jusRentIt

Project Report

Group 15 - Manpreet Bance, Alison Hoang, Tehmina Ahmed,
Ka Low, Hana Uddin

April 2019

Contents

List of Figures

List of Tables

Acknowledgements

The entire team would like to express our gratitude towards our supervisor, Dr. Basil Elmasri, for his continued support and guidance throughout the course of the entire software project, providing assistance from conception of the idea to project completion.

We would also like to thank the Department of Computing at Goldsmiths, University of London, for the opportunity to work on this project as a team.

Chapter 1

Introduction

jusRentIt is designed to be a platform based in London where businesses can lease their spare inventory (rather than keep it in a warehouse racking up storage costs), as well as the general public who can rent or lend anything ethical. Certain age demographics are more likely to use jusRentIt such as young people aged 18-25 and students. The platform can be used to rent any item, if it's safe and legal! This can vary from big items such as houses and helicopters to smaller items such as tools and toys.

According to our primary research; the group found that small businesses such as Kaspas Ltd were leasing their spare inventory privately in order to generate extra income. Kaspas is a dessert parlour branch who were offering their unused equipment such as waffle makers and crepe pans to family and friends for events and personal use. "Doing this helps us to clear storage for perishable stock and allows us to make a bit of extra money" – Tashir Gujjar (Manager of Kaspas).

Secondary research proved that Millennials (our primary target market) are preferring to rent items including houses, cars etc. as opposed to buying because of it's a short term, quicker and cheaper alternative in a city which is proving hard for young people to save and invest in. According to 'Simply Business', the average rent in London is £665.87, which is much cheaper than the average monthly mortgage repayment of £1,280 (1.4).

From this the group derived that our target market will be increasingly searching for rental accommodation – which will be one of the many facilities hosted on our web application. Our target market can use our platform to find rental accommodation as well as leasing the furnishings to go in it, the group aims to provide a multi-purpose site which is versatile to meeting the needs of its user. This is evidence to show how a platform like jusRentIt is key to filling a gap in the market.

1.1 Motivation

The inspiration for this concept came when one of the members of the group was looking to host an extravagant birthday event but didn't want to buy all the decorations and party gear; as it was only to be used once in a year. Therefore, the group did some research online to try and rent the equipment instead but was unable to find a sufficient resource such as a web application or app where the group member could rent a variety of products. The group member who faced this issue had to rent them separately from different platforms, and this was unnecessarily demanding, wasted time and was more costly due to extra delivery fees. This sparked the idea for an online platform which

allows users to rent a variety of different products from one site, with safety satisfaction and convenience.

1.2 Scope

jusRentIt aims to combat issues people may have with space, budget and time by allowing users to rent out items on demand for short term leasing. This means that potential users can access a 'try before you buy' scheme where people can test out products using this platform and then evaluate whether they wish to purchase the item or not. Alongside this, users may be able to save money by using a product for single use rather than having to buy it outright. For example, a bouncy castle.

1.3 Objectives

As a team we decided our main objective is to fill a need for a group of people i.e. a gap in the market. jusRentIt fills a need for a short-term leasing platform with a variety of products.

Especially for those within the age range of 18-45 (our target market) as our primary and secondary research indicates. Another Objective was that our web application should be user-friendly and simple considering our target market consists of a wide range of age demographic.

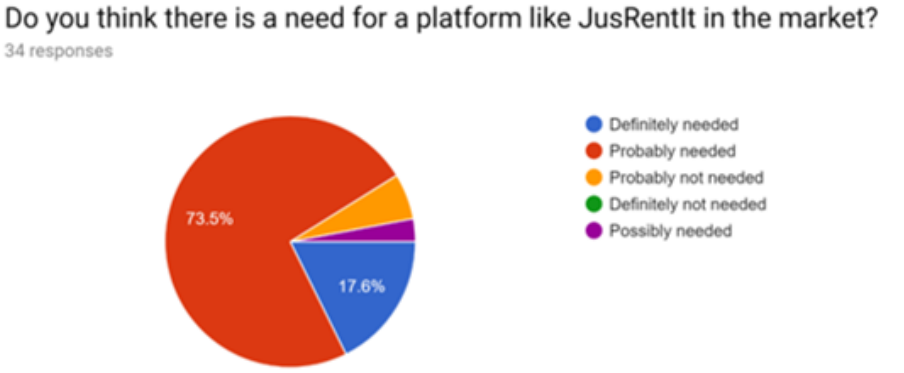


Figure 1.1: Chart: Need for jusRentIt

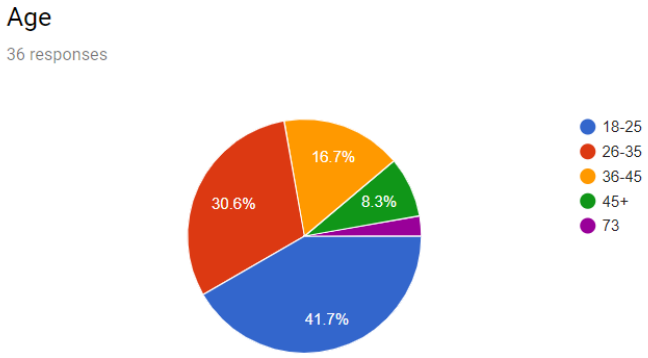


Figure 1.2: Chart: Age of Potential Users

Furthermore, the group also highly prioritises the safety of our 'renters' and 'lenders', therefore our group has devised a liability insurance and safety deposit policy which means that the products being leased should remain protected at the cost of the renter of the product via a safety deposit. The liability insurance is an online contract (terms and conditions) that specifies how the renter agrees to return the product in the condition it was collected, or their deposit will be deducted. Also, that jusRentIt doesn't take responsibility for agreements and transactions made with cash as they are untraceable.

To combat online payment fraud, our group also decided to incorporate PayPal as a method of safe and reliable payment. Our objectives vary according to the user of jusRentIt and the reason for they are using it, taking this into account, the group aimed to design jusRentIt in a way which it was flexible and accessible by all within our target market.

Chapter 2

Technical Specification

2.1 Background

The technical architecture of the web application is that of a layered architecture, this being due to the natural fit in the way the project is structured. The logic of a layered architecture allows the separation of the Front-End, Middleware and Back-End, however still allowing for cohesion between these, which will be detailed further.

The presentation layer refers to the front-end, displaying the web application in a format that the users are able to interact with - achieved using HTML, CSS, PHP, Ajax and JavaScript. The business layer processes the user's commands and confirms/denies them by making logical decisions - this being done using PHP script, also being able to request access to the database. The back-end consists of a database logic layer, where data is stored in a MySQL database.

2.2 Application Frameworks

A framework is an abstraction where software is written to provide generic function to achieve a task or tasks. This web application has made use of some frameworks to contribute to the functioning of it - mainly being in the Front-End.

2.3 Full Stack Development

As previously mentioned, the web application was designed and structured in a way conforming to the stack structure: Front-End, Middleware, and Back-End. This section will further discuss the specification of technologies in each tier of the stack and the reasons behind their uses:

2.3.1 Front End

HTML

HTML is the standard markup language used to create web pages. It is an abbreviation for Hyper Text Markup Language which is used in the design of a website defining the structure and these structures are differentiated and defined using HTML tags. These tags are then used by the browsers to render the webpage translating it into a readable form to users. As per its function, HTML was used in the web application to structure each of the pages of the web application. This decision to use this was due to its compatibility between devices and browsers in being able to display content correctly

and as designed - this being a criteria in the testing of the web application which needed to be met.

CSS

CSS are Cascading Style Sheets used to style components in a web page or multiples web pages simultaneously. This was used in the web application to style each of the pages which had the same or similar components creating a template making a more uniform and thus appealing look to users. As intended, we implemented CSS to create a uniform and professional look for the web application. We were able to achieve this by creating a CSS style sheets externally as opposed to internally or inline because of its planned use to style all components of each web page. This style sheet was then imported at the head of each HTML page to style the relevant components on that web page of the web application.

PHP

PHP - or Hypertext Preprocessor - is an open-source server-side scripting language which has HTML embedded, is used to make dynamic and interactive web pages. PHP was made use of in the web application in the scripting, more specifically, where the page was interacting with other components - i.e. accessing the database for Login purposes. The use of PHP was natural due to the fact of how it works well conjointly with the back-end which in this case was MySQL of which this combination make good. PHP script has been used to make connection with the MySQL database as well as further script in order to carry out different CRUD operations.

JavaScript

JavaScript is a scripting language used to program the behaviour of web pages and components of a web page. Similar to CSS, however, this focusses on behaviour in comparison to styling of a web page. Use of JavaScript in the web application was through being embedded in Ajax to create a more reactive web page.

Ajax

Ajax - asynchronous JavaScript and XML - is a web scripting language used in web pages to allow applications to send and receive data asynchronously. This being without the need for a user to have to refresh the webpage and interfering with the display and behaviour of the existing web page.The use of this in the web application are the login page transitioning into the registration page upon the user interaction through a mouse click rendering the registration form on top of the login page. Ajax proved to be beneficial through the fact that less resources are needed and used to render a webpage as elements are already loaded yet user interaction triggers another response - this is less time consuming as less traffic is created in the server in a request and consequently lesser bandwidth usage.

2.3.2 Middleware

phpMyAdmin

phpMyAdmin is a free open-source administration tool written using PHP, used to handle the administration to MySQL through an easy to use interface on a web browser. phpMyAdmin was used in the application to facilitate the connection between the PHP layer and database layer of our web application enabling a connection and access between

them. phpMyAdmin was used to create the tables and fields which would be used in the application helping lay out and structure the way data would be stored in the database and which data would be stored. This was very useful as the combination between PHP and MySQL is very natural and work hand in hand with each other making it easier to use and implement within the web application.

2.3.3 Back End

MySQL

MySQL is an open-source relational database management system which utilises SQL - Structured Query Language. MySQL was used to perform CRUD tasks including: Creating, Reading, Updating and Deleting data. MySQL was used as the database server containing the information which would be used on the web application available to both the users and admin for security purposes. The MySQL server was managed in terms of its administration by phpMyAdmin GUI, however, functions were limited here and the MySQL server was accessed through terminal allowing for more functionality.

Chapter 3

Project Management

3.1 Labour Management

Prior to commencement of the project, we were allocated resources in the form of the number of hours able to spend on the project until completion as well as meeting the completion date. This allocation was 1320 hours divided between the 5 members involved in the group resulting in a budget of 264 hours per person. Following the gathering of the resources available to the project, a timeline was set with provisional dates of when certain tasks should be completed by referred to as milestones. The following table breaks down the areas which were agreed between group members to lead during the development and implementation phases of the project:

No.	**Team Member**	**Development Area**	**Reasons for Decision**	**Objectives**
1	Alison Hoang	Front End Development	Familiar with Front-End development, strong knowledge of front-end languages	Login, Registration, Search
2	Hana Uddin	Testing, Middleware	Understanding of the testing process following initial market research and gathering user requirements	PHP Script, MySQL Queries
3	Ka Low	Front End Development	Strong design background, understanding of front-end development languages and design features	Product, User Account, Dashboard
4	Manpreet Bance	Project Management, Back End Development	Understanding of databases, prior experience with SQL and MySQL, Strong leadership and management background	MySQL Database, MySQL Queries
5	Tehmina Ahmed	Research	Understanding of user requirements and customer-facing role, confident with asking users questions to obtain valuable information	Research, User Requirements

Table 3.1: Labour Management

Taking this approach with regard to the allocation of labour means we are less susceptible to certain interruptions including:

1. Overlap of tasks/Unfair workload - As the tasks were divided between each of the group members, this made it easier to track what needed to be done and by who preventing overlap as well as any one member having to take on a greater workload as a result. The front-end was the bulk of the development phase and therefore had two members carrying this out, rotating and sharing developments and offering help to each other. The tasks were divided so that no members were having to wait for another member to complete a task before moving forward which would have caused potentially severe delays in timescale. The project manager was continuously on hand to offer any support needed as he had the knowledge to support both back end and front end development in the event of any issues but this was not necessary - as well as this, this made sure that there was communication and updates between both the areas.
2. Communication issues - As will be discussed further below, the strong communication between the team allowed for quicker response with any issues which had arisen and therefore a faster resolution. All group members were in constant contact with each other which allowed for the group as a whole to be more aware of what others were doing at what stages they were to affirm the pace they should be working at. Having poor communication would have meant that the group members would be oblivious to their counterparts progress and what they could be doing to support others, if needed.

3.2 Methodology

3.2.1 Agile Development Method

The Agile Project Management Methodology was utilised where, by definition, an iterative process made use of at each stage of development with close collaboration between group members in all areas of the development stack requiring constant effective communication. Using the Agile method meant that the project was dealt with the most effiencient way possible with larger tasks being broken down and divided among those who were most suited to completing these tasks effectively.

3.2.2 Scrum

Scrum is a framework that is used to implement the Agile development method whereby the project can be managed more efficiently in terms of meeting an allocated budget and a deadline or timescale - in this case, being the amount of hours we had allocated to each member of the group and the deadline this had to be completed by. The scrum was broken down into the following roles: Product Owner where quite simply all members of the group took on this role in the sense that accountability was taken for each component of the web application which was led by each member. Additionally, there was a Scrum Master led mainly by Tehmina Ahmed where she led and took responsibility for making sure all processes were being completed as set out prior to commencement of the task.

As part of the Scrum method, we implemented 'Daily Stand-ups' in the form of scheduled digital meetings at the beginning and end of each working day to confirm what would be done that day and whether any support was needed and if further resources needed to be spent in a particular area.

3.2.3 Kanban

We will also make use of the Kanban method where we can collaboratively monitor the work completed by each other and add any forthcoming tasks and plan how to go about

doing these - we use Trello to achieve this.

3.3 Version Control

3.3.1 Git

We are also making use of version control through Git to be able to track changes and again be able to work together more collaboratively and take responsibility of tasks which are being completed by each group member.

Chapter 4

Analysis

4.1 System Requirements

Below are listed our non-functional and functional requirements for our software. Our functional requirements consist of vital elements that our web application must have in order to fulfill its purpose. However, our non-functional requirements are a set of standards that our web application can be evaluated against.

4.1.1 Functional requirements

The table below contains the requirements the group created – some of these are vital requirements and had to be met successfully in order for the web application to work. Whereas, other requirements are optional and were created to enhance the user experience and functionality of the web application.

ID	Description	Vital/Optional Requirement Type
FR-01	Users must be able to register (sign up)	Vital
FR-02	Users must be able to sign in	Vital
FR-03	User's details must be stored securely in a database	Vital
FR-04	Website must show tab of different categories and sub categories	Vital
FR-05	User must be able to add products once registered and upload images	Vital
FR-06	User should be able to rate products using star system	Optional
FR-07	User should be able to write a description for the product	Optional
FR-08	User should be able to edit/alter the product's details	Optional
FR-09	Website should be able to show users suggestions based on recent searches	Optional
FR-10	Search bar/tab should show relevant products according to user searches	Vital
FR-11	User must be able to add products/items to rent cart	Vital
FR-12	User must be able to remove a product from rent cart	Vital
FR-13	User must be able to securely pay and checkout	Vital
FR-14	Payment details must be stored securely in databases	Vital
FR-15	User must tick textbox containing liability insurance and safety deposit policy before leasing product	Vital
FR-16	User must be able to communicate with other users via messaging on jusRentit	Vital
FR-17	Lender must list whether item is to be delivered or collected	Vital

Table 4.1: Functional Requirements

4.1.2 Non-functional requirements

These are standards which our web application must/should meet. Some of these requirements are vital and necessary whereas others are optional. Our final prototype reviews suggested to us the types of non-functional requirements that our team can measure against to enhance our site; which are listed in the table below.

ID	Description	Vital/Optional Requirement Type
NFR-01	The site should be simple and straightforward to use; in terms of design and function – to satisfy our wide target market audience	Vital
NFR-02	The site should be aesthetically appealing	Optional
NFR-03	The search suggestions (if created) should be useful and relevant	Optional
NFR-04	Sub categories of products should be inclusive of any and every possible, ethical product	Vital
NFR-05	User profiles should list amount and quality of deals done in the past	Optional
NFR-06	The website must fit all screen sizes	Vital
NFR-07	The website should provide user with useful feedback messages	Vital

Table 4.2: Non-Functional Requirements

4.2 Stakeholders

Our target market is (18-45+) – these are the individuals our group targeted in regard to designing the web application around their needs and understanding, because according to our survey our group found that this age range had the highest interest in the app combined (Fig 1.2) and would be most likely to use it as well as recommending it to their family and friends (Fig 1.1). Gumtree, Rent-a-Car and Shpock are some of our competitors.

However, our web application differentiates from theirs in the sense that jusRentIt offers a variety of products to lease rather than specific products or services, i.e. Gumtree leases property such as apartments and houses only. Shpock has a similar function and style to our web application but it's a platform for trading rather than leasing a product and it is only available as an iOS/Android app; making them an indirect competitor. Small business owners are also stakeholders, because as users of jusRentIt they can offer products or spare inventory for lease in order to boost profits by creating this extra avenue of income.

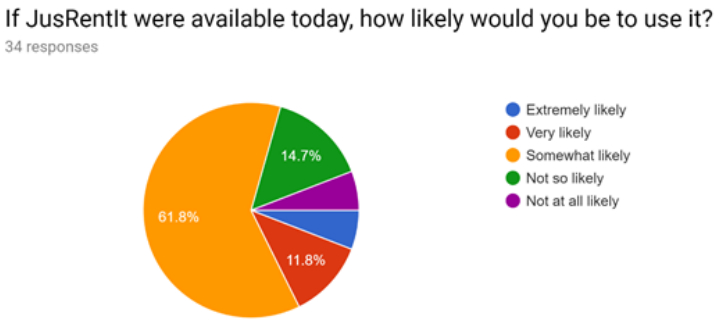


Figure 4.1: Chart: jusRentIt Likely Usage

4.3 Use Case

is a graphic display of intercommunication between users or parts of a system which shed light on the requirements of the system and the role each component plays in order to achieve a successful product/outcome.

4.3.1 Use Case Diagram

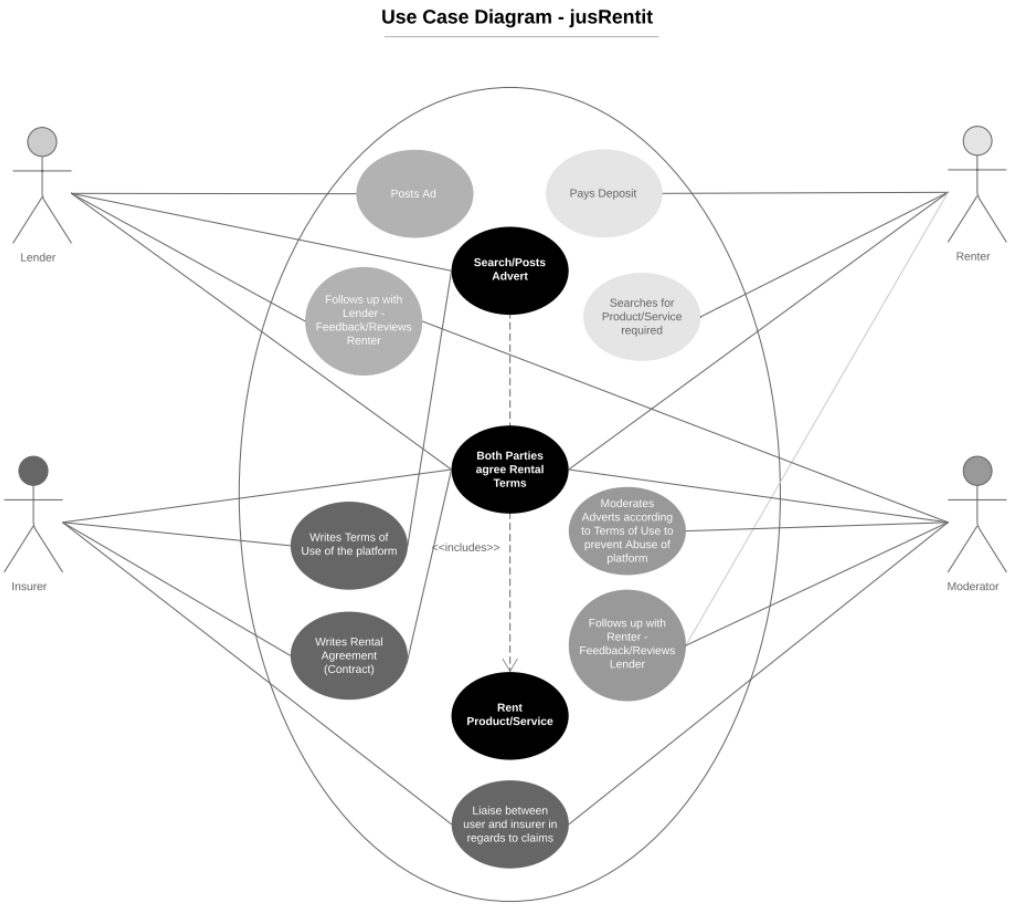


Figure 4.2: Use Case Diagram

4.3.2 Use Case Scenarios

Use Case Scenario	**Rent an Item**
Outcome to be Acheived	Renting an item from a Lender
Cause of Post Condition	Website User
Prerequisite	User has registered an account and has selected an item to rent
Successful Scenario Steps	1. User creates account or signs in 2. Website displays home page 3. User searches website or selects item displayed on homepage 4. User clicks button 'add to cart' 5. User views cart, then continues to browse/proceeds to payment 6. User fills in personal details, name, etc 7. Users mutually agree whether item is to be delivered or collected, and how long it is to be leased for 8. If users agree, then deal goes ahead 9. User pays safety deposit 10. User agrees and ticks liability insurance checkbox 11. User pays amount listed by lender
Post Condition	Item successfully rented, safety deposit paid, and liability insurance agreed

Table 4.3: Use Case Scenario: Rent an Item

Use Case Scenario	**Lend an Item - Posting a Product Advert**
Outcome to be Acheived	Lending an item to a Renter
Cause of Post Condition	Website User
Prerequisite	User has registered an account and uploaded images and description of item to lend
Successful Scenario Steps	1. User creates account or signs in 2. Website displays home page 3. User clicks 'upload' button 4. User uploads images and writes description of item 5. User states whether item is to be delivered or collected 6. User sets price and safety deposit 7. User sets approx. time item is preferred to be rented for 8. User saves changes so that other users may be able to see item information 9. User waits for offers
Post Condition	Item successfully leased, safety deposit received, and liability insurance agreed and item delivered or collected

Table 4.4: Use Case Scenario: Lend an Item

Use Case Scenario	**Post a review with star rating system**
Outcome to be Acheived	Post a review about product, lender/renter Rate 1-5 Stars (Worst to Best) - Product
Cause of Post Condition	Website User
Prerequisite	User has registered an account, rented out or leased a product to/from other users
Successful Scenario Steps	1. User signs in 2. User searches and selects item or profile to review 3. User writes review for renter/lender regarding quality and genuineness of deal 4. User rates product from 1-5 (Worst to Best) 5. User submits review so others can see and make an informed decision
Post Condition	Item previously rented, reviewed and rated

Table 4.5: Use Case Scenario: User/Product Review

Use Case Scenario	**User Registration**
Outcome to be Acheived	User registers on website using personal details and creates username and password
Cause of Post Condition	Website User
Prerequisite	User visits the jusRentIt website and clicks 'sign up' button
Successful Scenario Steps	1. User visits jusRentIt website 2. User selects sign up option 3. User fills personal details form 4. User selects option to receive subscription emails 5. User saves changes, details are stored in database 6. User browses website or logs out
Post Condition	User login successfully created

Table 4.6: Use Case Scenario: User Registration

Chapter 5

Design

5.1 Design Models

The design of jusRentIt follows an e-Commerce (C2C) Consumer-to-Consumer business model where there would be an interaction between two consumers - in this case it would be a renter and the lender of the product. The interaction consists of transaction stage and the confirmation of activity. While browsing on the application, the "log-in" would be optional as a Guest option would be available. The design was intended to be as user-friendly and simplistic as it can be to fit all the stakeholders and other potential users. It would also allow users to customise the service to fit their requirements in both sections (lending and renting) in ways such as recommended products and saved products for later transactions.

While using the C2C business model, the project would to create a web application where it would be able to use on multiple devices such as desktops and mobile phones but primarily as a website rather than a mobile application. This would be achieved by using HTML as the front-end of the application with CSS styling bearing in mind to keep it as simplistic as it can be. Ajax would be used to load documents which requests or retrieves the data from a server which would be especially useful for pages such as the log-in page. This means that the application would be more reactive. Prototypes were used to plan the design before the implementation of the web application such as paper prototypes and a digital interactive prototype.

The paper prototype provided a simple visualisation of the upcoming web application and gave ways to help improve functionality wise. Implementations of the recommendation products and the strategic "top-left" logo was some of the ideas developed later on. The notion of the top-left logo was statistically proven that users were more likely to remember the logo 89 percent more than anywhere else on the page.

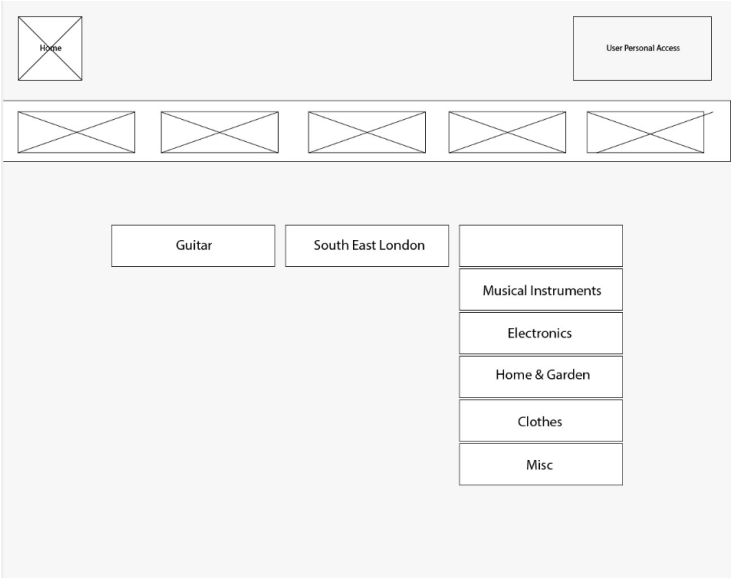


Figure 5.1: Paper Prototype 1

An interactive digital prototype was later created for a more higher-quality version of the previous prototype. This would be the prototype where we would use to validate the user experience before the release of the completed service. This was achieved using Adobe XD where it would simulate the interactions of the user's clicks, allowing transitions to different sections of the web application.

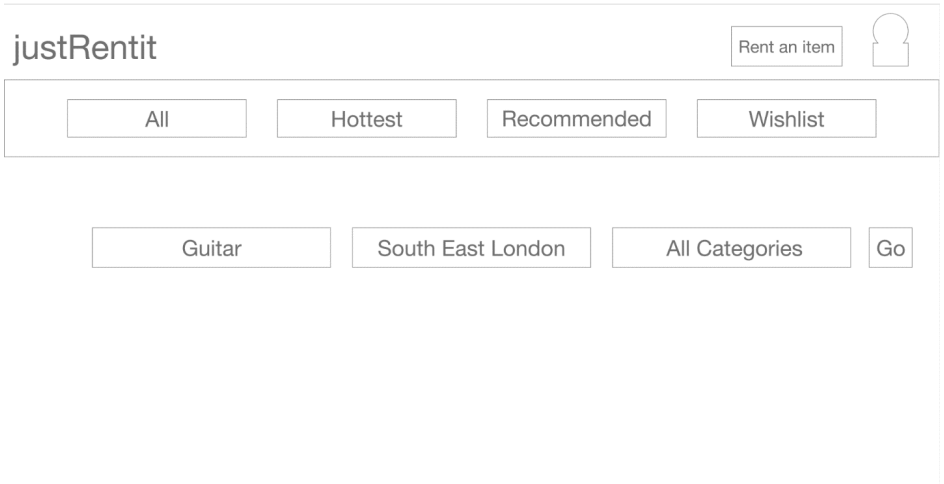


Figure 5.2: Paper Prototype 2

5.1.1 Activity Diagram

We planned out the activities in the C2C model using an activity diagram which shows the not only the user's process requests but the systems process requests also. This illustrates the workflow of the application between the user and the system.

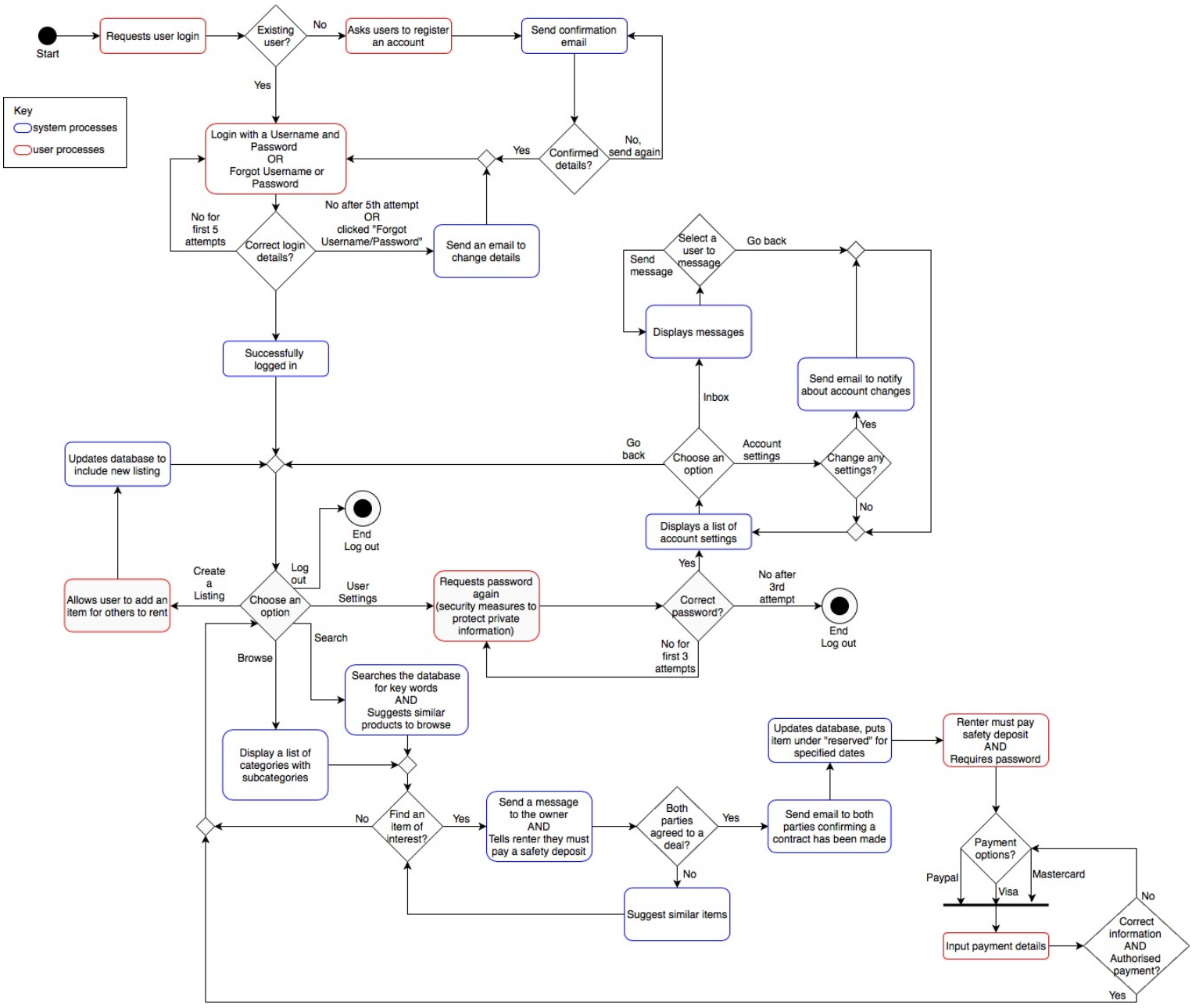


Figure 5.3: Activity Diagram

5.1.2 Sequence Diagram

The Sequence Diagram shows the process of the interactions between the two consumers in sequential order. This would include the browsing of the application, the communication of the two consumers and also the transactions of the two consumers. The sequence diagram can be seen in Figure 9.1 where there is a clear, simplified version of what has been expected of the web application in the form of a Minimum Viable Release, or MVR.

5.2 Database Design

The Back-End of jusRentIt would consist of a database where we would MySQL linking the front and back-end of the application using JavaScript and PHP. Using MySQL, there would be many tables that would store different types of data such as User ID, addresses, listings and other personal details. Much of the data would be unique such as the User ID, password and product listing. The database would have be secure due to the personal information inputted in the database such as addresses, passwords, first names and last names. Such data that are unique would be Primary/Foreign keys in the database to uniquely identify these records in the tables. This makes it possible to search for listings/products while consumers/users are on the website using keywords or filters.

The database also stores ratings on the user that is lending out their product where it would store the input of how the user felt about the product they were renting out. This would be presented as stars on the front-end but on the back-end (database) it would be a value up to 5. Filters would use Boolean operators (AND/OR) to narrow down the list of results to find the desired or specified product. The filters would be changed to a case to case basis depending on what the initial search was, in the case of Figure 4 – due to the search being "Guitar", the filters are accommodated to what the users are likely going filter out while distance filters would stay there regardless of what has been searched.

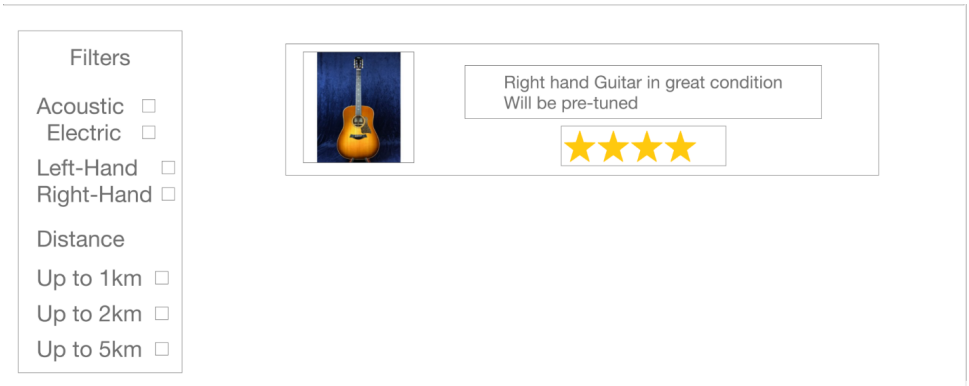


Figure 5.4: Interactive Prototype: Rating

5.3 UI Design

The UI Design was specifically designed in a way where it would be completely simplistic, thus being able to use by all users. Referring to figure 5.1, it would show you the complete bare bones of the design of the jusRentIt web application. It allows the user to be able to navigate straight away without any confusion. The main point of the design was to allow the user to search their desired item, it is placed in the centre so it is the first thing the user sees when they enter the web application. It is also slightly bigger than anything else so the user's eye and attention is towards the search bar. Navigation bar is placed above to explore each section of the jusRentIt while the content is shown below. It is planned that the navigation and the top section of the application would be reused throughout the different sections that this application has to offer and would be fixed at the top. This is to make the accessibility of the website easy for the users to go back and forth and is a quality of life feature. The user profile, renting button and the

logo makes it easier to go back to the homepage, put a new listing or edit their personal details. While discussing about colours, we found that grey would be great as it's not as plain as white but not as hard on the eyes as some other colours such as purple or yellow.

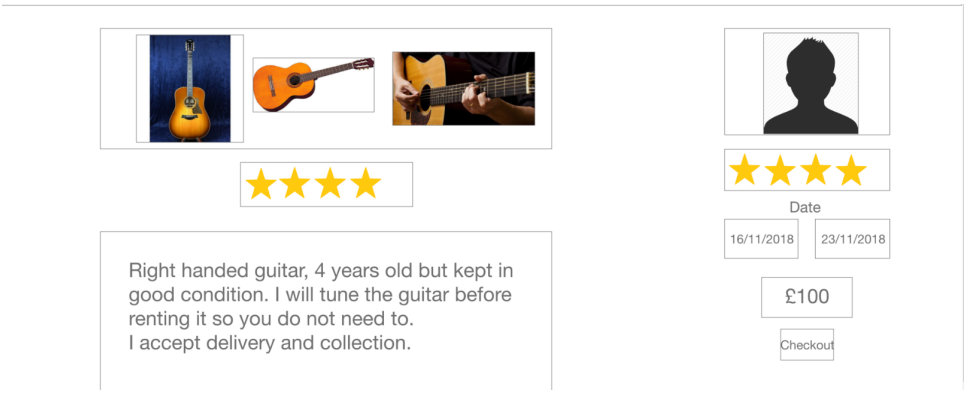


Figure 5.5: Interactive Prototype: Listing

Once clicked on an existing listing, the UI would provide an easy to read structure to the user. As shown in figure 5, the user that submitted this listing would be on the right with their rating and fees for the item (deposit) with the option of choosing the dates to rent. There would be also a rating for the item itself which is shown below the images of the item and description of the item. This covers majority of the page as it is the main point of this section.

The UI of the check-out section is kept simple and as clear as possible to ensure that the user has no problems doing transactions on the application. The numbers are to be made large and clear to show that the pricing is transparent with no hidden costs.

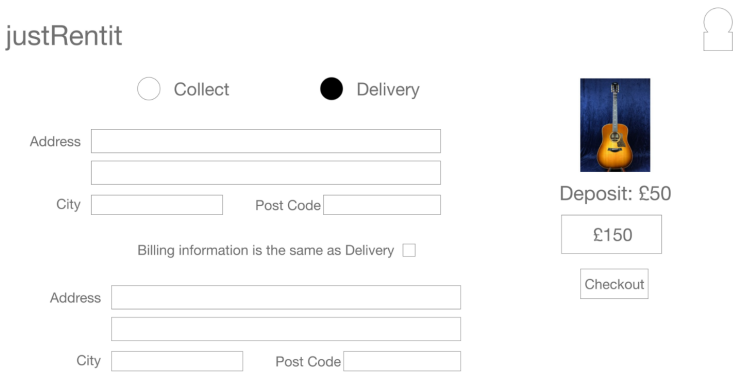


Figure 5.6: Interactive Prototype: Checkout

The user profile consists of 2 different navigation bars, the static navigation bar at the top and the personalisation navigation bar for the user. This is the section where users can change their personal details such as passwords, email addresses and payment details. It is also designed to list out the order transactions and list out the listings that the user has submitted.

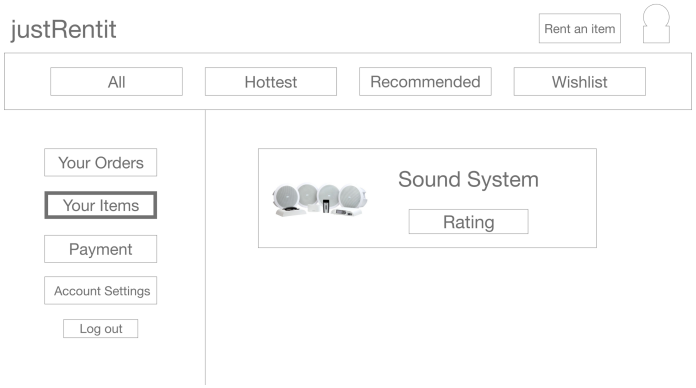


Figure 5.7: Interactive Prototype: User Profile

The Listing section follows the simplistic concept to keep everything consistent, this is where they can put their product images and description as well as their fees or deposits.

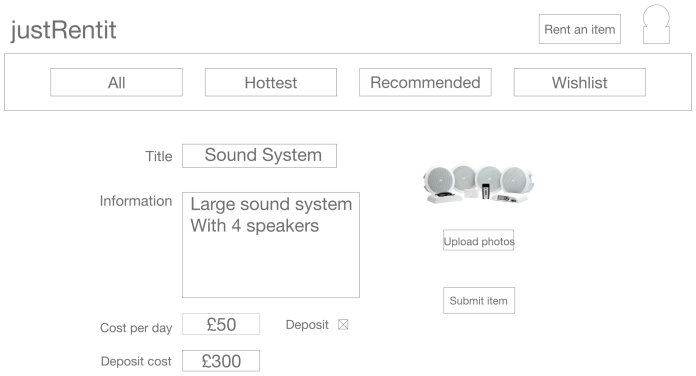


Figure 5.8: Interactive Prototype: Create Listing

Chapter 6

Implementation

6.1 Front End

Front end web development, also known as "client side programming", is based on everything that happens in the user's browser. From the layout, user interface and user experience, it is what the users can see and interact with. Since the project heavily focuses on the user base, it was essential to us that the group made existing and potential clients feel comfortable with using the web application. The group established early on in the project that a website based web application would be best as it is the most easily accessible via multiple devices such as smart phones, tablets, laptops etc.

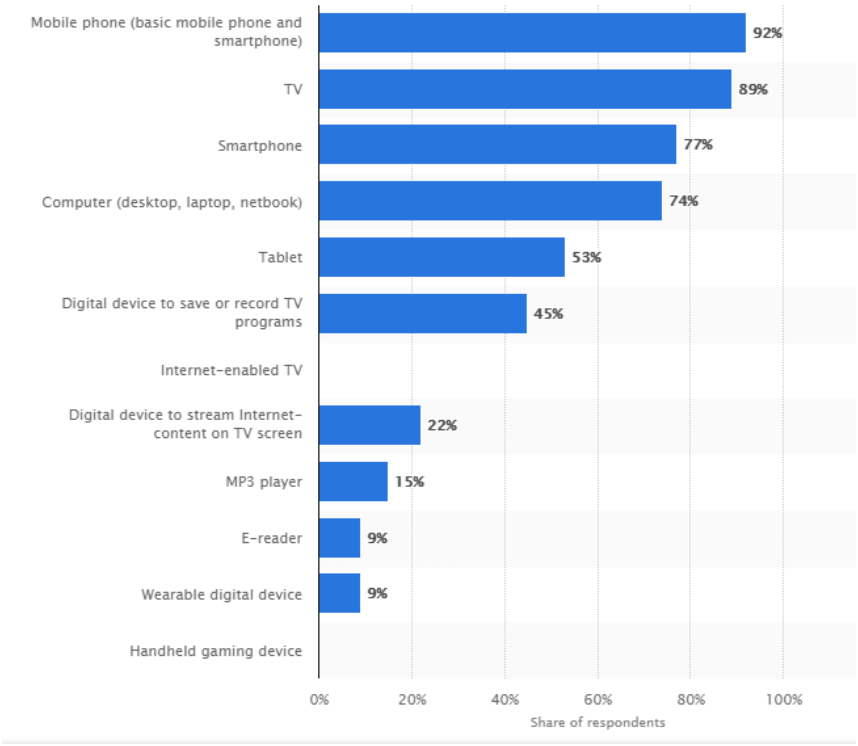


Figure 6.1: Statistics of Electronic Device usage in the United Kingdom in 2017

This meant that the group had to ensure that the website was responsive. There is an

example of this via the login page that is shown below. The page should rescale/adjust its size depending on the size of the window to allow every element to be viewed clearly.

Figure 6.2: Responsive: Login Page

The login screen of the website was created first due to it being the first thing the users would see. Quite a lot of thought went into it as the group wanted to make sure the web application would leave a lasting impression. It uses form validation and business logic to obtain the user account from the database (line 15 of figure 6-3). Throughout the web app, the group wanted to ensure that the website seemed similar to other online sites and have a familiar feel so that the users would not get overwhelmed and confused with the different elements of the website. So, the group included important aspects such as a username and password field and a link as an option to sign up.

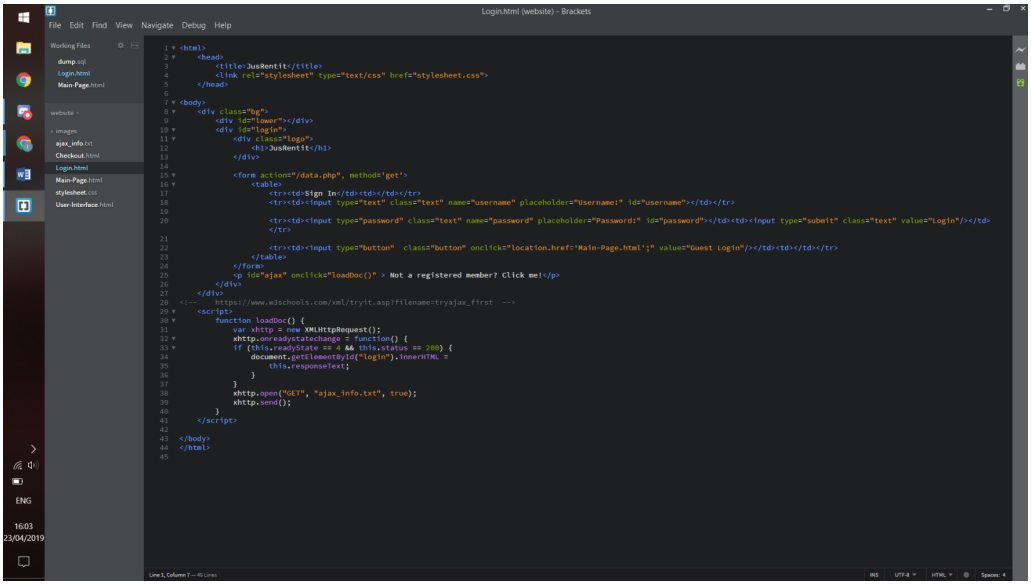


Figure 6.3: Code: Login Page

Something the group thought was an important feature for the website was to implement a guest login button. This allowed users to view the website without being a

registered member, in case they wanted to browse through the catalogue. The group thought this was important because jusRentIt does not have a big reputation and may encourage new users to view the website. The guest login feature would have restrictions compared to a registered user such as a reduced range to view products and inability to rent out a product for safety reasons.

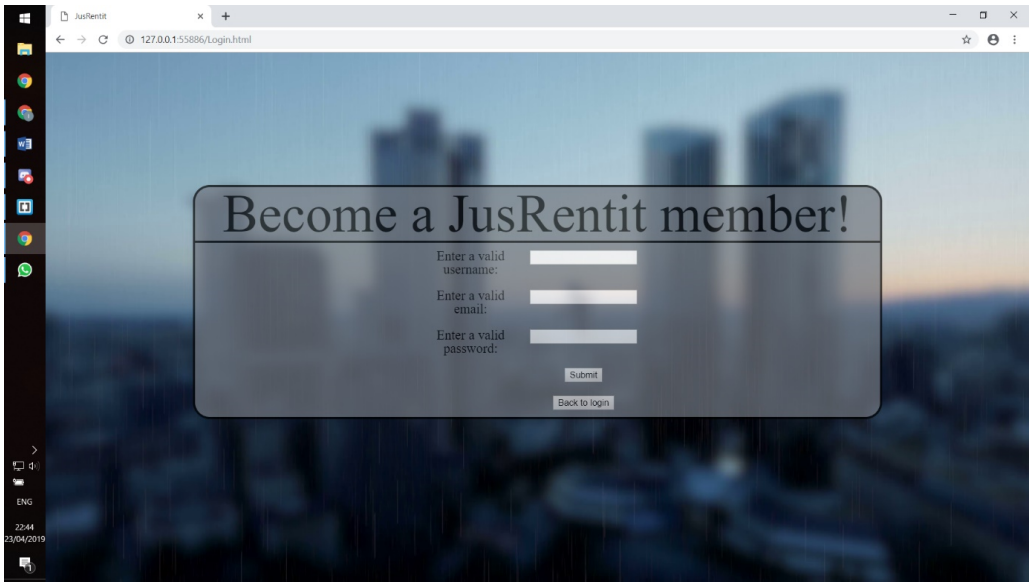


Figure 6.4: Ajax: Signup Page

The group implemented Ajax (line 25 of figure 6-3) into the login screen in order to make the user experience more enjoyable. Ajax works by retrieving server data for specific elements without reloading the whole page, in the case it loads up a text file within the login info page and does not have to reload the background image and logo etc which allowed users to "move" to the sign-up page without having to load a new webpage/url.

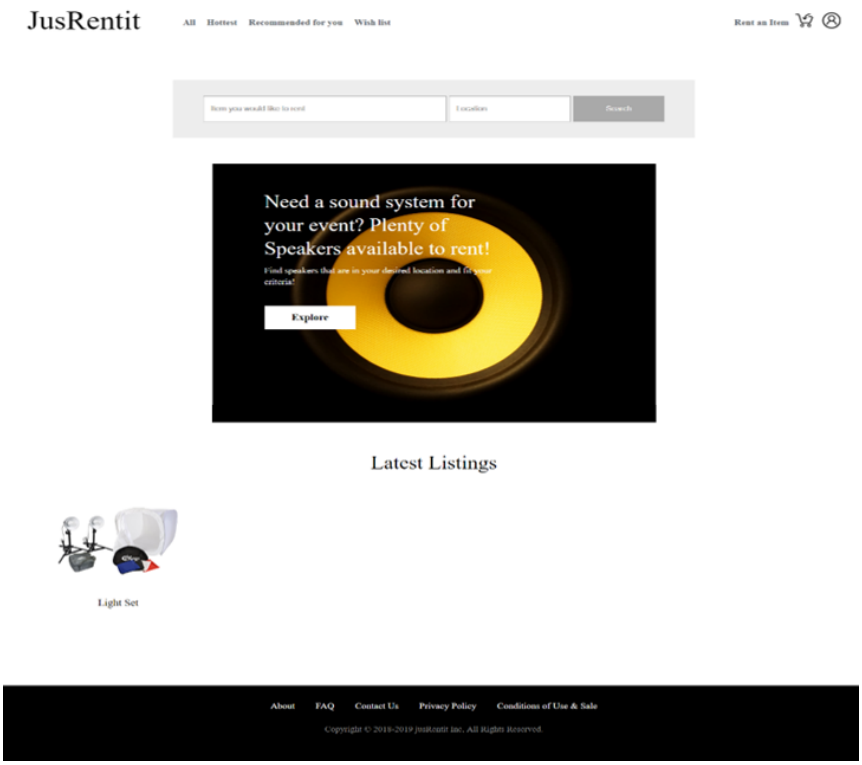


Figure 6.5: Main Page

The main page is where the users are able to search for different items available to rent. This page should be different for example, if a user wanted to put up an advert to lend an item or if the user wanted to search and borrow an item. The group aimed to make the page appear clean and easy to use. Users would be able to click on the logo located in the top right to return to this page and it should contain links that allow the users to navigate throughout the website. Such as looking at different listings, heading to checkout or user settings.

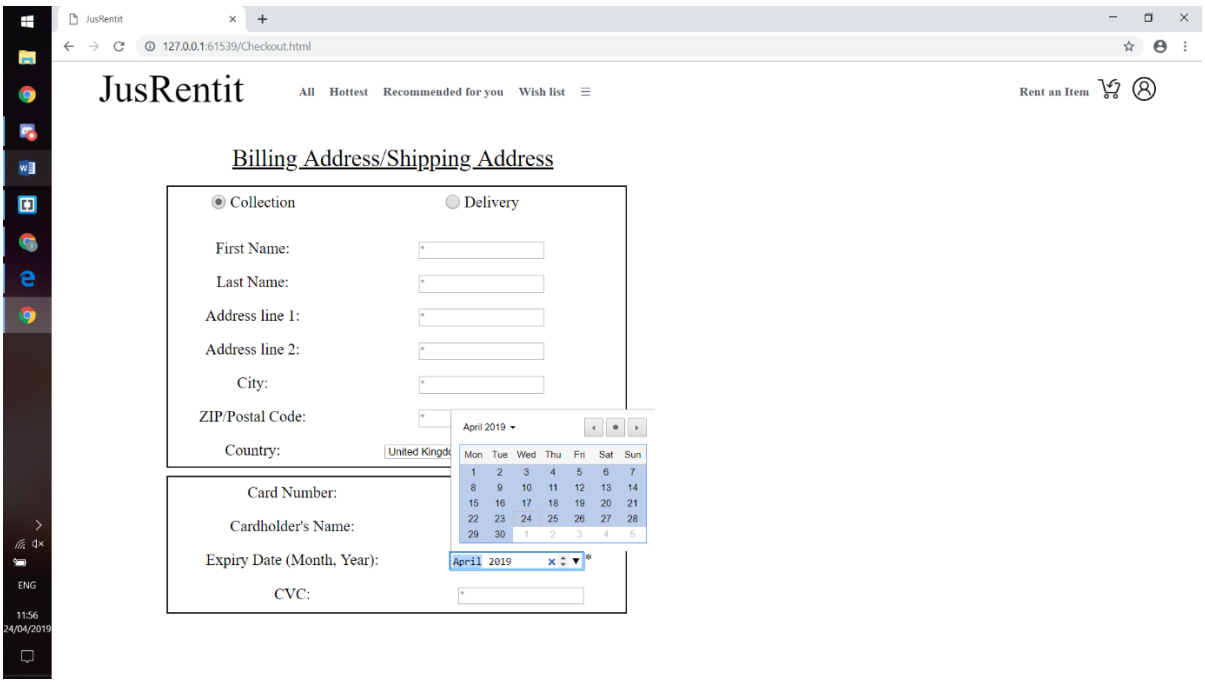


Figure 6.6: Checkout Page

The checkout page as shown above, contains the standard contact details with the use of a form, it also includes features such as a radio button in case users decide to choose whether they would like collection or delivery, a drop down with a list of every country in the world and also a calendar for their card expiry date(they are able to type the date if they so choose). All of these were implemented with the idea of making the user experience feel more enjoyable and interactive.

6.2 Back End

The database consists of two key components of data. Private data which includes user login details and payment details and public data such as the different listings of items available to rent and the tags they would appear under when searching the database.

6.3 Website Development

Throughout the process of making the website, the group was initially split into smaller groups, those working on the front end and those working on the back end. This is because each group would need to use different languages and having teammates who understand and can help debug each other's code was crucial. Twice a week, members of the group would meet up to swap code and test one another's code to help decide which parts needed to be improved or fixed before user testing. Next time it would be important for the different groups to swap between front end and back end in order to help understand the project more and further develop their skills.

6.4 Disruptions and Solutions

The majority of the problems were due to learning various new languages and techniques that were needed for the website. This was mostly learning how to use Ajax, which one of the lab assistants said is very important in web development and a good skill to learn so we really wanted to try and implement it. Also, PHP that was used to connect the front end and back end together. We needed to spend extra time researching about what they were used for and how to implement the code to work with the website.

Another problem came across was that the Back End server was being hosted on one computer. This made it difficult to integrate the website together as all of the data was only being stored on one computer, anyone who wanted to interact with the website and add data would not be transferred to the group's server. This disruption was not solved in the end but the plan would be to host it on the university servers. This meant that our website would always be connected unless there was a server maintenance but also that different devices are able to connect to our website and database.

There were also a few issues among the group regarding the use of GitLab. The project manager spent some time setting up a dedicated git project for the whole group to use but not everyone knew how to use it. There were a couple of issues with frequent commits and uses and occasional merge conflicts but slowly overtime and practise, everyone became accustomed to it with the help of one another.

Chapter 7

Testing

7.1 Testing Criteria

Before conducting the testing, different functionality was to be tested. This included the checking of the Renter and Lender features of the site. The first 6 tests were testing out the renting features and the other 6 were testing the lending features.

The following tables are test cases that are assessing the overall application. Some of the test cases produced are repetitive but that is to ensure everything is working and is fairly tested.

Test	**1**
Content	Responsive buttons, logging in and whether featured items appear and can be easily rented.
Input	The user enters the required information for login and goes to the main page and checks all featured items appear that can be potentially rented.
Pass Criteria	Application is responding to the user when pressing buttons, logging in, etc featured items appearing which can be easily rented.

Test	2
Content	The user can sign up to the website. Renting out items with the new account that has been made.
Input	Signed up. Explore and checking responsive webpages to be able to rent goods.
Pass Criteria	User is able to sign up. Products can be rented out.

Test	3
Content	searching up goods to be able to rent them out with no issues.
Input	The user explored the page, searches up items. Checked it can be rented out.
Pass Criteria	Searching up items that give a related product that can be rented out. Useful suggestions.

Test	4
Content	Checking the guest login is working. Exploring the pages as normal.
Input	Logging in as a guest, exploring pages.
Pass Criteria	Logging in as a guest while all the core functions are working.

Test	5
Content	Checking whether the featured items are showing up on the main page
Input	User exploring the main page, selecting featured items.
Pass Criteria	Selecting featured items that can be rented out.

Test	6
Content	Checking if the site is responsive.
Input	Selecting all the buttons, searching up and renting goods.
Pass Criteria	Everything generally responsive.

Test	7
Content	Signing up to the site and lending out products.
Input	User signs up and sends out a product of their choice.
Pass Criteria	Signing up and lending products out were successful.

Test	9
Content	The button features working such as search up and navigation bar.
Input	Search up items and use the navigation bar to check for responsive web pages.
Pass Criteria	Searching up items and checking if the navigation bar works successfully.

Test	10
Content	Testing emails to be registered when signing up to be a lender.
Input	Signing up and lending out a product.
Pass Criteria	Signing up with the ability to lend out products.

Test	11
Content	Testing the billing information input to see if it works.
Input	The user rents out a product and check if billing information is appearing correctly.
Pass Criteria	Billing information appears when checkout is completed.

Test	12
Content	Testing the use of similarly spelled suggestions in the search bar.
Input	Searching up the wrong spelling of the word speakers to check if any suggestions show up.
Pass Criteria	Spelling suggestions are shown when the user inputs the wrong spelling.

7.2 Testing Results

This was the result of the test cases that showed whether it was a success when conducting different features of the site.

Test	Result
1	Pass
2	Fail
3	Fail
4	Pass
5	Fail
6	Pass
7	Fail
8	Pass
9	Pass
10	Fail
11	Pass
12	Fail

From these results, it is clear that there are improvements that are needed which includes fixing the process of the email so users can sign up, featured items, posting items and spell suggestion due to the non-functionality of the website caused by the issues faced as mentioned regarding the failure to successfully link the Front End with the Back End database.

7.3 Formative Evaluation

As a prerequisite to the design and creation of jusRentit, we made a survey to measure our website against in order to judge its level of success to see whether it meets the needs of our potential customers in terms of design and functionality. With the survey, we selected people supposedly within our target market to test our website. From Figure 1.2, we determined the age group of the website so, with this information the testing was done with two participate types; the business owners and regular (any occupation) that have the age range of 18 to 45+. In doing this, we discovered how we can accommodate different age ranges and discover whether any changes are needed.

The testing was done on two different days which was spilt into two groups. The groups were separated into groups where they test out vital features of the web based application which are renter and lender feature as well as other functional requirements. Test number 1 – 6 was grouped on February 14th; testing out the renting features whereas test number 7 – 12 was grouped on February 15th was testing out the lending features. Having two separate occasions testing the main features allowed us to quickly

identify any faulty issues that occurred when using renter/lender features which are vital requirements for the website. After the testing of the website, user feedbacks survey was done to identify how improvements could be made. The group has a survey was asking for user feedback as it will be used to quickly identify any improvements that are needed.

Section 7.2 showed the testing result. The result is not as positive as anticipated as there were six failures displayed within the site, this clearly affected the user feedback where 7 out of 12 said it was a user-friendly interface, this shows the website is not catered well enough. The problem that has occurred with the testing users were signing up, featured items, posting goods and spell suggestion. With these issues arising, we have found the problem area that is needed to be fixed. The user feedbacks survey (Fig 9.7 - 9.10) showed a correlation of the testing result and user feedbacks, this present that the vital requirements of the website was not implemented as it should have been.

Analysing the response, many improvements could be as a lot of different suggestions were given to make from the users, from that it shows the website needed to add more unique features to increase user experience. The user testing suggested that improvements are needed, such as adding more featured items and adding more to the website in terms of design.
Basing off the user feedbacks and the testing result displays that the website is an unfinished product that extra time was needed in order to achieve the initial goal and gain better execution of the overall application.

7.4 Functional Requirements Evaluation

To be able to meet the requirements of the applications. Testing of the application was done in order to ensure the usability of the website. These testing are the main functionality was defined form the section 4.1.1. The testing for the application was done which can be seen in section 7.1. The tables presenting is the evaluation of the functionality of the application.

Functional requirement ID	**FR-01**
Test Description	The application allowing the user to register to the website.
Input	User registers by filling in all the necessary information (name, email, and password) for signing up the website.
Pass Criteria	After signing up confirmation message pops up which is then directing the user to the main page.

Functional requirement ID	**FR-02**
Test Description	The application allowing the user to sign in
Input	Signing in to the web application
Pass Criteria	The user is able to sign in while using the site normally

Functional requirement ID	FR-03
Test Description	When user is signing up to the web application it must be able to be stored securely within the database
Input	User enters billing information
Pass Criteria	The sensitive information is stored in the database

Functional requirement ID	FR-04
Test Description	Website must show different categories that shows has a sub categories
Input	User looks through the featured items and navigation bar
Pass Criteria	Different categories are showing

Functional requirement ID	FR-05
Test Description	User adding product within their account and uploads an images
Input	Registers to the web application and upload a product with images included
Pass Criteria	Uploading a product that displays the right image

Functional requirement ID	FR-06
Test Description	Rating system, so user testing out if products are able to be rated
Input	User selects an item that can be rated and checks if the star system is working
Pass Criteria	User was able to rate a product

Functional requirement ID	FR-07
Test Description	User testing out if they can write a description of the goods
Input	Post a product that allows the user to write a description of the product
Pass Criteria	Description of the product was shown

Functional requirement ID	FR-08
Test Description	User should be able to edit product after posting
Input	After posting a product the user will check if goods are able to be edited after posting
Pass Criteria	The posted product was able to be edited

Functional requirement ID	**FR-09**
Test Description	The web application should show suggestion of the goods depending of the search history
Input	User explores until suggestion comes up
Pass Criteria	Suggestions of product appears

Functional requirement ID	**FR-010**
Test Description	Search bar/tab should show relevant products according to user searches
Input	User explores the page then uses the search bar to check if any suggestion has be given
Pass Criteria	The search bar shows suggestions

Functional requirement ID	**FR-011**
Test Description	User must be able to add products/items within the shopping cart
Input	The user will pick a product that they choose and checks if it is displayed on the cart
Pass Criteria	The product is displayed within the shopping cart

Functional requirement ID	**FR-012**
Test Description	User must be able to edit within the shopping cart which is removing products
Input	Goes through the shopping carts and removes an item
Pass Criteria	Items gets removed

Functional requirement ID	**FR-013**
Test Description	The user must be able to go through checkout payment securely
Input	Rents an items and inputs private details
Pass Criteria	Payment details pops up when checking out the chosen goods

Functional requirement ID	**FR-014**
Test Description	Payment details must be stored securely within the database
Input	Rents an item out
Pass Criteria	When renting an item and going through the payment process, the database must be secured

Functional requirement ID	FR-015
Test Description	The user ticking the textbox that contains liability insurance and safety deposit policy before leasing product
Input	User lends out a product
Pass Criteria	The textbox appears and user was able to tick it off

Functional requirement ID	FR-016
Test Description	User able to communicate with other users with the web application
Input	The user communicate with other users by sending a message within the site
Pass Criteria	The message was successfully sent

Functional requirement ID	FR-017
Test Description	The lender users must be able to describe whether the product is for delivery or for collection
Input	User lends out a product and describes the collection method
Pass Criteria	The product had the collection method shown.

7.5 Non-Functional Requirements Evaluation

This is checking if the non – functional requirements have been met which is detailed in section 4.1.2. These are explaining the standard the website must withhold.

- NFR-01 – The site is easy to use so the application was designed to be straightforward, where once they're logged in the user is directed to the main page with featured items to make it easy to understand how to use the website.
- NFR-02 - The interface of the site is on the simpler side, so it is easy on the eyes. This is also referring to NFR-01 of how it should be easy to use so having a simple interface will allow user-friendly interface.
- NFR-03 – The search bar will have a spell suggestion to ensure the user is able to quickly identify products of their need and correct their spelling when it is needed. NFR-04 – Navigation bar is added to the website which is sub categorise of products for users to easily direct their choice of products or give them another method of cursing around the site.
- NFR-05 – User profile has not been added to the website. It will be added in later updates which will allow the user to show their history of lending out goods and other necessary information to help users to trust the source.
- NFR-06 – The website can accommodate all screen sizes which includes mobile phone displays and other electronic devices.
- NFR-07 – All necessary feedback messages are given such as the wrong password, logging in, billing information, etc. This is added to ensure the user understand the navigation of the site and to troubleshoot for users.

Chapter 8

Conclusion

8.1 Summative Evaluation

The overall objective of the project was to create a new approach for renting/lending any items which include unique items that are hard to find in order to target a wide range of audience, including the niche market. This a new way of monetising goods for potentially creating another source of income. Overall the specification of the application is to find products that can be rented or lent out with any products.

The objective was to produce renting/lending features, to do this the front end had typical functionality inputted which includes feature items, registering, user profile, and search bar. The front-end development was done first which was featuring common front-end functionalities such navigation bar, featured items, search bar, etc. With the making of the web application, the hardest part was implementing AJAX with the HTML. This was added to prevent the webpages from reloading but instead updates the page. To accomplish the integration of AJAX was by using XMLHttpRequest. This was added for the login page to allow better user experience so it would smoothly transition to the next webpage. When adding AJAX there was a learning curve because it is a mix of JavaScript and XML. This was a new technique for the group, so a lot of research was done in order to execute a smooth changeover of the webpages.

For the back-end implementation MYSQL database was used for route handling, form validation, user feedback messages, and user details. Before setting up the back-end, database normalisation was made which is putting data in normal forms. The group made the decision of making database normalisation, so it reduces redundant data and ensures data dependency by storing only related data within the table. After normalising the database; setting up the database was easily done as the group referred to the database normalisation to set up the database.

As both the front-end and back-end implementation was done, different techniques were used to connect the database with the website so when any user information can be added to the database. Queries were written to perform CRUD functions that were put into a PHP script which was integrated into the website. This was one of the biggest challenges the group was facing as PHP was new language for the group, so further research was done but even with the additional research it was not properly implemented as result of that a lot of issues occurred such as when registering to the web application since emails couldn't be processed, preventing users from having an account made.

Due to the lack of time further development of the website could not be made therefore producing an unfinished outcome. If more time was given, then there would have

been the better outcome as more researched would have been done such as integrating PHP script and perfecting the web application features.

8.2 Potential Development

These potential further developments are a suggestion that was given to the testing users. With the feedback of the web application and using the survey we have asked questions which features would they prefer to have and many were given. When the requirements are fulfilled, further developments can be done to improve the site which includes having a detailed user profile.

The user profile would have information such as:

- Active or inactive status
- Selling rates
- Average rating of the user
- Reviews of the lender/renter experience

Videos will be added. When looking through products, having another form of displaying goods will increase the user experience as having a demonstration will help the user to make a better decision.

GPS location is a feature to consider, where the user can find products based on their location. This is so it prevents them from traveling far and grants the option of finding the nearest product. Having a location feature will come in handy as it can be merged with other features as it a multi-purpose function.

Filter button that allows the user to refine the products. This is so they can cut down the products by having a current location, categories and price range. With this, it can introduce new products that are more directed to the user needs.

Camera feature, where the user can take the picture on demand without needing to go through the gallery and upload a picture. This is for a quicker process that allows the user to lend out but with a speedier approach.

Chapter 9

Appendices

jusRentit User Feedback Survey https://docs.google.com/forms/d/1Mn6BWRkqKmUDTN2t7an5roULwe85fBIH7e-zgq7CjhU/edit#responses

jusRentIt - GitLab Group 15 https://gitlab.doc.gold.ac.uk/software-project—group-15/jusrentit/

Figure 9.1: Backlog

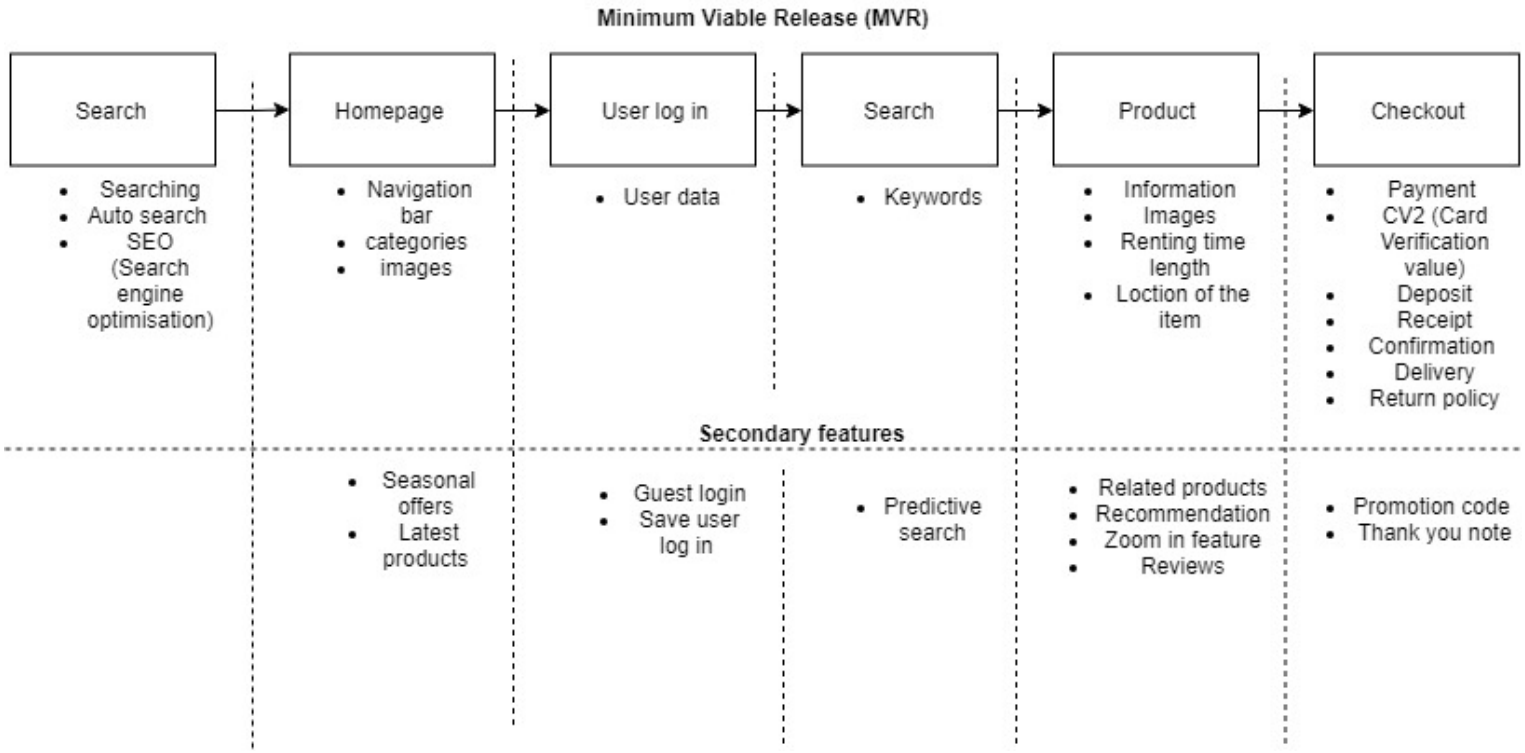


Figure 9.2: Posting Ad

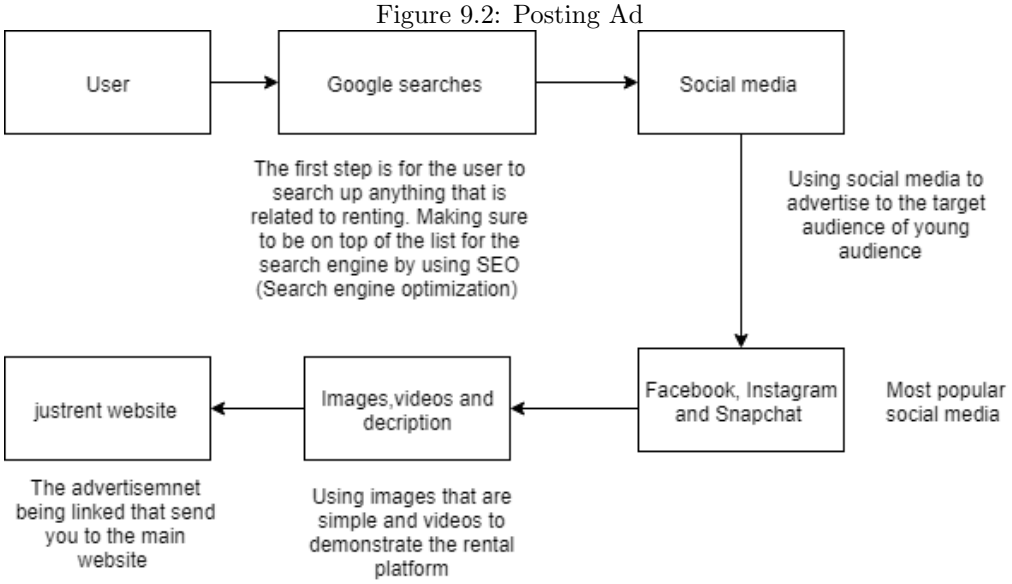
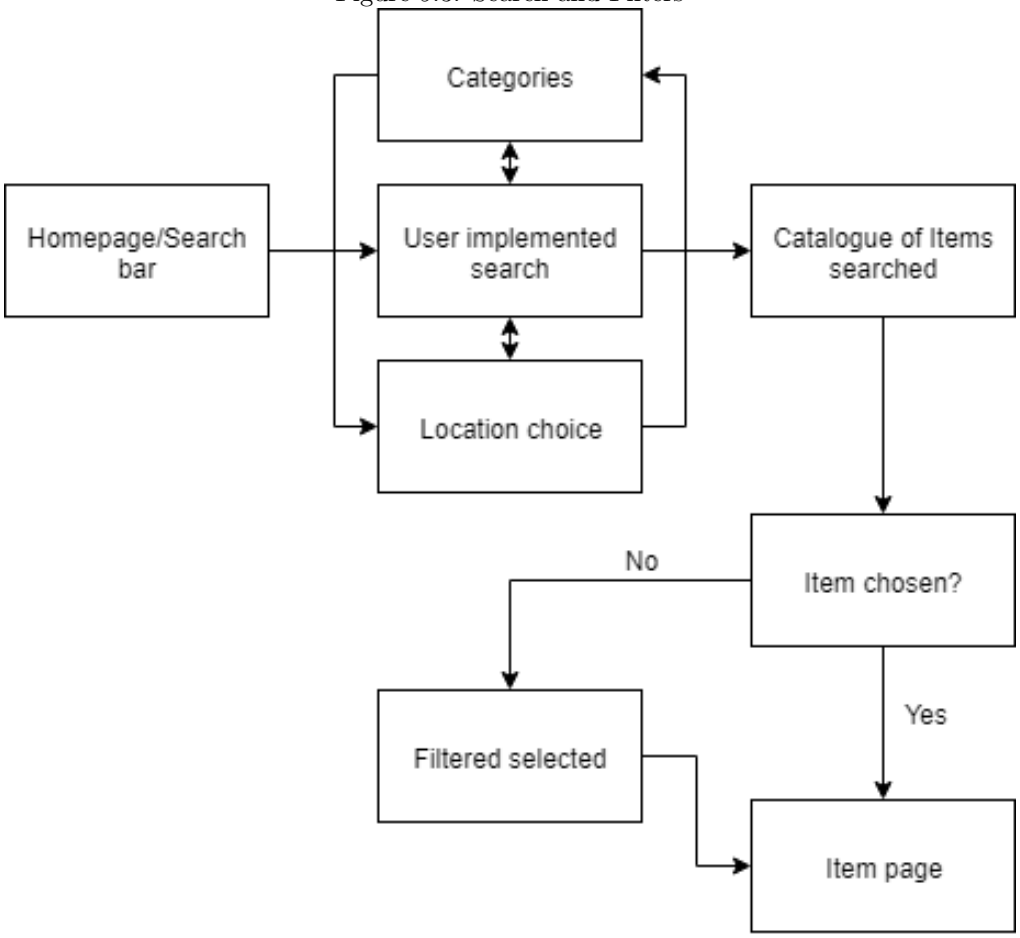


Figure 9.3: Search and Filters



Both registered and unregistered users will have the same experience when searching items on the website where they will search for their desired item, categories (optional) and location to help break down the search to a small size.

The items are then displayed, the user can choose their item that they choose to rent or they can either cut down the list further by applying filters.

Figure 9.4: Payment

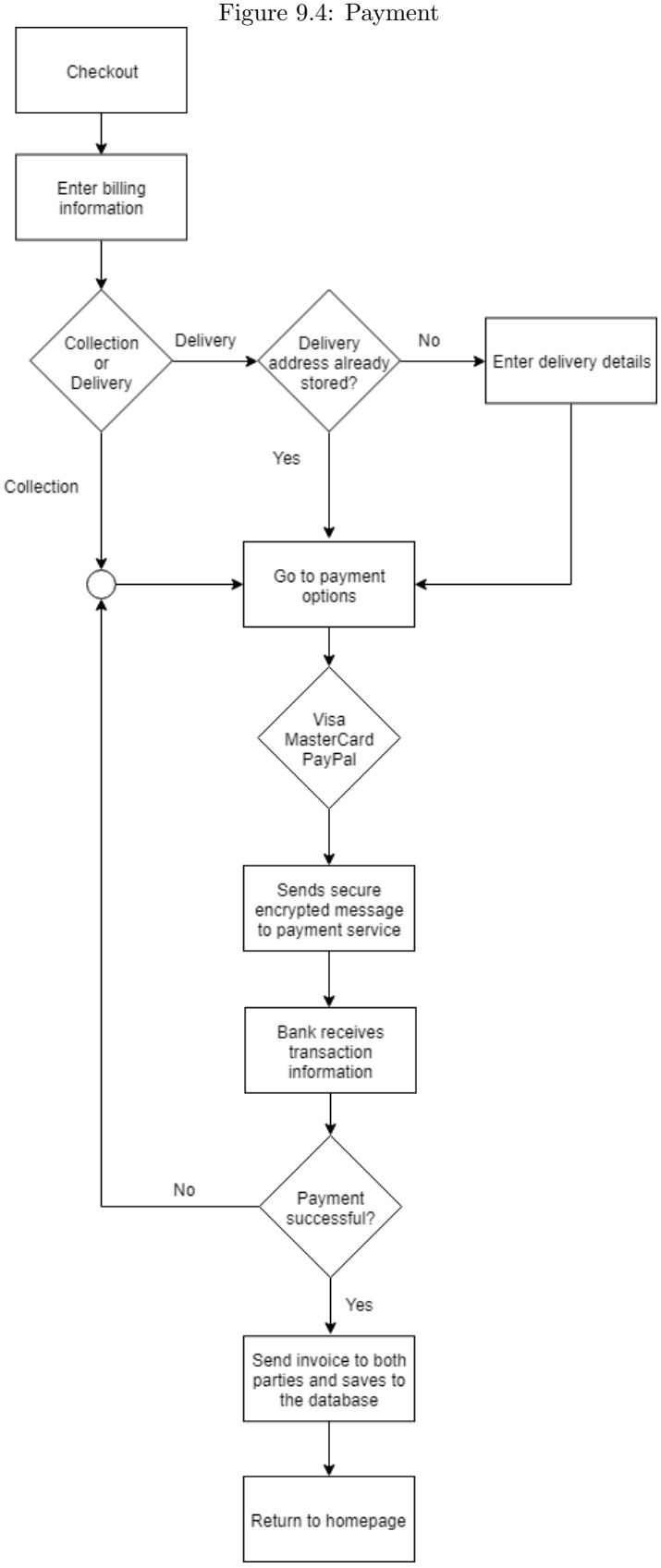


Figure 9.5: Shopping Cart

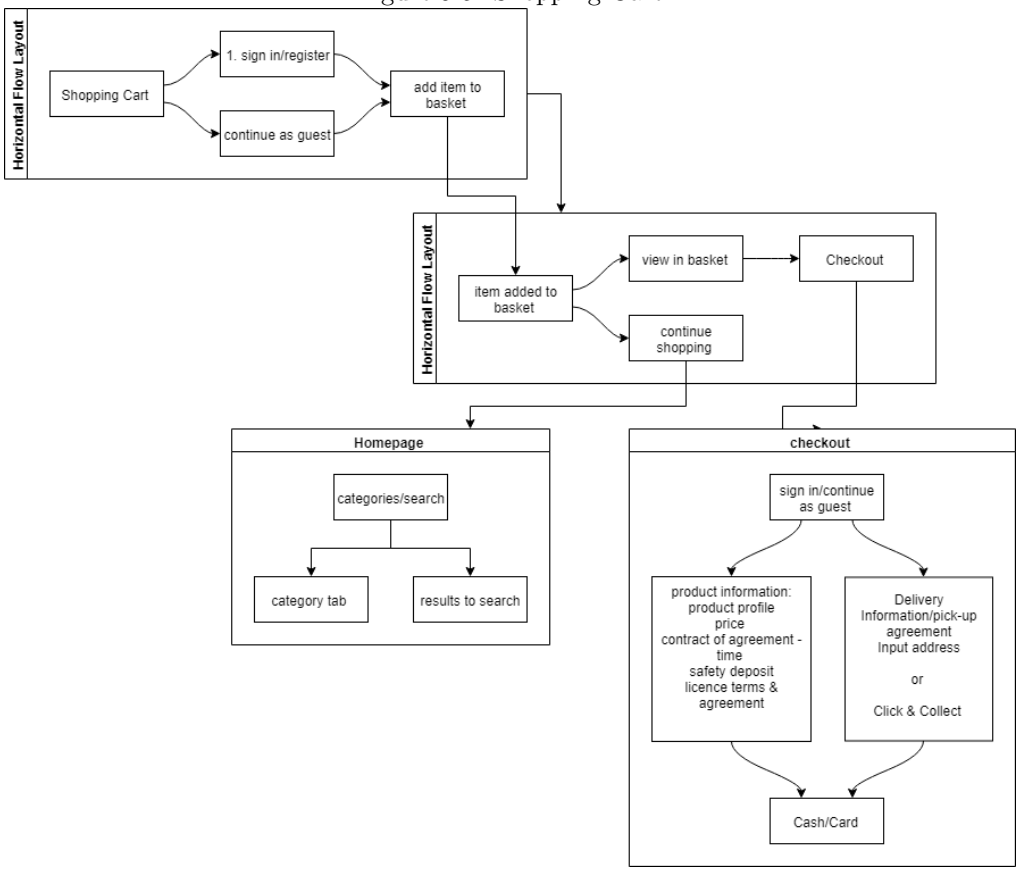
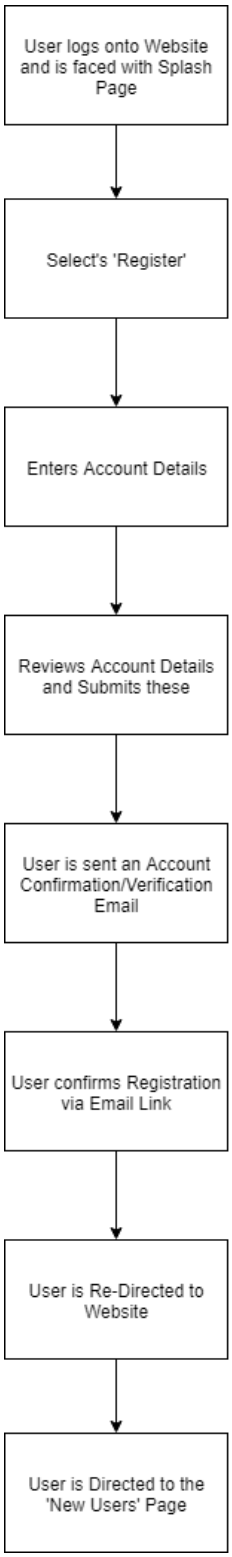


Figure 9.6: User Registration



User Registration - compatible with both Lenders and Renters

I want to be able to create an account to be able to have full user access to the service.

Scenario: Unregistered User should see a link to user signup
GIVEN as an unregistered user
WHEN I go to the main site
THEN I should see a link to signup as a user

Scenario: Unregistered User should see a registration form
GIVEN as an unregistered user
WHEN I go click on 'signup'
THEN I should see a form allowing me to sign up to the site

Scenario: Unregistered User should receive confirmation email on signup
GIVEN as an unregistered user
WHEN I fill out the signup form
AND click submit
THEN I should receive a confirmation email with a link that activates the account

Scenario: Unregistered User should be able to activate user account
GIVEN as an unregistered user
AND I have received a confirmation email
WHEN I click on the activation link
THEN I should have an activated account
AND I am automatically logged in
AND I am taken to the landing page for new users

Acceptance Criteria

The user cannot register an existing account

Users will be given the option to register as a Lender or Renter

The user cannot submit the form without all mandatory fields being completed

Information provided by the user during registration is held on a database of all registered users

The user should receive a confirmation/verification email if the provided email during registration exists

The account activation link in the confirmation email is only valid for 24 hours where their account will be confirmed

Figure 9.7: Survey Chart 1

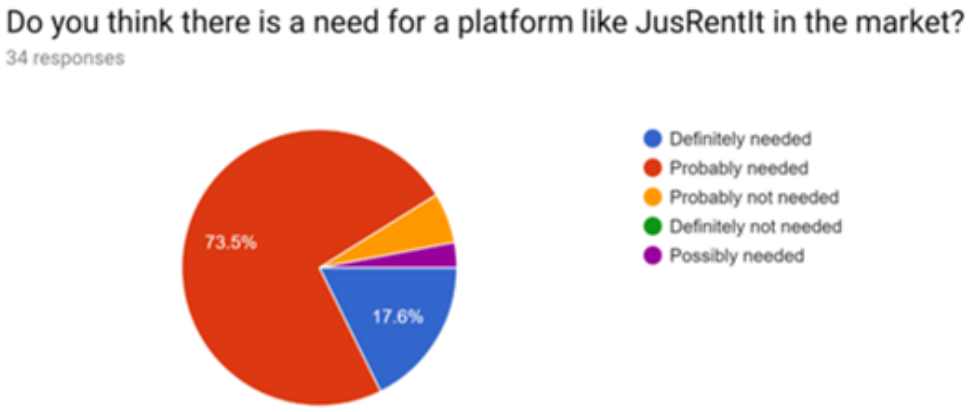


Figure 9.8: Survey Chart 2

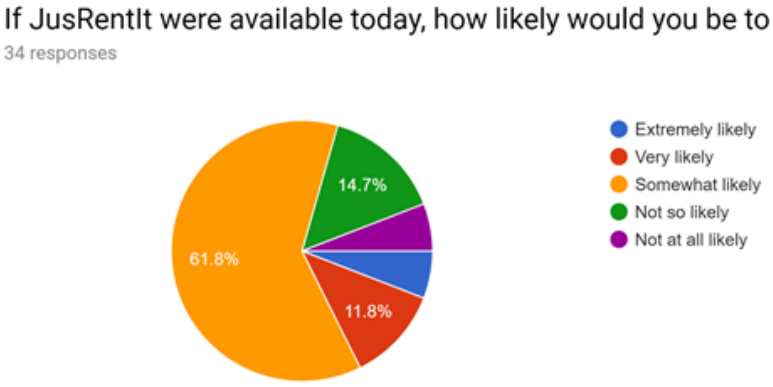


Figure 9.9: Survey Chart 3

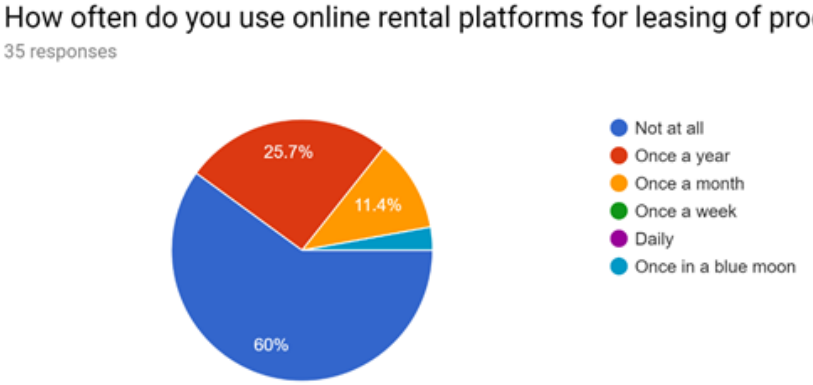
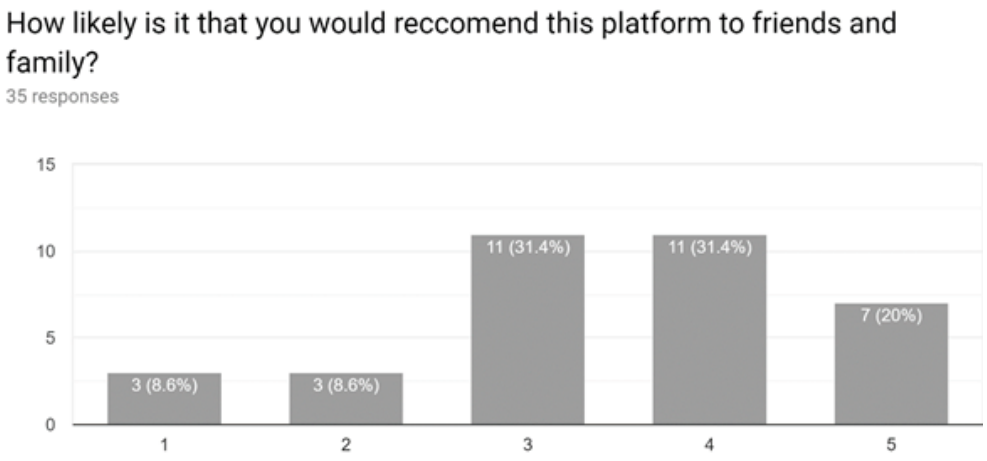


Figure 9.10: Survey Chart 4



Chapter 10

References

Bandakkanavar, R. (2018). Software Requirements Specification document with example - Krazytech. [online] Krazytech. Available at: https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database [Accessed 9 Dec. 2018].

Simply Business Website Written by: Anna Delves, 24/10/2017 Accessed on: 19/04/2019 https://www.simplybusiness.co.uk/knowledge/articles/2017/10/london-rent-prices-a-short-guide/

The Beginner's Guide To Scrum And Agile Project Management Written By: Andrew Littlefield, 02 Sept 2016 Accessed on: 19/04/2017 https://blog.trello.com/beginners-guide-scrum-and-agile-project-management

User Registration with Email Verification using PHP and Mysql Written By: Shahrukh Khan, 24 July 2014 Accessed on: 15/02/2019 https://www.thesoftwareguy.in/user-registration-email-verification-using-php-mysql/

The Ultimate Web Development Course 04-02: Bare bones HTML (1) Written By: Richard Stibbard, 29 Sept 2013 Accessed on: 15/02/2019 https://bit.ly/2DEprSn

END OF DOCUMENT