SOFTWARE REQUIREMENTS SPECIFICATION (SRS)
FOR

RentO

Team Name: The Underbots

Class: Fall 2022 – Tues. and Thurs. Class

Instructor: Professor Edmund Dantes

Revision History

Revision Letter	By	Change Description	Date
1.0	The Underbots	Initial Draft of the SRS	10/21/2022

Table of Contents

Table of Figures

List of Tables

1. INTRODUCTION

1.1 Scope

This Software Requirements Specification defines the requirements for the The Underbots' project called RentO. The RentO web application is a website where people feel comfortable and safe renting out and leasing electrical equipment for passive income. This website will focus on electrical equipment and tools from music, gaming, video recording, and many more.

1.2 Product Value

The value of this product is that it will create a friendly community where people are able to rent an equipment or lease their own and either make passive income while they focus on something else or rent one out for a project or to test it and see if they enjoy it, without going to a major store like Rent-A-Center or Craigslist.

1.3 Intended Audience

The intended audience is for people that want to rent their electrical equipment, such as guitars, speakers, laptops, and many others, for passive income, instead of having it collect dust. It is also intended for people that want to use this same equipment without buying it or wanting to try it first and decide if that is the right product for them.

1.4 Intended Use

The intended use of RentO would be to use a web application that feels safe and user friendly, unlike other places like Craigslist. The user feels confident in using our website to either rent an item or place their equipment up for rent and try to make some money.

2. FUNCTIONAL REQUIREMENTS

FUNC_WEB_001: The web application shall have a button that contains our logo and when clicked, takes the user to the home page.

FUNC_WEB_002: The web application shall have a search function to find a specific item or a type of item the user is interested in.

FUNC_WEB_003: The web application shall have a drop-down menu that will show assorted options for the user, including categories of the items they are searching for, recently added items, and price range of the items.

FUNC_WEB_004: The web application shall have a filter system that will make the search results easier to look at and help narrow the search based on price, categories, condition, the date item was added and the ratings.

FUNC_WEB_005: The web application shall have a function that will be used to access the database of RentO and easily acquire the necessary data to complete a task.

FUNC_WEB_006: The web application shall have a mock payment system to showcase how the users will pay for the rental and how the users that rented out their items will get paid.

FUNC_WEB_007: The web application shall have an email system, where the users are able to communicate with each other by clicking on the link that sends them to their email, ready to type and send a message to the user they are interacting with.

FUNC_WEB_008: The web application shall have a two-way review system, where the users will be able to review the items they use and the users as well, based on their interaction with each other.

FUNC_WEB_009: This web application shall have a cart button that would take the user to a page where it displays the items that the user has added, ready to be rented. This will display the price, name of item, user that is leasing it, and how long the item is being rented out. It will also include a button link that will take the user to the checkout page, where they make the final transaction.

FUNC_WEB_010: The web application will have a Checkout page that will be the destination to make a transaction of a rental. This will show the payment method to be used, the total amount, including price of the item, tax, fee for the website and if offered by the leaser, price of shipping. Once the order is final, the details will be added to the Order History page and the data is stored in the database.

FUNC_WEB_011: This web application shall have a order button, where it will display the user the items they had rented or are currently renting out via a list of history transactions. All this will be shown in an Order History page (FUNC_ORDER_001).

FUNC_WEB_012: This web application shall have an account button that would display the Account page for the user to do different task, including signing in, logging out, adding or changing details about account and accessing other pages, such as order history. This button would change design to let the user know they are signed in, such as displaying their name.

FUNC_WEB_013: This web application shall have a footer section that will have the Contact Us/ FAQ/Help links to the appropriate page for assistance or questions. It will also include a link to an About Us page to display information about the company and store. And it will also have a copyright symbol for security and legal purposes.

FUNC_WEB_014: This web application will have a Back to the Top button, so that the users are able to go to the top of the page instantly, instead of scrolling back.

FUNC_ACC_001: The web application shall have an account system where the user's data is stored into a database and can be access for future use, including rental history, items to be listed, search history, payment stored, user's address and contact details, and unique ID to attach to items they listed.

FUNC_ACC_002: The web application shall have a login and password system to maintain proper security and privacy for the user.

FUNC_ACC_003: The web application will have a unique identifier for the account system so no two accounts are the same and will be easier for identifying items listed and who is renting it.

FUNC_ACC_004: The web application will have a function to allow users to create an account by filling out basic information, such as email, name, and password and soon after generating a unique ID for their account.

FUNC_ACC_005: The web application will have a function for the user to properly login and logout of their account. This will be used for security and privacy reasons.

FUNC_ACC_006: The web application will have a review system for the account that will display the current rank the user has obtained from their previous interaction with other users. This will show how the number of users that have reviewed the account, as well as the ranking, based in 5 stars, where one star is bad service, and 5 stars is considered excellent service. This data will be stored in the database.

FUNC_ACC_007: The web application shall have a settings function where the user is able to change details of their accounts, such as email address, name, contact information, address, preference in the items they are interested, and payment method. This data will be stored in the database.

FUNC_ACC_008: This web application shall have a page where the user is able to list their items for rent, by listing the items and adding details, including name of item, description, condition, price, photos of item, and how it can be delivered to the renter, via local pickup or shipping.

FUNC_ACC_009: This web application shall have a page where the user is shown the items that they still have and the items that are still being rented, including contact information for the person they lease it to.

FUNC_ACC_010: This web application shall have a function where the user is able to notify the website that they have received the item from the person they rented out to, including the option to review the renter and instantly put the item up for rent with minimal effort, using the stored data of the item, wit the option of editing a few details.

FUNC_HOME_001: This web application shall have a listing of the items that can be rented on the homepage. This will show assorted items, either the ones that the user prefers or just random items to see if the user is interested.

FUNC_HOME_002: This web application shall have a section for recommended items based on what the user has searched or rented in the past. This will be a separate listing compared to FUNC_HOME_001.

FUNC_CART_001: This web application shall have a cart page where it will show the items that the user wishes to rent out before making the final transactions. While FUNC_WEB_009 is for a button/link, this will be the page where we will have the details mentioned in the previous function displayed for the user. This includes name of item, price, owner of item, and duration of the item being rented.

FUNC_CART_002: This web application will have a function where the user is able to save the items listed in the cart for later use, so they do not have to worry about finding the item again.

FUNC_CART_003: This web application will have a checkout button that will link the user to the checkout page, which is FUNC_WEB_010.

FUNC_ORDER_001: This web application shall have a Order History page where the user can see the transactions they have made and what item they are currently renting. This data is retrieved from the database. Clicking each item will give them more details in a small popup box, such as the total costs, the contact info of the leaser, and the name and description of the item.

FUNC_LIST_001: This web application shall have a Listing Page where more details are being displayed of an item the user has clicked, including the photos, condition, description, and star rating.

FUNC_LIST_002: The web application shall have an Add to Cart button, to allow the user to place the item they wish to rent into a cart for storage until they go to the checkout page.

FUNC_LIST_003: This web application shall have a Contact Seller section for the user to have access in case they need to communicate with the leaser and ask questions or comments.

FUNC_LIST_004: This web application shall have a function in the Listing Page where it will display if the item is available for rent or is out for rental.

FUNC_LIST_005: This web application shall have a function where if the unique ID of the item matches with the unique ID of the leaser, they are able to see a button to edit the listing, such as images, details, delete the listing. If it does not match, the button will be hidden.

FUNC_SIGNIN_001: This web application shall have a Sign In/Create page where FUNC_ACC_004 will be used to allow a user to create an account for the RentO site. This will also use FUNC_ACC_002 to have the user login using an email address and password.

FUNC_SIGNIN_002: This web application will have a function that allows the user to reset their password in case they have forgotten via email.

FUNC_SIGNIN_003: This web application will have an email verification system where the user will receive an email containing a unique code that must be entered in our website in order to verify their email address.

FUNC_CONTACT_001: This web application shall have a page dedicated to Contact Us, Help and FAQ sections for the user to have in order to contact us or seek assistance.

Constraints:

- Versions of web browsers need to be up to date. Will not support older versions. Main four browsers to support will be Chrome, Firefox, Safari, Edge.
- Need to deal with many sizes of display, since this can be used on PC, tablets, and smartphones with various display sizes.
- Users need to have access to the internet to use our web application.
- Lenders need access to a camera (regardless if it’s a smartphone or actual camcorder), in order to take pictures of the items they want to rent out and add it to their listing.
- Users will need to have access to an email in order to use the web application including renting or leasing an item.
- The programming language will be JavaScript and will be limited to what JavaScript has to offer.

- The project will use a mock payment system to demonstrate how the transaction and payment method works. The group will not implement any real money system until we agree to work on this project after Comp 490/491.

- The web application will only be available in English. As of now, there are no plans to add various other languages.

3. EXTERNAL INTERFACE REQUIREMENTS

3.1 User Interface Requirements

EXTINTF_WEB_001: The web application shall have proper handling of sensitive information, such as hiding the password and credit card information so that no unauthorized user can see them.

EXTINTF_WEB_002: The web application shall have the mouse and keyboard as the main input source for the user on PC and touch screen for tablets and smartphone devices.

EXTINTF_WEB_003: The web application shall follow the standard guidelines established by the industry of web development to make sure our users get the best experience using our website.

EXTINTF_WEB_004: The web application shall have a help page, where the user is able to go to and either ask questions to our team or see if their questions are in the FAQ section.

EXTINTF_WEB_005: The web application shall have the ability to change the display size from PC to smartphone, based on the size of the window on PC/Mac and able to adjust to the smartphone's display size.

EXTINTF_WEB_006: The web application shall be able to be accessed in any of the main four browsers, including Microsoft Edge, Google Chrome, Mozilla Firefox, and Apple Safari.

EXTINTF_WEB_007: The web application will use the software, Figma, to make the layout of our interface very user friendly and able to edit with ease if the user requests or provides feedback.

3.2 Hardware Interface Requirements

EXTINTF_WEB_008: The web application shall have all server-side components executed on a server-class computer, which in our case would be Amazon Web Services.

EXTINTF_WEB_009: The web application shall have all client-side components be executed on personal computers of several types, including PC, Mac, iPhone, Android, Tablets, and many others.

EXTINTF_WEB_010: The web application must be accessed via the internet, so all the hardware, regardless if it's server or client side, must be connected to the internet via LAN, Wi-Fi, hotspot, mobile data, and various other methods.

3.3 Software Interface Requirements

EXTINTF_WEB_011: The web application shall communicate with MongoDB in order access the data stored in the database for our website.

EXTINTF_WEB_012: The web application shall communicate with a Sales and Tax system to get the accurate cost for the items being rented, including sales tax, if the state the transaction is being made has any.

EXTINTF_WEB_013: The web application shall communicate with NodeJS to use manipulate the data from the database and use it accordingly for the website.

3.4 Communication Interface Requirements

EXTINTF_WEB_014: The web application must follow the client-server model. Communication between the server and client will be done in the Amazon Web Services and will use a HTTPS protocol connection over the internet.

EXTINTF_WEB_015: The web application will use a simple mail transfer protocol or SMTP for the users to use to communicate among each other in our website. This is a TCP/IP connection.

4. NON-FUNCTIONAL REQUIREMENTS

4.1 Security

NONFUNC_WEB_001: The web application will follow industry standards to maintain a secure web application. The practice includes defining access-level requirements of the user's account and prevent someone other than the owner from accessing and potentially stealing personal information. Database security must also meet industry standards to protect the information of our user.

4.2 Capacity

NONFUNC_WEB_002: The web application will have a database, using MongoDB, which will be hosted at the Amazon Web Services. This database will be able to store the data we need for our web application and will be able to increase the capacity, in the event we obtain a high volume of transactions, account creation, and listings, using the Amazon Web Services.

4.3 Compatibility

NONFUNC_WEB_003: The web application shall be up to date and be able to be used by various devices, using the most updated web browser to ensure that the user gets the best experience.

4.4 Reliability

NONFUNC_WEB_004: The web application shall not have any system error, such as a crash or hang, unless either the server-side or client side has experienced a loss of internet connection, or the hardware being used on either side is having issues, insufficient RAM, Processor being overworked, slow or interrupted internet, etc.

4.5 Scalability

NONFUNC_WEB_005: Since the web application is based on the Amazon Web Services, scalability is managed on their end, at an additional cost, meaning in the event that our web application is experiencing a heavy workload, AWS is able to increase the scalability and maintain a well performance.

4.6 Usability

NONFUNC_WEB_006: The web application shall be easy for users to use without any assistance but will also include accessibility, such as increase text-size for easy reading and text to voice speech for users with impaired vision. The main tools to use the web application will be a keyboard, a mouse, or the touchscreen, if device has that.

4.7 Other

NONFUNC_WEB_007: The web application will display disclaimer, copyright, trademark, and policies for legal reasons and to inform the user the terms and conditions for using our site.

5. QUALIFICATION PROVISIONS

Qualification in this specification is interpreted as requirement verification. The following are the base definitions for the verification methods.

A – Analysis: Use of analytical data or simulations under defined conditions to show theoretical compliance. Used where testing to realistic conditions cannot be achieved or is not cost-effective. Analysis (including simulation) may be used when such means establish that the appropriate requirement, specification, or derived requirement is met by the proposed solution. Examples include the reduction, interpretation or extrapolation of test data.

D – Demonstration: A qualitative exhibition of functional performance, usually accomplished with no or minimal instrumentation. Demonstration (a set of test activities with stimuli selected by the developer) may be used to show that the CSCI, or a part of the CSCI, response to stimuli is suitable (e.g. observation of fin deployment, etc.). Demonstration may be appropriate when requirements or specifications are given in statistical terms (e.g. mean time to repair, etc.).

I – Inspection: The examination of the CSCI code against applicable documentation to confirm compliance with requirements. Inspection is used to verify properties best determined by examination and observation.

T – Test: An action by which the operability, supportability, or performance capability of the CSCI, or a part of the CSCI, is verified when subjected to controlled conditions that are real or simulated. These verifications often use special test equipment or instrumentation to obtain very accurate quantitative data for analysis.

Table I. Requirements Verification

SRS Req. ID	Paragraph Title	Verification Method
FUNC_WEB_001	Shall have a button that contains our logo and when clicked, takes the user to the home page.	Inspection
FUNC_WEB_002	Shall have a search function to find the specific item or a type of item the user is interested in.	Demonstration
FUNC_WEB_003	Shall have a drop-down menu that will show the assorted options to choose from.	Inspection
FUNC_WEB_004	Shall have a filter system that will narrow the search results based on the user's choice of preference.	Demonstration
FUNC_WEB_005	Shall have a function that will be used to access and acquire information from the database of RentO	Demonstration
FUNC_WEB_006	Shall have a mock system to showcase how users will pay for the rental and how leasers will get paid.	Test
FUNC_WEB_007	Shall have an email system, where the users are able to communicate with each other.	Inspection
FUNC_WEB_008	Shall have a two-way review system, where users are able to review the products, they rented and each other.	Demonstration
FUNC_WEB_009	Shall have a cart button that would take the user to the page where it will display the items they added.	Inspection
FUNC_WEB_010	Shall have a checkout page that will show the total costs of the rentals and ask for the payment method. Order will be added to the Order History page.	Test
FUNC_WEB_011	Shall have an order button, where it will display the user the items they had rented or is currently being rented.	Inspection
FUNC_WEB_012	Shall have an account button, where it will display the Account page, in which the user can access, edit and perform various tasks in regard to their account.	Inspection
FUNC_WEB_013	Shall have the footer section that will have useful information, including Contact Us, FAQ, and About us.	Inspection
FUNC_WEB_014	Shall have a Back to the Top button for easy access or the users.	Inspection
FUNC_ACC_001	Shall have an account system where the user's data is stored and can be accessed via email and password.	Inspection
FUNC_ACC_002	Shall have a login and password system to maintain proper security and privacy for the user.	Demonstration
FUNC_ACC_003	Shall have a unique identifier for the account system so no two accounts are the same.	Inspection

SRS Req. ID	Paragraph Title	Verification Method
FUNC_ACC_004	Shall allow the user to create an account by filling out information, such as email, password, and name.	Demonstration
FUNC_ACC_005	Shall have a function for the user to properly login and logout of their account.	Demonstration
FUNC_ACC_006	Shall have a review system in the Account page, where the users see how many reviews they received and what is their current star rating.	Inspection
FUNC_ACC_007	Shall have a settings function to allow user change details to their account, including email address, name, contact information, and address.	Demonstration
FUNC_ACC_008	Shall have a page where the user is able to setup a listing for their items for rent, including adding information and images.	Demonstration
FUNC_ACC_009	Shall display a page of the user's inventory, including what they have available and what is still being rented.	Inspection
FUNC_ACC_010	Shall have a function that allows the user to notify the website that they have received their items and is either ready to relist it or make some changes to the details or delete the listed item.	Demonstration
FUNC_HOME_001	Shall have a listing of the items that can be rented on the homepage. This will show assorted items, either the ones that the user prefers or just random items to see if the user is interested.	Inspection
FUNC_HOME_002	Shall have a section for recommended items based on what the user has searched or rented in the past. This will be a separate listing compared to FUNC_HOME_001.	Inspection
FUNC_CART_001	Shall have a cart page where it will show the items that the user wishes to rent out before making the final transactions.	Inspection
FUNC_CART_002	Will have a function where the user is able to save the items listed in the cart for later use, so they do not have to worry about finding the item again.	Demonstration
FUNC_CART_003	Will have a checkout button that will link the user to the checkout page, which is FUNC_WEB_010.	Inspection
FUNC_ORDER_001	Shall have an Order History page where the user can see the transactions they have made and what item they are currently renting.	Inspection

SRS Req. ID	Paragraph Title	Verification Method
FUNC_LIST_001	Shall have a Listing Page where more details are being displayed of an item the user has clicked, including the photos, condition, description, and star rating.	Inspection
FUNC_LIST_002	Shall have an Add to Cart button, to allow the user to place the item they wish to rent into a cart for storage until they go to the checkout page.	Inspection
FUNC_LIST_003	Shall have a Contact Seller section for the user to have access in case they need to communicate with the leaser and ask questions or comments.	Inspection
FUNC_LIST_004	Shall have a function in the Listing Page where it will display if the item is available for rent or is out for rental.	Inspection
FUNC_LIST_005	Shall have a function where if the unique ID of the item matches with the unique ID of the leaser, they are able to see a button to edit the listing, such as images, details, delete the listing. If it does not match, the button will be hidden.	Demonstration
FUNC_SIGNIN_001	Shall have a Sign In/Create page where FUNC_ACC_004 will be used to allow a user to create an account for the RentO site. This will also use FUNC_ACC_002 to have the user login using an email address and password.	Inspection
FUNC_SIGNIN_002	Will have a function that allows the user to reset their password in case they have forgotten via email.	Demonstration
FUNC_SIGNIN_003	Will have an email verification system where the user will receive an email containing a unique code that must be entered in our website to verify their email address.	Demonstration
FUNC_CONTACT_001	Shall have a page dedicated to Contact Us, Help and FAQ sections for the user to have in order to contact us or seek assistance.	Inspection
EXTINTF_WEB_001:	Shall have proper handling of sensitive information, such as hiding the password and credit card information so that no unauthorized user can see them.	Inspection
EXTINTF_WEB_002:	Shall have the mouse and keyboard as the main input source for the user on PC and touch screen for tablets and smartphone devices.	Inspection
EXTINTF_WEB_003:	Shall follow the standard guidelines established by the industry of web development to make sure our users get the best experience using our website.	Inspection

SRS Req. ID	Paragraph Title	Verification Method
EXTINTF_WEB_004:	Shall have a help page, where the user is able to go to and either ask questions to our team or see if their questions are in the FAQ section.	Inspection
EXTINTF_WEB_005:	Shall have the ability to change the display size from PC to smartphone, based on the size of the window on PC/Mac and able to adjust to the smartphone's display size.	Demonstration
EXTINTF_WEB_006:	Shall be able to be accessed in any of the main four browsers, including Microsoft Edge, Google Chrome, Mozilla Firefox, and Apple Safari	Inspection
EXTINTF_WEB_007:	Will use the software, Figma, to make the layout of our interface very user friendly and able to edit with ease if the user requests or provides feedback.	Inspection
EXTINTF_WEB_008:	Shall have all server-side components executed on a server-class computer, which in our case would be Amazon Web Services.	Inspection
EXTINTF_WEB_009:	Shall have all client-side components be executed on personal computers of several types, including PC, Mac, iPhone, Android, Tablets, and many others.	Inspection
EXTINTF_WEB_010:	All the hardware, regardless if it's server or client side, must be connected to the internet via LAN, Wi-Fi, hotspot, mobile data, and various other methods.	Inspection
EXTINTF_WEB_011:	Shall communicate with MongoDB in order access the data stored in the database for our website.	Demonstration
EXTINTF_WEB_012:	Shall communicate with a Sales and Tax system to get the accurate cost for the items being rented, including sales tax, if the state the transaction is being made has any.	Test
EXTINTF_WEB_013:	Shall communicate with NodeJS to use manipulate the data from the database and use it accordingly for the website.	Demonstration
EXTINTF_WEB_014:	Must follow the client-server model. Communication between the server and client will be done in the Amazon Web Services and will use a HTTPS protocol connection over the internet.	Inspection
EXTINTF_WEB_015:	Will use a simple mail transfer protocol or SMTP for the users to use to communicate among each other in our website. This is a TCP/IP connection.	Inspection

SRS Req. ID	Paragraph Title	Verification Method
NONFUNC_ WEB_WEB_0 01	Will follow industry standards to maintain a secure web application. The practice includes defining access-level requirements of the user's account and prevent someone other than the owner from accessing and potentially stealing personal information. Database security must also meet industry standards to protect the information of our user.	Inspection
NONFUNC_ WEB_WEB_0 02	Will have a database, using MongoDB, which will be hosted at the Amazon Web Services. This database will be able to store the data we need for our web application and will be able to increase the capacity, in the event we obtain a high volume of transactions, account creation, and listings, using the Amazon Web Services.	Inspection
NONFUNC_ WEB_WEB_0 03	Shall be up to date and be able to be used by various devices, using the most updated web browser to ensure that the user gets the best experience.	Inspection
NONFUNC_ WEB_WEB_0 04	Shall not have any system error, such as a crash or hang, unless either the server-side or client side has experienced a loss of internet connection, or the hardware being used on either side is having issues, insufficient RAM, Processor being overworked, slow or interrupted internet, etc.	Inspection
NONFUNC_ WEB_WEB_0 05	Scalability is managed on AWS' end, at an additional cost, meaning if our web application is experiencing a heavy workload, AWS can increase the scalability and maintain a well performance.	Analysis
NONFUNC_ WEB_WEB_0 06	Shall be easy for users to use without any assistance but will also include accessibility, such as increased text-size for easy reading and text to voice speech for users with impaired vision. The main tools to use the web application will be a keyboard, a mouse, or the touchscreen, if device has that.	Inspection
NONFUNC_ WEB_WEB_0 07	Will display disclaimer, copyright, trademark, and policies for legal reasons and to inform the user the terms and conditions for using our site.	Inspection

6. NOTES

6.1 Acronyms and Abbreviations

Table II. Acronyms and Abbreviations

Abbreviation	Full name
AWS	Amazon Web Services
FAQ	Frequently Asked Questions
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet-Protocol
LAN	Local Area Network
MongoDB	Mongo DataBase
NodeJS	Node JavaScript
PC	Personal Computer
RAM	Random Access Memory
TCP	Transmission Control Protocol
Wi-Fi	Wireless Fidelity