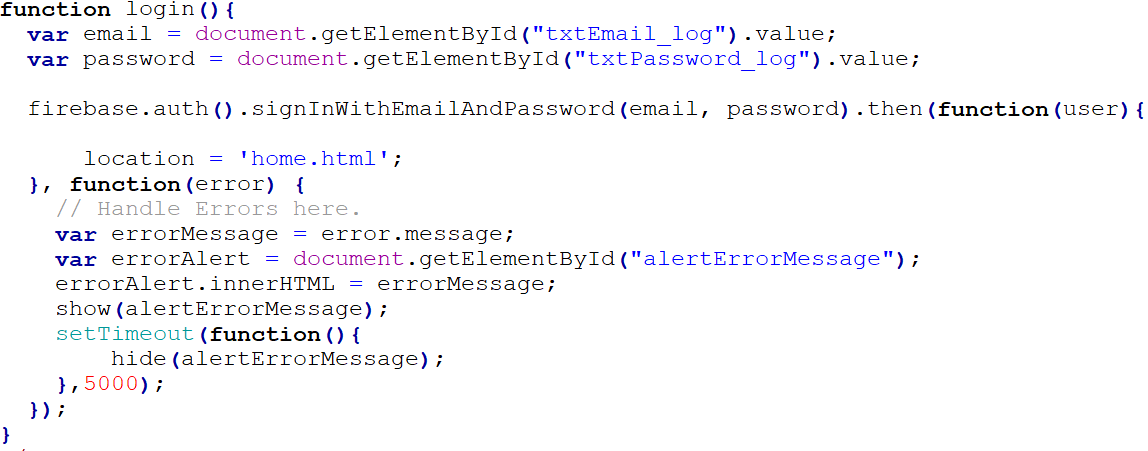


**Figure 10: Database query to find is username exists.**

### Login Page

The log in page simply has one function which is to provide a way for existing users to log into their account. I used the design template of the ‘Index – Sign Up’ page but changes the page content container to fit the purpose of the page. I created another web form that takes two inputs, email and password. Next, I used JS to collect the value of the user inputs and used the Firebase ‘sign in with email and password’ method to check the database for matching credentials and sign in a user.

This method was placed in a function called ‘login’ which is called when the user clicks the log in button. I used the ‘then’ promise to handle the succession and errors of the method. If the method is successful, the location of the page is changed to the home page using the JS object ‘location’. However, if there is an error, I used ‘function(error)’ to get the error message. I created an empty <div> under the login in button. I created a class called ‘hidden’ and used CSS to set the display of the element containing that class to none. I created another class called ‘show’ and used CSS to change the display to block. For each class I created a JS function with the parameter of an element, so it’ll be easier to hide and display elements. The <div> used to contain the error message will have the class ‘hidden’ unless there is an error. If an error occurs with the log in function then I used the JS property ‘.innerHTML’ to set the content of the div to the value of the error message and called the ‘show’ function to show the error message div. To hide the message after a certain amount of time, I used the ‘setTimeOut’ method to call the hide function with the div’s ID as a parameter after 4000ms (4s) (*see Figure 11*).

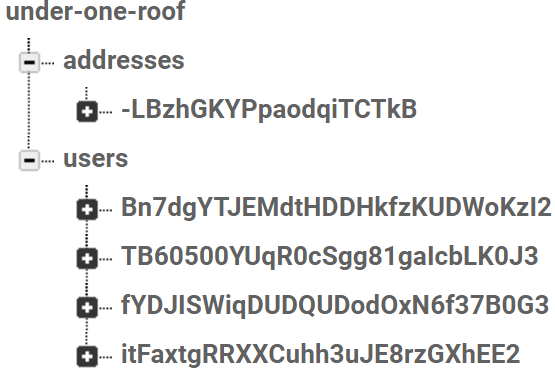


**Figure 11: function containing method to log in existing users.**

I used test data to test the functionality of both create user and sign in methods to ensure they work perfectly before moving on to the next stage.

### Structuring & Implementing Firebase Real Time Database

Designing the structure of the web application database was another huge challenge for me. One of the unique features of the web application was each property registered would have its own unique home page which will be viewable by its registered tenants and landlord. Secondly, if a landlord user has multiple properties, they would need to be able to switch between each property only showing the details regarding that property throughout the website. It was a challenging task to design a database that I’ve never used before, that meets these requirements but after planning and research, I was able to put together the right structure (*see Figure 12, also see* ***Appendix One*** *for full database structure breakdown*).



**Figure 12: Compact Firebase RTD structure**

The first key under the database’s root directory is ‘addresses’. This is where the all properties registered on the website will be stored, each under their own unique key ID. Under their ID is where all the information about that property is stored such as registered tenants, maintenance issues, notification details, current lease details, and messaging service chat history.

The second key under the database’s root directory is ‘users’. All details of the user are stored under this key, with the user’s information stored under their user unique identification (UID) that’s generated when they register. Under their UID will also have their renting addresses, reminders they create, and setting to their profile. For landlord user; tradesman details and their home address will also be stored under their UID.

To tackle to problem of allowing only authorised user to see a property’s home page and information. I used the property’s unique key as an access key. Therefore, to access the property’s home page and information, the user must be in possession of the access key. How I implemented this method is further discussed later in the report.

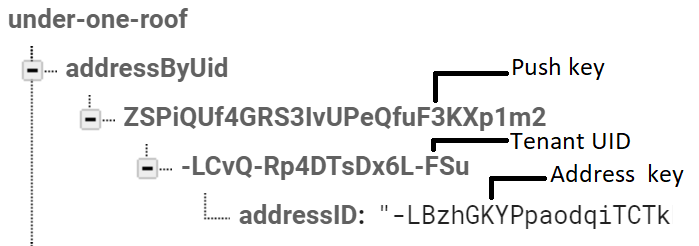
### Home Page

The home page is arguably one of the most important pages throughout the whole web application. On this page the user can perform many activities within a few clicks and is the connection to other pages within the website. Again, similar to the previous page, I used the basic HTML structure and styling as the other pages but changed the contents of the ‘page-contents’ container.

First, I designed the menu icon by using HTML5 elements ‘div’ and ‘span’. I used CSS3 to create padding, margins, and change the background colour to look like a typical three-line menu icon also known as ‘hamburger’. Next, I created a div which contains the menu items and used HTML elements <ul> and <li> to create the list items. In the top right corner, I used the Font Awesome exit icon which is used to close the menu. At the bottom of the list items I created a sign out box using the div element which will act as button to sign out users. I used CSS to change the colour of the text and background of the div when the user hovers over it to show that it’s a button. I styled the menu div and items to have a gun metal grey background that is slightly transparent with off white writing to have a simple effect that’s easy to read and user friendly. Keeping with the ‘mobile first’ method, I set the width of the menu to fit the whole screen and used media quires to change the width on larger screens.

To provide the functionality of the menu icon I used JS and jQuery to show and hide the menu when the menu icon or the exit icon is clicked. With CSS ‘transition’ I controlled the way the menu is shown and hidden but slowing the transition of its width for 3 seconds instead of being instant. This was something I never used before and was useful to learn because it provides a smooth effect to the transition of element sizes.

When a tenant user is directed to the home page after logging in, if ‘registeredProperty’ in the database under their UID is an empty string then a pop-up box is displayed using the show function. In this box contains the address they signed up with and the name of the landlord which is retrieved from the database by using the UID stored in ‘Landlord’ under the addresses key in the database. They’re then prompted to confirm if the details are right, if so, then a request is sent the landlord. At first, I stored the request under the addresses key in the database, but it was difficult to query due to bad structure. To overcome this problem, I created a new key under the root ref called ‘addressByUid’. With the tenants UID as a key and the addresses UID stored as a child of the key, it was easier to query the request (*see Figure 13*).

****

**Figure 13: New key created to handle tenant request queries.**

If the details were incorrect then the user has an option to search for their landlord by their username. jQuery ‘key up’ method was used to find a match to the users input in the database. Once matched is found, the user UID of the landlord is used to find all address the landlord has registered. All of the address found is are appended to the pop-up box with a send request button where the user can send a request to join the property. Request handling is discussed further in ‘4.2.5.5. Tenants page’.

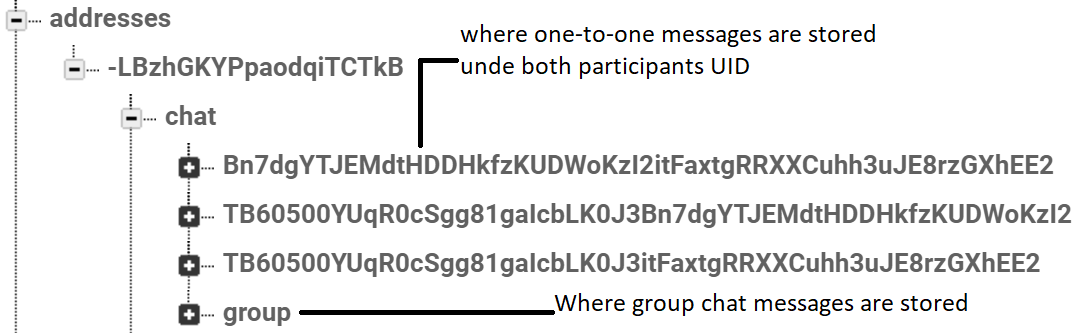
When a landlord user is directed to the property home page for the first, if they’ve registered a property, they’re prompted to select one to load. Once loaded, if the property has no registered lease, they’re prompted to complete a form for the details of the current Once completed, they have full access to the rest of the website.

The payment details of the lease are displayed in the ‘Lease Payment Schedule’ div on the home page. This is the section where both landlords and tenants can see the current status of the rental payment. The landlord can view they payment history, log payments and add a late payment fee with the use of HTML buttons which is reflected on the information shown in the div.

A notification div is also present on the home page. This is where the latest notification regarding the property can be viewed. Personalised notifications such as reminders are also shown in the notification div but only for the user who created it. Each notification is stamped with a ‘for who’ child in the database to which is used to show certain notification depending on the value of the child. They’re also stamped with the Firebase timestamp so that when queried, they’re ordered and appended to the div according to the time stamp (descending).

The third and final feature of the home page is the instant messenger between the registered tenants and the landlord. The messenger container was created from divs and is split into three sections. The first section in the far-left side where all participants of the messenger are displayed in categories (landlord and tenant). They names are listed with their current online status underneath. The second is the message display, where each message is pulled from the database and appended to the div according to their timestamp, similar to the notifications. The third section is the message input, this is where the user types a message and send it to the group. Each message is pushed to the database with the UID and username of the sender, a timestamp, the date it was sent, and the time it was sent. (*see Figure 11*).

The user can also start a one-to-one messenger with any user by clicking the username of another user in the left div. This opens a new tab where the content of the chat is stored under a key with the value of Both of the chat’s participants UID (*see Figure 14*).

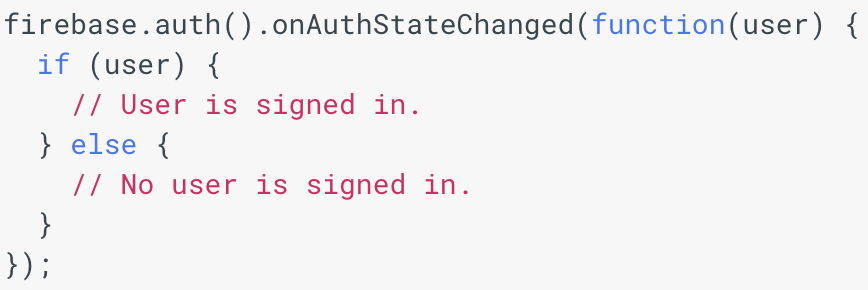
****

**Figure 14: Database structure for group and one-to-one messages.**

### Menu Item Pages

#### Profile Page

The profile page is where all of the user’s details are displayed for other users within the property group to see. The page is separated into three sections; basic information, contact information, and emergency contact information for tenant users.

****

**Figure 15: Get the currently signed-in user method.**

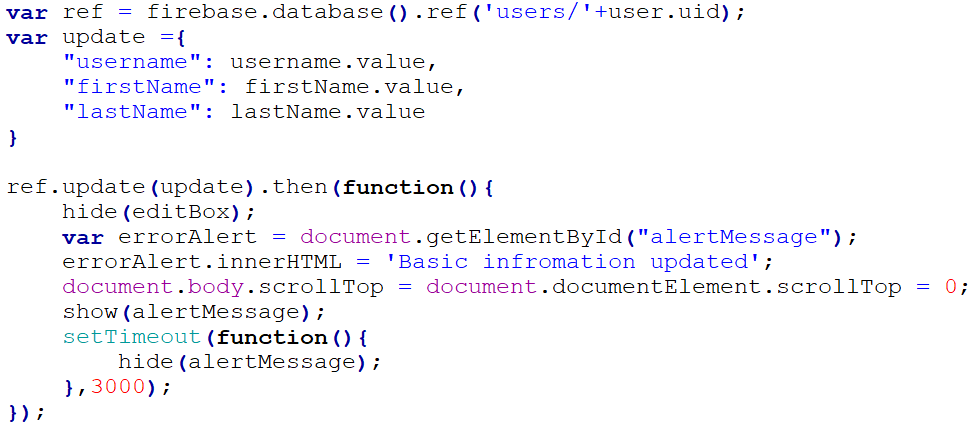
To pull the information from the database and display it in HTML elements I first checked to see if the user is currently signed in but using the ‘get the currently signed-in user’ method (*see Figure 15*). If the users is not signed in, I used the location object to redirect them to the index page. However, if the user is signed in I used the UID to query the data in the database and find their details. At first, I wasn’t too sure how to go about it therefore I watch YouTube videos and search Stack Overflow but the methods they used didn’t work for me. I eventually found a solution on Stack Overflow that reads the information from the database by directing to its location (*see Figure 16*).

****

**Figure 16: JS code used to pull users first name from database.**

I reused the code for each value I needed to read from the database. Further into my project I discovered a more efficient way to get multiple data under the same key. Instead of reusing the code for each value, I could stop the directory at the child containing all the data I wanted to read such as the user UID child. Then use data.val().*nameOfDataKey* for each value (e.g. ‘data.val().firstName’ will get the value of firstName).

Each information container has an edit icon which is used to edit and update the information. To do this I used the update method with the parameters of the child and it’s new value. To keep my code clean and easy to read, I created a variable to contain the child and it’s new value and used the variable as the parameter to the method. (*see Figure 17*).



**Figure 17: Code to update database.**

#### Properties Page

The property page has a very simple structure and is only available to landlord user. I used the hide function (mention in 4.2.2. Login In Page) to hide the link to this page in the menu from other users. I done this by checking the users ‘user type’, if they’re a tenant then the hide function is called to hide the link.

At the top of the page is a button to add/register a property on the website. This button shows a pop-up box where the user can enter the details of the property and save it in the database. To do this I used the same technique as the update method but instead of using ‘update’ I used the ‘push’ method to push data with a unique key to the appropriate location in the database.

To display the users registered property I used the same technique to collect the user details on the profile page however instead of using the ‘value’ event, I used the ‘child\_added’ which is triggered every time a child node is modified. With the data that’s collected from the database, I used jQuery to append the data wrapped in HTML to the body of the webpage. The reason I used the child added event listener is because every time the users adds a property it’ll append the data to the body without the need to refresh the page to see the changes made.

An edit and delete icon is present in the top corners for each property the user has registered. When the property is edited, I used the same method to update the database as I did on the profile page. To remove the property from the database when the user clicks the remove button, I used the ‘remove’ method instead of the ‘update’ method. The user is also presented with a confirm box to give them an option to cancel the deletion or continue with it in case they pressed the delete icon by accident. For both icons, I set the value of the element value to equal the first line of the address. I also set the parameter of the on click function to ‘this’ to retrieve the value of the value attribute when the function is called. In the function, I used parameterName.getAttribute(‘value’) to get the value attribute’s value. I done this so when the delete button is clicked, it knows what property to query and delete.

#### Transmen Page

The tradesman page is also a page only accessible by landlord users therefore I used the same method to hide the link in the menu from other users as I did for the property page. This page has a very similar layout to profile page and functions like the property page as it appends each tradesman the user has registered in the same way.

So that the user can contact a tradesmans easily, I wrapped the email address in an anchor element with the attribute (href=’mailto:emailAddress’). This will open up the email system with the email already in the ‘to’ section of the email template ready to send an email. I done the same for the phone know but replace mailto with (<tel:mobileNumber>).

#### Maintenance Issues Page

The body maintenance page consists of a HTML table with a button to create a maintenance issue at the top. Each maintenance issue created is read from the database and appended to the HTML table in the same way the tradesman details and properties are appended to the body of the web page. When the title of the maintenance issue is clicked it triggers a function that show a pop-up box containing all the details of the maintenance issue. In the top corners are two Font Awesome icons to edit the issue and email it to a tradesman. The edit button is only visible to the create of the issue and the landlord while the email button is only visible to the landlord. When an issue is created, the UID of the creator is sored with the details of the issue which is then used to query if the current user can see the edit button.

When the edit button is clicked, the content of the pop-up box is cleared using (.innerHTML = “”) and the editing fields are appended. Only the title and the notes of the issue are editable, and the issue is updated on the database using the update method. If the user tries to exit the editing process they’re prompted with a confirm box and informed that if they exit the details they’ve entered will not be saved. This is known as ‘forgiveness’ which allows users to undo mistakes.

When the email button is clicked, it triggers another pop-up box to open where the user can select a tradesman to email the issue to. When the email icon is selected it opens the users emailing system with the tradesman email placed in the ‘to’ section, the issue title as the subject of the email, and the details of the issue in the body of the email.

#### Tenants Page

The tenant pages contain the full name a contact details of each registered tenant of the property. It’s laid out exactly the same as the tradesman and profile page however each tenant has a button to view their full profile. At first, I started to create a duplicate profile page to display the details of the tenant, but during that process I thought there must be a more sufficient way to view a profile from the ‘tenants’ page.

I thought of a solution which involved storing the UID of the tenant in local storage and when the profile page is loaded, check to see if the tenant UID local storage is empty, if so read the current user’s details from the database, if no pull the details of the UID contain in the local storage. I then used ‘window.onbeforeunload’ which is triggered before a page is unloaded to clear the tenant UID local storage. This is to prevent the details user of the UID stored in local storage being read from the database when the current user leaves and views their own profile.

In the top left corner of each tenant appended to the page is a speech icon. When clicked, it opens a new tab where the current user can speak to that tenant in a one-to-one messenger. If the current user is a landlord, a delete icon will be visible in the top right corner. Again, the tenant will be removed from the database using the remove method and the user will be provided with a ‘forgiveness’ option to cancel the process.

If the user is a landlord, when the page is loaded I test to see if the user is signed in. If they are, I query the database to check if any tenants have requested to join the property. If there are request for the current property, a pop-up box is shown with a list of all request for the property. The user then has an option to accept or reject the request. If accepted the tenant is added to the property group.

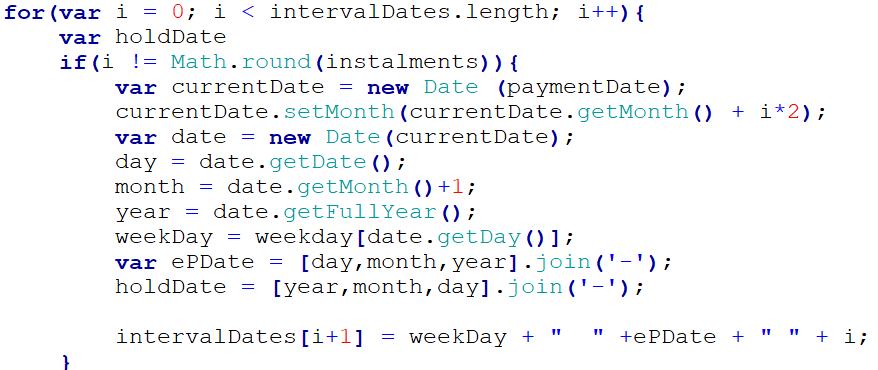
#### Documents

The documents page was intended to provide the function to upload important document involving the property. Unfortunately, I couldn’t find a solution to store files and images to the Firebase database, although this is something I would have certainly looked into more if I had more time.

#### Expenses

The expenses page if another page for landlord users only and was hidden from other user in the same way as the other pages were hidden. This page has a very similar lay out to the maintenance page and function exactly the same in terms of creating and editing expenses.

For each expense displayed in the HTML table, there’s a button to view the payments of that expense. When the button is clicked it opens a pop-up box displaying the payments of the expense and when their due. To calculate the instalments, I calculated the total number of days between the start date and the end date and divided it by the amount of days in the interval of the expense to get the number of instalments (e.g. weekly = 7 days). Within a for loop that runs the length of the instalment, I used the JS date object to add the interval period to the current loop date which produces the instalment dates (see Figure 18).



**Figure 18: for loop to calculate expense instalment dates for ‘every 2 months’ interval.**

To display the due message, I used an if statement to check if the instalment date is greater than the current date. According the outcome will determine what due message is displayed in the pop-up box.

When the expense title is clicked is opens another pop-up box containing the details of the expense and the option to edit it. When editing, only the notes and paid Boolean can be edited. If the user exits before saving, they’re presented with a ‘forgiveness’ option confirms they want to leave before saving.

#### Reminders

The reminders page is structure the same as the maintenance issues page, with a HTML table and button to create a reminder. The reminders and pulled from the database and appended to the table. When the title is clicked, the details of the reminder are displayed in a pop-up box which can be edited and deleted.

#### Settings

The setting page is regrettably the last page I created. It contains settings for the profile page, notification, and account. It has a small internal navigation, with each section created from the div element. Each section can be edited to meet the user’s needs which is updated on the database using the update method.

At this stage I had already created tested and received stakeholder’s approval of the other pages therefore after I has finished with the settings page, I realised I was required to change the way information is displayed according to the settings information on the database. Unfortunately, I had no time to do this however the user can still control setting, but it won’t have an effect on the rest of the website

# Chapter 5 – Testing

This chapter will explain the testing process carried towards the end of the development stage.

## User Testing

Testing of the web-application was done throughout the development stage to ensure each function works as It should. However, I gathered the close group of stakeholders for an hour to test the functionality of the website and provide their feedback. To keep the testers anonymous, I will refer to them as tester one and tester two.

Tester one is a landlord and used the website as they would if it was completed and available online. While they used the website, I took notes of the way they used it and the comments they had. The result from tester one is as follows:

* Developer – “User looks unsure to what the mobile gif image is on the index page and seems confused and uncertain.”
  + This feature was intended to show a glimpse of using the web application on a mobile device. However, as a user and not a developer, it can be unclear to what the gif means and its purpose.
* Developer – “User enter extra white spaces when entering their address which caused problems when querying address.”
  + Solution - When user saves details to clear white spaces at beginning and end of string and duplicate white spaces.
* Tester One - “when I edit my profile page I’m not informed that the data will not be saved if I exit”.
  + Solution - Prompt user to confirm they want to exit editing process.
* Tester One - “when I accept tenant request and there are no more left, the box should disappear, but it just stays there although its empty.”
  + Solution - Hide tenant request pop-up box when there’s no requests.
* Tester One – “I can’t email my tenants as a property or as a whole group”
  + Solution – Allow landlord user to email tenants as a property group and a mass email to all of his registered tenants.
* Tester One – “I created the lease and saved it bit I’m not able to edit it.”
  + Solation - Allow user to edit lease.

Tester two is a tenant and was also informed to use the website as if it was available online and to provide feedback as they go along. The results are as follows:

* Tester Two – “When I submit my details after signing up I’m not informed if my details were successfully submitted”
  + Solution – Display message to inform user details have successfully be submitted to improve the user journey.
* Tester Two – “When the landlord either accepts or rejects my request, I’m not notified.”
  + Solution – inform user that their request has either been accepted or rejected by landlord to improve user journey.
* Tester Two – “I can’t view my landlords profile not access his contact information.”
  + Solution – Allow tenant users access landlords profile.
* Tester Two – “When waiting for landlord to respond to request, the background of the website is still visible and is confusing”
  + Solution – “When a tenant user is waiting for request response, hide the content of the website in the background.”

The results from both testers were analysed and used to correct and change the imperfections of the website to make it more user-friendly. This is a very beneficial way to test the functionality of the web application meets the user’s requirements. All of the above mention missing features have been implemented since the testing workshop. The web application was then tested again and was happy with the final deliverable.

# Chapter 6 – Conclusion

In this chapter I will discuss the conclusion of the project, assess the succession of the project as a whole and potential improvements, then reflect my personal overview on the project.

## Project Conclusion

The project was carried out as planned to use an agile approach. It was structure, developed and tested in a professional manor which is reflected on the final deliverable.

Developing and designing the structure and the layout of the project went extremely well because it’s something I enjoy and feel comfortable doing. The background research also went very smoothly, and I discovered more that helped throughout the project than I initially anticipated. The requirements analysis stage wasn’t so intimidating as it seems to be. Conducting the interviews helped me to understand the importance of gathering requirements from stakeholders in a different way than reading it in a book.

Some aspect of the project didn’t go to well. When developing the web application, I didn’t take into consideration the impact of coming across problems has on time. I would be lying to say the project was a smooth process, some problems that I faced took me days to solve. Due to the agile approach I was able to move on or start something else and come back with a fresh mind to tackle certain problems.

One of the memorable problems I face was thinking of a way to create a personal home page for a property that is the same for all user related to it but has personal features such as reminder alerts. In the initial stages of the project I was looking for way to dynamically create HTML web-pages when a user creates an account, which took a while for me to realise I was asking the wrong questions.

In the interim investigation and planning report, I initially planned to use the SendBird SDK to implement the instant messaging service on the web application. However, as my understanding and knowledge how to user the Firebase database. I was able to design a way to the instant messaging app to be implemented without the SendBird SDK but with the use of HTML, CSS, JS and the Firebase database.

Overall, the project went well and a lot better than I expected. The only thing I would do differently is interact with the stakeholders more and code neater when developing.

## Assessment of Succession of Project

In the early stages of the project, aims and objects were set to be accomplished by the end of the project. Some were added and changed to fit the final deliverable which is discussed in the beginning of the report (*see* ***Appendix One*** *for original copy*) The current state of the web application fulfils the aims and objectives and will hopefully become a useful tool for targeted users one day. Throughout the project when I reached out the close group of stakeholders, positive feedback was always received with a couple suggestions and changes which was a good sign that the project is moving in the right direction to be successful.

To professionally assess the succession of the project, quality checks were mentioned in the interim and planning investigation report prior to the project. All quality checks were met despite one which is ‘all user reequipments men’. Controlling the settings of how the user interaction and share their data with each other and the application was the requirement that wasn’t met. Although the user could change the setting, it didn’t have an effect on what setting changed. Due to this, the project can be deemed as a failure, however all other requirements were met, and stakeholder are pleased with the final application which makes the project successful.

## Personal Reflection

Personally, I enjoyed every step taken in developing this web application. It allowed me to use all of my skills I have learnt prior and during university and push myself to learn and do things I wouldn’t normally do. This project help to expand me skill is web development and taught me important aspects of property management I can use after I graduate.

I was to continue on this project outside after submission, one feature I would certainly add is the ability to make payments though the web application and set up standing orders. I believe this feature would make the web application a big competitor in a big market.

I can proudly say; the process and completion of this project is one of my proudest achievements and I hope to use as part of my personal portfolio to help achieve a successful career in web development and the IT industry.

# Bibliography

Author Online , 2018. *Communication between Landlords and Tenants Integral to Solving Disputes – How Property Management Software Can Help.* [Online]   
Available at: https://www.arthuronline.co.uk/communication-between-landlords-and-tenants-integral-to-solving-disputes-how-property-management-software-can-help/  
[Accessed 03 04 2018].

Cederholm, D., 2010. *CSS3 for web designers.* 1st ed. New York: A Book Apart.

Citizens Advice, 2017. *Disrepair - what are the tenant's responsibilities?.* [Online]   
Available at: https://www.citizensadvice.org.uk/housing/repairs-in-rented-housing/disrepair-what-are-your-options-if-you-are-a-social-housing-tenant/disrepair-what-are-the-tenant-s-responsibilities/  
[Accessed 04 04 2018].

Cox, K., 2015. *Introduction to Requirements Analysis,* Brighton: s.n.

Firebase, 2018. *Authenticate with Firebase using Password-Based Accounts using Javascript.* [Online]   
Available at: https://firebase.google.com/docs/auth/web/password-auth

Firebase, 2018. *Firebase Authentication.* [Online]   
Available at: https://firebase.google.com/docs/auth/  
[Accessed 05 04 2018].

Firebase, 2018. *Firebase Realtime Database.* [Online]   
Available at: https://firebase.google.com/docs/database/  
[Accessed 05 04 2018].

Gasston, P., 2013. *he modern Web: multi-device Web development with HTML5, CSS3, and JavaScript.* San Francisco: No Starch Press.

GOV, 1985. [Online]   
Available at: https://www.gov.uk/private-renting/repairs  
[Accessed 03 04 2018].

Hall, Z. D., 2017. *15 essential websites that every landlord should know about.* [Online]   
Available at: https://www.telegraph.co.uk/property/landlord-guide/essential-websites-for-landlords/  
[Accessed 05 04 2018].

Harkin, K., 2013. *The crucial importance of requirements management.* [Online]   
Available at: https://www.theneweconomy.com/technology/the-crucial-importance-of-requirements-management  
[Accessed 06 04 2018].

Hodgson, D. P., 2014. *Conducting an effective stakeholder interview.* [Online]   
Available at: https://www.userfocus.co.uk/articles/conducting-an-effective-stakeholder-interview.html  
[Accessed 06 04 2018].

Keith, J., 2011. *HTML5 for web designers.* 1st ed. New York: A Book Apart.

PhpMyAdmin, 2018. *Bringing MySQL to the web.* [Online]   
Available at: https://www.phpmyadmin.net/  
[Accessed 04 04 2018].

Property Investment Project, 2013. *Landlord Property Inspection Guide & Notice Template.* [Online]   
Available at: http://www.propertyinvestmentproject.co.uk/blog/its-important-to-inspect-your-buy-to-let-property/  
[Accessed 03 04 2018].

Rollingsons, n.d. *What are my duties as a landlord?.* [Online]   
Available at: http://www.rollingsons.co.uk/Features/what-are-my-duties-as-a-landlord  
[Accessed 03 04 2018].

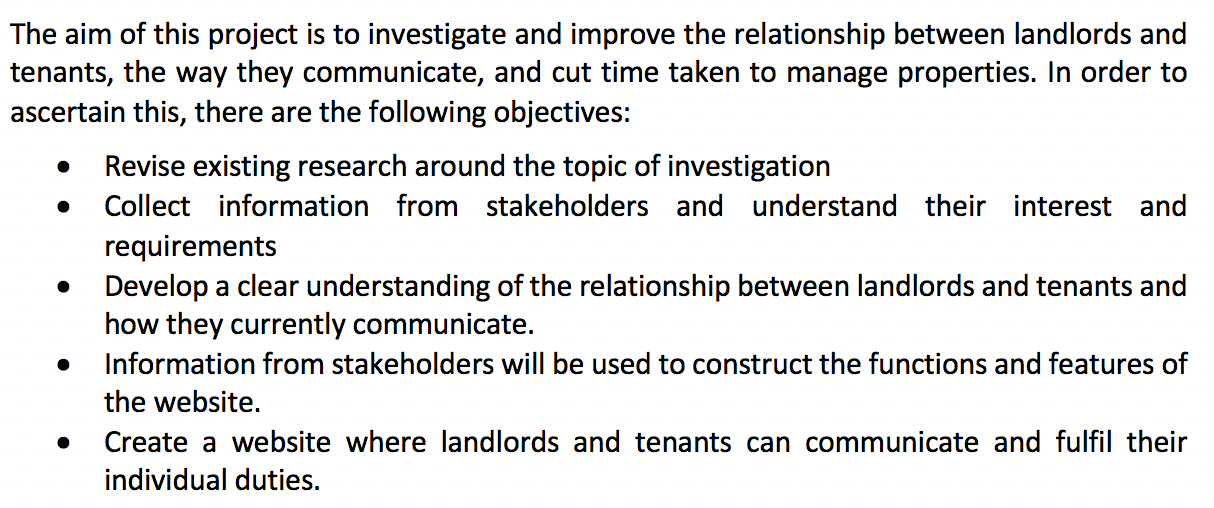
Shelter, 2017. *Tenant responsibilities.* [Online]   
Available at: https://england.shelter.org.uk/housing\_advice/private\_renting/tenants\_responsibilities  
[Accessed 04 04 2018].

TechTarget, 2017. *SQL (Structured Query Language).* [Online]   
Available at: https://searchsqlserver.techtarget.com/definition/SQL  
[Accessed 04 04 2018].

W3 Schools, 2018. *JavaScript Tutorial.* [Online]   
Available at: https://www.w3schools.com/js/default.asp  
[Accessed 04 04 2018].

# Appendix One

## Original Aims & Objectives

****

**Figure 19: Aims and Objectives created during the interim investigation and planning report**

## Landlord Responsibilities

* *“the property’s structure and exterior*
* *basins, sinks, baths and other sanitary fittings including pipes and drains*
* *heating and hot water*
* *gas appliances, pipes, flues and ventilation*
* *electrical wiring*
* *any damage they cause by attempting repairs”* (GOV, 1985).

## Interview Transcript for Tenant Interview

“*Introduction: I’m developing a website for landlords and tenants where they can carry out all their duties such as manage properties, manage rental payments, raise maintenance issues, instantly communicate, and send notifications.*

*Q1. As a landlord/estate agent/tenant, what would you expect from this website?*

*A1. I think communication between tenants and landlords is very important so to have a way to instantly communicate with the landlord is certainly a big expectation from the website.*

*(Interviewer) In an interview with a landlord, I stated that there may possibly be a notice board on the website where any actions made that both landlords and tenants should be aware will be posted to the notice board. I suggested that the notice board should be on the homepage and the landlord countered my suggest and stated that either a 1-1 messenger chat or group chat, depending on the amount of tenants, should be on the home page and the notice board be replaced with a notice centre like Facebook. What do you think of this?*

*I think that is a really good idea to have the messenger chat on the home page and maybe have the top ten notices on the side with an option to view all of them.*

*(Interviewer) and where would this option to view all notices take the user?*

*To a page where all of the notices are stored, and to make it easier to find a certain notice, it would be useful to sort notice by date and/or category.*

*Q2. What other tasks do you think the website should enable you to perform?*

*A2. Should be able to raise maintenance issues that are sent directly to the landlord, so it can be dealt with as quickly as possible. Tenants should also be able to see the progress of the maintenance issue and receive updates.*

*(Interviewer) How would you like to receive these updates?*

*Personally, I would prefer to receive them by text as I always have my phone on me however, when creating an account there should be an option on how you wish to receive notifications and update such as text or email.*

*(Interviewer) Anything else?*

*Making payments though the website would be useful.*

*(Interviewer) unfortunately payments though the website is not an option in this project is there anything else such as viewing rental payments you could find helpful?*

*Yes, being able to view rental payments and upcoming rental payments would be very helpful.*

*Q3. What information do you think would be useful to view on the website that would make it easier in day to day life?*

*A3. Rental payments, important dates such as when a property is being viewed or when someone is coming to fix something in the house.*

*(Interviewer) Okay, thank you for your time, if there is anything else you think you missed please contact me on the information I provided you with. If there is anything else I need a second opinion is it okay to contact, you on the information you provided me with?*

*Yes.”*

## Transcript for Landlord Interview

*“Introduction: I’m developing a website for landlords and tenants where they can carry out all their duties such as manage properties, manage rental payments, raise maintenance issues, instantly communicate, and send notifications.*

*Q1. As a landlord/estate agent/tenant, what would you expect from this website?*

*A1. I have two properties that I own and manage myself therefore I would need to be able to manage both properties on the website and fluently switch between the two on the home page. It would be useful for tenants to make rental and other payments through the website and rental information updated accordingly.*

*(Interviewer) In this case, payments made through the website is not possible, what would be your preferred solution for this?*

*If payments aren’t possible, setting a date when rent is due, logging payments on the website will be a suitable alternative.*

*(Interviewer) How would you like to be notified?*

*I respond quicker to emails as I’m always on my computer for work however it would be nice to receive both an email and text message. There should be an option when setting up to choose how to be notified.*

*(Interviewer) If there’s a problem making a rental payment how will you look to solve it.*

*Well, depending on what kind of problem it is, I will look to come to some sort of arrangement based on the tenancy contract. If not I currently have a condition, if rent is due 7 days late I add a £50 late payment fee. Also, a function to log payments like I said earlier should be on the home page, so I can quickly manage the rental payments……It would be nice if I could set a ‘reminder’ for both me and the tenant.*

*(Interviewer) what would you expect the reminder to do?*

*Send a message or an alert about the reminder to our email address and/or mobile phones and show on in notification.*

*Q2. What is the ideal way to receive and deal with maintenance issues?*

*A2. Currently if my tenants have a maintenance issue, they will email me with the start of the subject red/yellow/green/, red being a major issue and green being a minor issue. It would be nice if this same structure was implemented into the website.*

*(Interviewer) So when the tenants are creating a maintenance issue, they select a colour code for the severity of the issue?*

*Yes.*

*(Interviewer) How would you like to receive this maintenance issue?*

*Again, by both email and text.*

*(Interviewer) How will you deal with the issue?*

*Depending on what kind of maintenance issue it is I will contact the appropriate tradesman and have the issue resolved ASAP. What would be nice is to create tradesmen profile where I can set up an email address and forward the maintenance issue to them so they get it straight away.*

*(Interviewer) So create one or more profiles for a tradesman/tradesmen with their name/company name, email and number? And when you receive the tenant’s maintenance issue, forward it to the tradesman?*

*Yes, and when the issue is dealt with it can be edited and marked as ‘completed’ only by the person who created the issue and myself, the landlord and sent to the notification as an update.*

*Q3. There will be a noticeboard within the website, what do you think should be posted on it and should both landlords and tenants be able to write to it.*

*A3. That’s a really good idea, it would be nice to have a noticeboard on the homepage where both landlords and tenants can see it and be up to date with what’s going on. I think any action what is made that will be use of knowing should be posted to the noticeboard such as a raised maintenance issue and updates associating with it. I do think both landlords and tenants should be able to write to the noticeboard. Come to think of, it would be better if we replaced the noticed board on the home page with the landlord and tenants group chat. One of my properties have 5 tenants and it would be nice to have the group chat on the home page and if they need to contact me separately the can on a 1-to-1 chat.*

*(Interviewer) So instead of a noticeboard, remove it completely from the website and replace it with a group chat or 1-to-1 chat with the landlord and tenant(s)?*

*No not remove it completely I think there should still be a notification centre like Facebook has but just not on the home page.*

*Q4. Is there anything you’ll like to add?*

*A4. A place where I can manage my expense and properties on the website would be hand. I think that actions such as reminders and issues expenses and properties should be able to be amended and deleted.*

*(Interviewer) Okay, thank you for your time, if there is anything else you think you missed please contact me on the information I provided you with. If there is anything else I need a second opinion is it okay to contact you on the information you provided me with?*

*Yes.”*

## Functionality of Web-Application Including Changes

Changes requested by stakeholders when they reviewed document are in italic red.

### Home page – Login

The homepage page of the website will be simple and basic. It will contain two options, one for the user to sign in with their email address and password and the other will be to create an account as either a landlord or a tenant. Underneath will be information about the website, the features it offers, and who the target audience is.

*Login function to be on a separate page for excising users.*

### Signing up as a Landlord

When the user choses to sign up as a landlord they will be taken to the signing up process which is start by asking the user to input the following details:

* *Username (Added) (required)*
* First Name *(required)*
* Surname
* Contact Number
* Date of birth
* Occupation *(Removed)*
* Email address *(Removed)*
* Confirm email address *(Removed)*
* Password *(Removed)*
* Confirm password *(Removed)*

This is the information used to create their account “*and the email address and password they enter will be the information they use to log into their account”* *(False)*. Next, they will be prompted to input the following details of the property/property they are renting *(Changed to enter the details of their home address)*:

* First line of address *(required)*
* Second line of address
* Third line of address
* Town
* *City (Added) (required)*
* County
* Postcode *(required)*

Underneath will be a button with the option to add another property to their account. Once the user has finished entering their property details they will be prompted to click the next button to move on to the next stage of the signing up process considering all required fields are completed.

This above activity was changed to: “Once the user has completed filling their personal details and all required fields are completed. They’re prompted to move to the next stage with a ‘next’ link. When this link is created they are prompted to enter the details of the properties they rent as follows:

* *First line of address (required)*
* *Second line of address*
* *Third line of address*
* *Town*
* *City (Added) (required)*
* *County*
* *Postcode (required)*
* *Option to confirm if property is vacant or not.*

*In this section there will be a button to submit the address which saves the address and clears the field to add register another address. Once an address is added a done button will be visible to complete the process. There will also be an option to skip the property registration process which will take them to the home page. When the user registers a property, they are informed that property has been successfully submitted. When they complete the process, they’re informed that all of their details entered in the sign up process have been saved.*

“The next stage of singing up will be to add tenants to the properties they have registered. The registered properties will be listed on this page with the following fields underneath each one:

* Tenants first name
* Tenants last name
* Tenants date of birth
* Tenants email address
* Tenants mobile number
* Tenants Occupation
* Student number (if applicable)

Under these fields will be a button to add another tenant to the property. This process can be skipped and completed at a later date however, once the user has completed adding tenants to the properties they’ll be prompted to move on to the next step of the signing up process. (when the signing up process is completed with registered tenants. The tenants will be sent an email with a link to sign up.” *(Removed)*

The next step will be to add details of the tradesmen to their account. These details will be used to forward maintenance issues to the according tradesmen. The following details will be required:

* Drop down list confirming field of work including (plumper/electrician/carpenter/cleaner/Other) – with other there will be a text field to enter details
* Company or individual name
* Email address
* Phone number
* Emergency contact number
* Postal address

Under these field will be an option to add another tradesman. This process can be skipped and completed at a later date however once completed the user can move on to the final step.

*This step has been removed from the sign-up process to allow the user to do this in their own time.*

The final step will be to customise the settings of the users account. The settings will go as follows:

1. Notifications
   1. How would you like to receive reminders – Email/SMS
   2. How would you like to receive alerts – Email/SMS

Once this is complete, the user will have completed the sign-up process, their account will be created, and they’ll be taken to their personal home page.

*This step has also been removed from sign-up process to allow the user to do this in their own time*

### Signing up as Tenant

Tenants will be emailed a link to sign up either once the landlord has completed the signing up process or the landlord has added tenant to property. When they have clicked the link, they’ll be taken to a page to confirm the address of the property they are renting. Once confirmed they will be taken to the first stage of the signing up process. All the field that have been completed by the landlord will be filled in already such as name, address, mobile number etc. If not, then the user will be required to fill in the following details:

* *Username (Added/Required)*
* First name *(Required)*
* Last name
* Contact number
* Date of birth
* Email address (already filled in if sent link by email) *(Removed)*
* Password *(Removed)*
* Confirm password *(Removed)*

This is the information used to create their account. They will be prompted to go to the final step where they will confirm the settings of their account. The settings will go as follows:

1. Notifications
   1. How would you like to receive reminders – Email/SMS
   2. How would you like to receive alerts – Email/SMS

Once this is complete, the user will have completed the sign up process, their account will be created and they’ll be taken to their personal home page.

The user will not be prompt to go to the next stage but to enter the address of the property they are living in (renting). There will be a link to submit details and take them to the next stage in the user journey.

### Personal Landlord User Home Page

The personal landlord home page will consist of a variety of function and start of processes the user can perform. Starting from the top of the page, under the page header will be a button to switch between properties the landlord is leasing. On the left hand side of the page will be a sidebar that contains:

1. Account
   1. Link to user's profile.
   2. Link to settings.
   3. Link to user’s properties.
   4. Link to registered tradesmen.
2. Management
   1. Link to reminders for current property.
   2. Link to maintenance issues for current property.
   3. Link to current tenants of property.
   4. Link to important documents.

*Menu layout changed to as follows:*

* *Notifications*
* *Profile*
* *Properties*
* *Tradesman*
* *Maintenance Issues*
* *Tenants*
* *Documents*
* *Expenses*
* *Reminders*
* *Settings*

At the bottom of the sidebar will be a small calendar which will show important information such as reminders on the dates. When the dates are clicked it will trigger a pop up box with details of the important information on that date. If there is nothing created on that date, the user will be asked if they want to create an event/reminder.

*(Calendar will not be implemented)*.

Next to the left side bar will be the instant messaging chat between the landlord and the tenant(s). This is where the landlord and tenant(s) can communicate instantly. On the right *(Changed to left)* of the chat box will be another sidebar that contains:

* A section indicating all the persons who are involved in the chat and whether they’re online or not. The user will also be able to click a name of a participant of the chat to open a 1-1 instant messaging chat.
* Notification box that shows the 10 most recent notification. At the bottom of this box will be a link to ‘show all’ of the notification in a separate page and a button to create a new notification.

*Notifications will be in a separate container on the homed page either above or next to the messenger box depending on the user screen size. If the screen size is small such as mobile the only the notification box heading will be visible with a downward arrow to view the notifications.*

### Personal Tenant Home Page

There personal home page for the tenant will be very similar to the home page for the landlord but there will be a few difference. Starting from the top, unlike the landlord’s home page, the tenants home page will not contain the switch property button as tenants will not be leasing more than one property.

The right-side bar will look exactly the same as the landlord’s home page however the left sidebar will have slight difference. The right-side bar will consist of:

1. Account
   1. Link to user’s profile.
   2. Link to settings.
   3. Link to landlord’s profile.
2. Management
   1. Link to reminders.
   2. Link to maintenance issues.
   3. Link to important documents.

*Menu layout changed to as follows:*

* *Notifications*
* *Profile*
* *Maintenance Issues*
* *Tenants*
* *Documents*
* *Reminders*
* *Settings*

At the bottom of the sidebar will be a small calendar which will show important information such as reminders on the dates. When the dates are clicked it will trigger a pop up box with details of the important information on that date. If there is nothing created on that date, the user will be asked if they want to create an event/reminder.

Calendar will not be implemented

### Profile Page

The profile page can be accessed by clicking the link on the home page. Once clicked it’ll take the user to the profile page the contains all information about the user that they entered during signing up. At the bottom of the page will be a privacy button that allows the user to select what information they want other to have access to see *(This feature has been moved to the property page)*. Some information such as name will not be able to be change the privacy settings. With tenants, details of an emergency contact will be added to the profile.

### (Landlord) – Property Page

The property page can be accessed by the link on the home page. When clicked the user will be taken to the property page where all the properties they lease that is registered on the website will be listed. Each property box will also have button to ‘go to property’ which will take them to the home page of that property. *(It will also have and edit and delete button)*

### (Landlord) – Tradesmen Page

The tradesmen page can be accessed by clicking the link on the home page. When clicked the user will be taken to the page that contains all the details of all tradesmen the user has registered on the website. The tradesmen will be listed with their names and profession underneath. When their name is clicked it’ll bring up the tradesmen full details which will include:

* Field of work.
* Company or individual name.
* Email address.
* Phone number.
* Emergency contact number.
* Postal address.
* Website.

In the details dialog box the use will also have an option to contact the tradesmen by email. Setting for this page can be edited with the setting page.

### Reminders

Reminders is a feature available for all users. It allows the user to set reminders for important thinks such as payments dates and viewings. To access the reminder page, the user can click the link on the home page. The reminder page will then open with a list of all reminders related to them in order of date (reminder with the date closest to the current date will be at the top). The format of the list will be: the title/subject of the reminder with the date and first few words of the description (if there is one).

When the reminder title/subject is clicked it will open the details of the reminder such as:

* Title/Subject
* Date
* Frequency
* Description (Notes)

If the user was the create of the reminder, then they’ll be able to edit and delete it. Altering the setting for reminders can be done on the settings page.

### Maintenance issues

The user can access maintenance issues by licking the link on the home page. This will take them to the maintenance issues page which will contain a list of all the maintenance issues raised by both landlords and tenants. The Issue title, status and part of the description will be visible. On the page there will be a button to create a new maintenance issue with the following fields:

* Title
* Category
* Severity: Red/Amber/Yellow
* Description
* Notes

When the issues are clicked it will open a page with all details of the issue. If the users are the creator of the issues or landlord then they’ll be able to edit, save, and delete the issue. Any changes made to the issue will be sent to the notification board. Both landlord and the creator of the issue can pass the issue as solve/completed.

### Tenants Page

The tenants link on the home page will take the user to the tenant’s page where the tenants of the current property are listed at the top followed by all of the user’s tenants. The tenants will be listed with their name mobile number and an option to email the tenant. There will also be an option on the page to email the tenants as a property group or as a whole.

“When the tenant is clicked their details will be opened which include:

* Name
* Address renting
* Move in date
* Move out date
* Date of birth
* Email
* Phone number
* Documents
* Personal notes”

This has been changed to a button. Each tenant will have a button to view their profile page.

### Documents

This page can be accessed by the home page. When the link is clicked it will open the documents page where all the documents related to the property as a whole are listed.

### Expenses

The expenses page can be access on the home page by all user. When the link is clicked it’ll open the expenses page for the current property. On this page the user will be able to see expenses created. The users will be able to create new expenses, delete and edit existing ones.

## Standalone Comparison Table

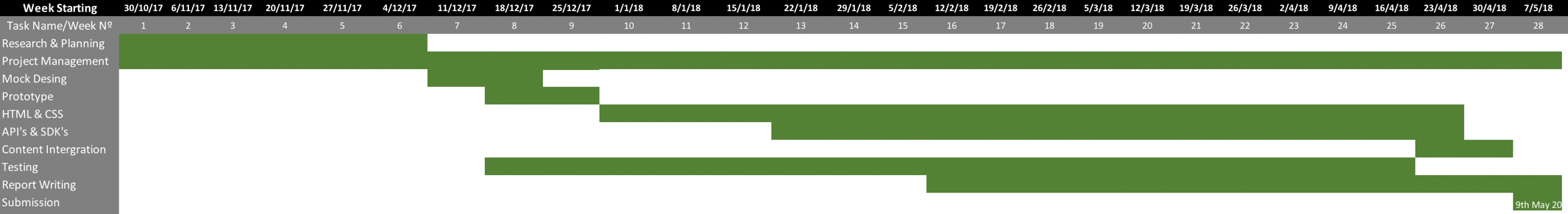
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Supporting Operating Systems | | |  | |  | |  | |
| Name | Windows | Linux | Mac OS | Cloud Based | Software Size | | Business Size | | Price | |
| Rent Manager | Checkmark | Checkmark | Checkmark | Checkmark | Unknown | | Small, Medium & Large | | Unknown | |
| Buildium | Checkmark | Checkmark | Checkmark | Close | 10GB | | Small & Medium | | Stating at £107.25 | |
| Property Boulevard | Checkmark | Checkmark | Checkmark | Checkmark | 12GB | | Medium & Large | | £175 | |

**Table 1: Comparing three standalone property management software.**

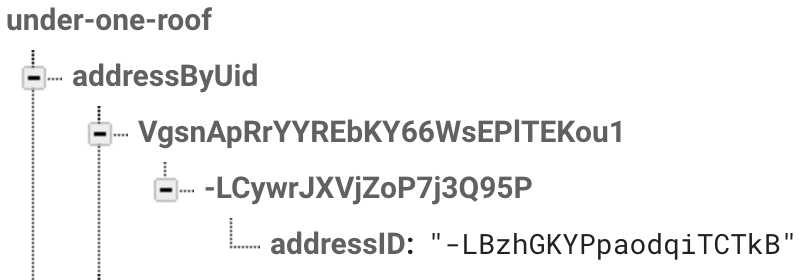
## Requirements Table

|  |
| --- |
|  |
| Stakeholder Interview Response | Requirements |
| “have the top ten notices on the side with an option to view all of them.” | Home page should have a notification board containing latest notifications and an option to view all notification. |
| “To a page where all of the notices are stored”  “should still be a notification centre like Facebook has but just not on the home page.  ” | The view all button will direct user to a separate page where all notifications are viewable. |
| “need to be able to manage both properties on the website and fluently switch between the two on the home page” | A button to switch between properties needs to be visible on home page. |
| “There should be an option when setting up to choose how to be notified.” | Setting to control notification setting must be available on the website. |
| “a function to log payments like I said earlier should be on the home page” | Log payments and view payment history on home page. |
| “if rent is due 7 days late I add a £50 late payment fee” | Log late payment fee on home page. |
| “Send a message or an alert about the reminder to our email address and/or mobile phones and show on in notification.” | Reminders should be sent to preferred device (email/mobile) and sent to the notifications. |
| “Currently if my tenants have a maintenance issue, they will email me with the start of the subject red/yellow/green/, red being a major issue and green being a minor issue. It would be nice if this same structure was implemented into the website.” | * Ability to create maintenance issues.   Categorise maintenance issue by severity (rend/amber/green). |
| “create tradesmen profile where I can set up an email address and forward the maintenance issue to them, so they get it straight away” | * Ability to create multiple tradesman profile.   Function to send issue details direct to tradesman from website. |
| “when the issue is dealt with it can be edited and marked as ‘completed’ only by the person who created the issue and myself, the landlord.” | * Ability to edit issue   When completed, issue can be marked as completed and sent to notifications. |
| “it would be better if we replaced the noticed board on the home page with the landlord and tenants group chat.” | Have an instant messenger between the tenants and landlord on home page |
| “if they need to contact me separately the can on a 1-to-1 chat” | Messenger must have the option to start a conversation with each induvial as (one-to-one) |
| “A place where I can manage my expense and properties on the website would be hand. I think that actions such as reminders and issues expenses, tenants and properties should be able to be amended and deleted.” | * Ability to create and manage expense. * Ability to manage properties. * A page to view and manage tenants.   Ability to edit and remove expenses, issues, tenants, and properties |
| **ADDED DURING DEVELOPMENT** | |
| Email (landlord stakeholder) - “on the home page where you can view payment, I need to be able to email a recipe of payment to tenants” | A button to send recipe of rental payment by email to tenants’ form view payments on home page. |
| Email (tenant stakeholder) – “on the index page (one where you sign up). If you’re already signed in you should be able to go to your home page. The signup box shouldn’t be visible either If I’m signed in” | * Users who are already signed in should be able to access their home page from the index page.   Create account box not displayed if user signed in. |
| Email (landlord) – “but if I have potential tenants of a property who are not users of the website, is it possible for them to access and view the maintenance issue page?” | Allow non-registered users to view maintenance page with an access password. |
| Email (landlord) – “can you make the add late payment fee button show only when its need?” | * Only show late payment fee button when its 7 days pass the due date and user hasn’t added late payment fee.   Ability to log payment 7 days late without adding late payment fee |

## Original Gantt Chart



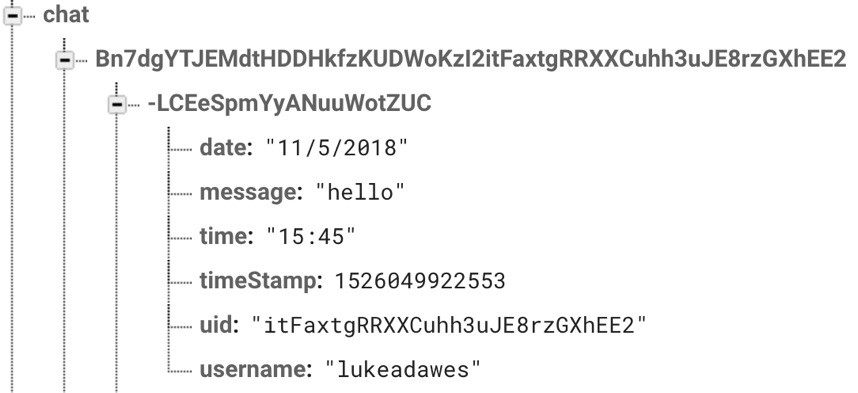
## Full Firebase Database Structure



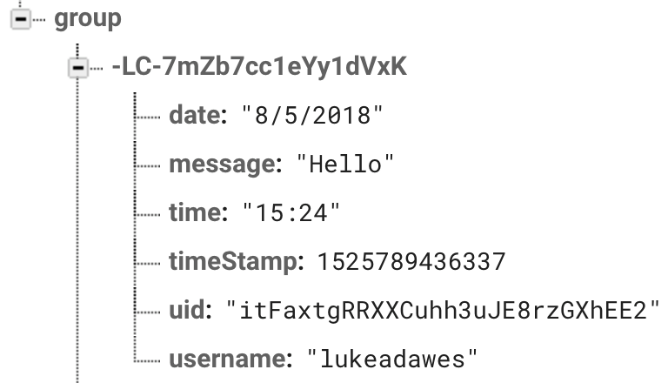
**Figure 20: Database structure to store and handle requests.**

****

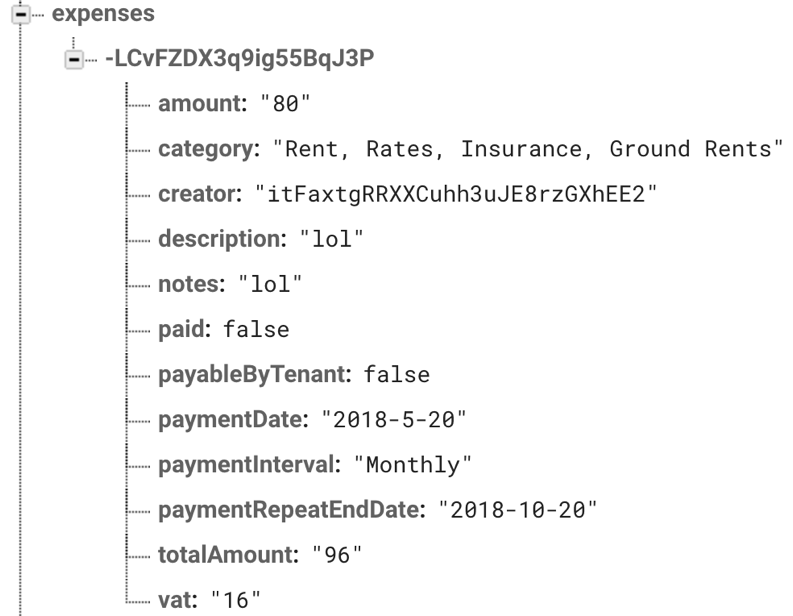
**Figure 21: Database structure for each registered property.**



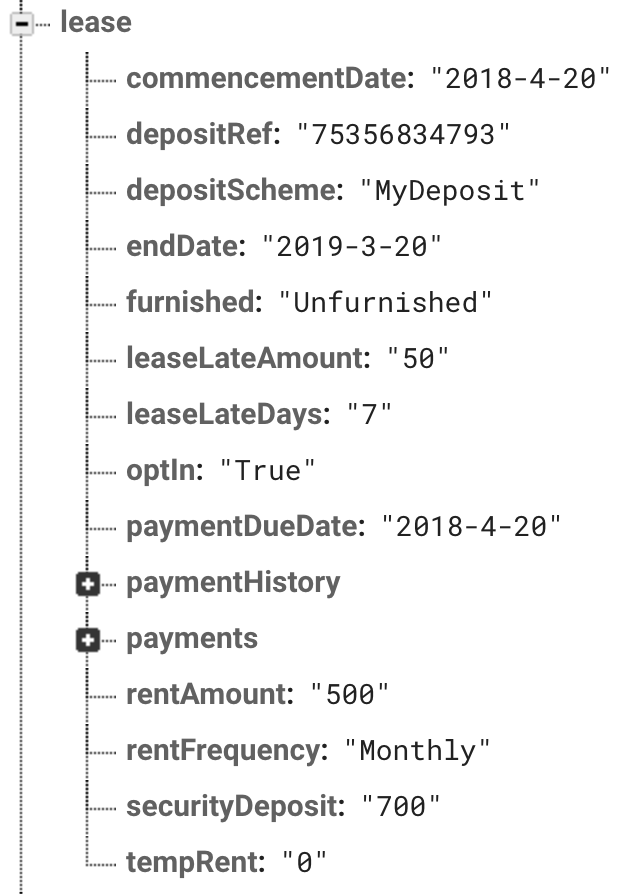
**Figure 22: Database structure for each message sent on 1-1 messenger.**



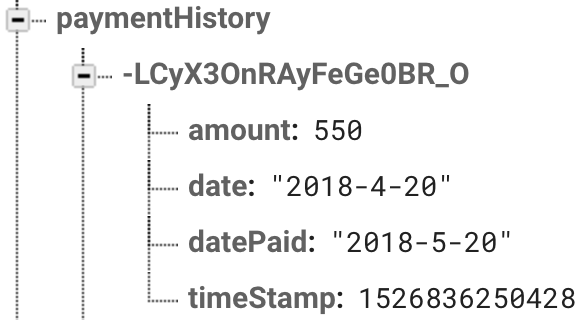
**Figure 23: Database structure for each message sent on the group messenger.**

****

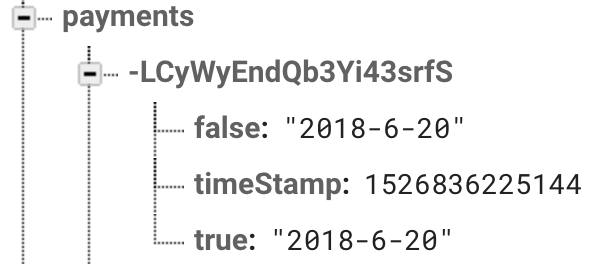
**Figure 24: Database structure to store each expense create.**



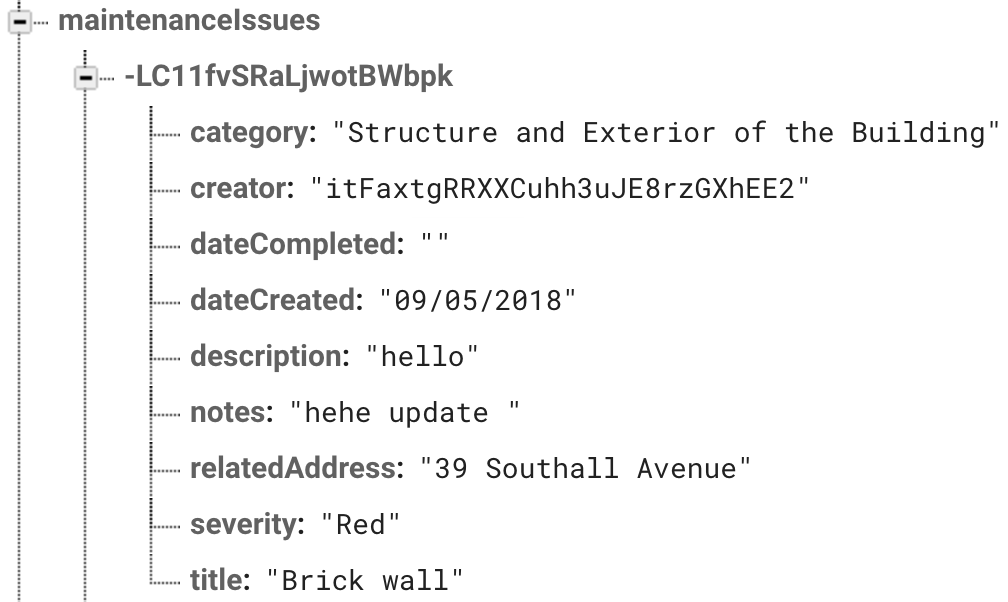
**Figure 25: Database structure to store lease information.**



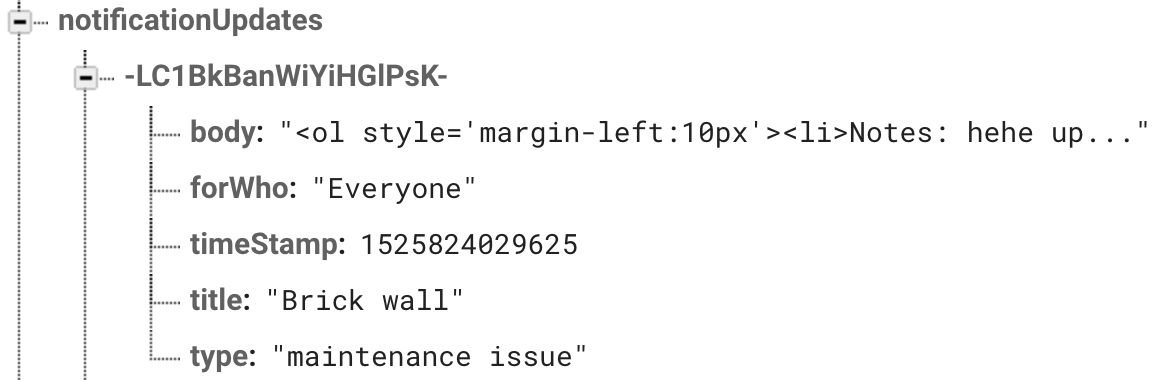
**Figure 26: Database structure to store the payment history of each rental payment.**

****

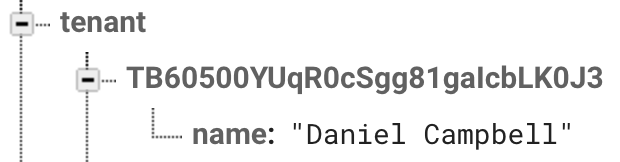
**Figure 27: Database structure to store each rental payment scheduled.**



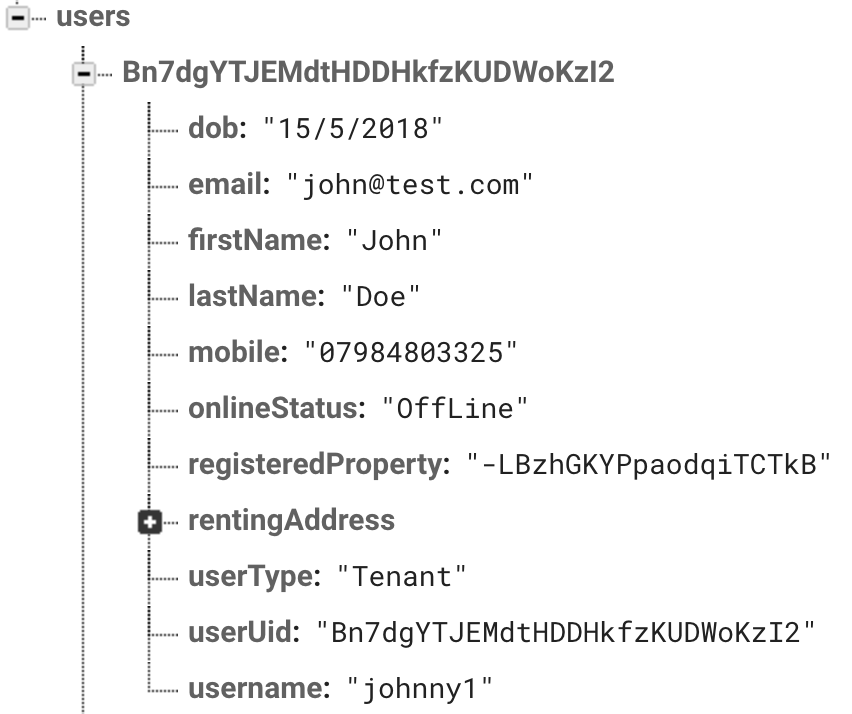
**Figure 28: Database structure to store each maintenance issue created.**

****

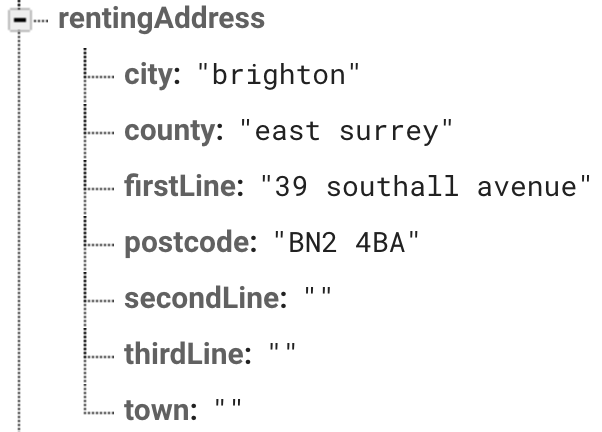
**Figure 29: Database structure each notification created.**

****

**Figure 30: Database structure to store each registered tenant to an address.**

****

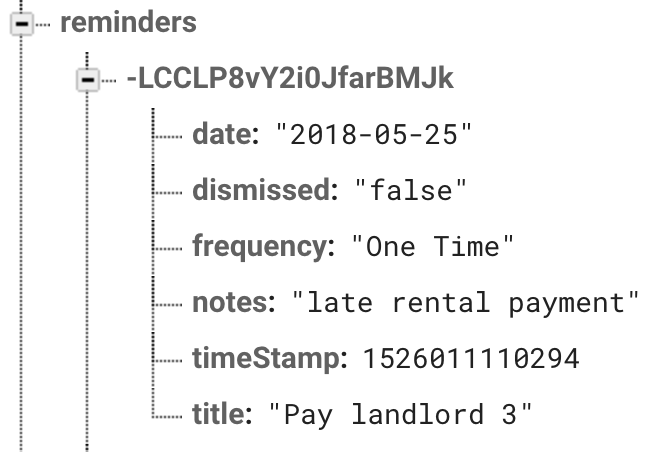
**Figure 31: Database structure to store registered user’s information.**

****

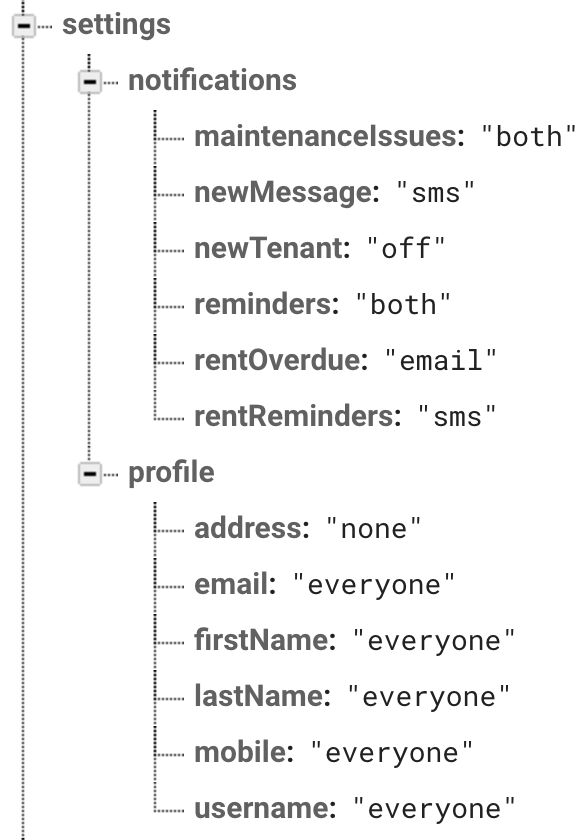
**Figure 32: Database structure how renting addresses are stored for tenant users**



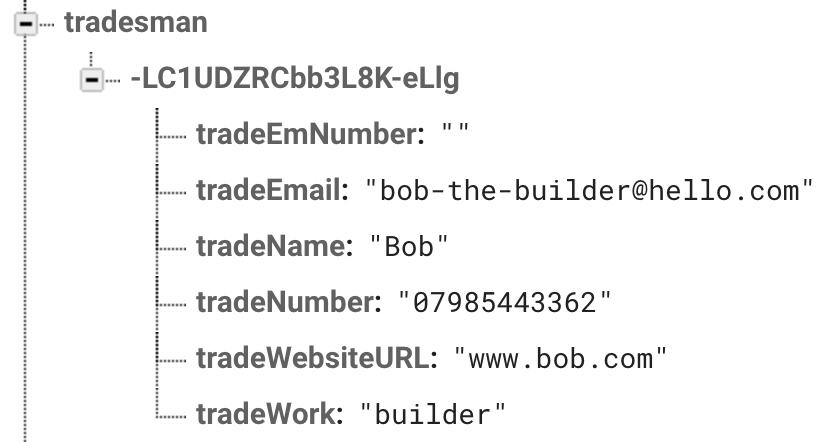
**Figure 33: Database structure how each renting addresses are stored for landlord users.**

****

**Figure 34: Database structure for each reminder the user creates.**

****

**Figure 35: Database structure to store user’s settings.**



**Figure 36: Database structure to store each tradesman details**

# Appendix Two

## Consideration of Ethical Issues

Good ethical practice was used throughout the project to protect all that are involved in the project. The two stakeholders who participated in the testing of the web application and interviews were informed that they had the right to withdraw as a participant and what their data will be used for. Their consent to participate in the project was confirmed before any type of data was recorded.

When handling data, I followed the regulations of the Data Protection Act (1998) to ensure participants data will be protected and secured. No sensitive data was collected from the participants and was referred to as a stakeholder or tester. Documents such as interview transcripts were stored in a locked draw and destroyed when no longer needed.

Since 25th May 2018, General Data Protection Regulation has changes the way individual’s data is handled in the EU. In the 19th week of the project I was made aware of this and since took the new regulation into consideration from then onwards.

Third parties supporting this project such as stakeholders were also informed of the university’s right to access of my intellectual property. This will include the final deliverable of this project therefore it was necessary to inform all supporting third parties.

## Consideration of Social Issues

A standard of social issues was always considered throughout the project. Web accessibility is an import issue that needs constant consideration because it focuses on people with all types of visibility who may access the web application. Alternative text was used for images on the website to allow users who may be blind to access information of an image. The ‘strong’ HTML element was also used to define text that needs to bold to allow user who also may be blind to understand when a text is bold though a screen reader.

## Consideration of Legal Issues

The Copyright Act (1988) provides five exclusive rights that protects copyright owners. This was a very import legal issue to practice in my project because it can have detrimental effects on the others. Images throughout the website are of my own and was not used or sourced from other people. To ensure users understand the legal side of the web application, they’re required to agree to the site terms and conditions before signing up.

## Consideration of Professional Issues

Professional practice was used by flowing methods and techniques professionals would use to achieve a successful project. This includes flowing a project methodology, using external liberties to make the development smoother, user testing, and commenting code so its clear for others to understand.

1. (Cederholm, 2010), (Keith, 2011), (Gasston, 2013) [↑](#footnote-ref-1)