

Business Requirements Document

“AutoBuild” - Automatic PC Part Builder

October 10, 2020

Saturday Solution

Connor Kobel	(015036474)
Daniel Gulland	(018648771)
Ian Ho-Sing-Loy	(026339220)
Nickolaus Marshall-Eminger	(025567241)
Sirage El-Jawhari	(018359144)
Zeinab Farhat	(017990360)

Table of Contents

1 Document Revisions	5
2 Saturday Solutions Member Directory	6
3 Introduction	7
3.1 Project Summary	7
3.1.1 Objectives	7
3.1.2 Background	7
3.1.3 Business Drivers	7
3.2 Project Scope	8
3.2.1 In Scope Functionality	8
3.2.2 Out of Scope Functionality	9
4 Business Requirements	10
4.1 Priority Legend	10
4.2 Master Functional Requirements Table	10
4.2.1 General	10
4.2.2 Security	18
4.2.3 Usability	20
4.2.4 Audit	22
4.3 Master Non-functional Requirements Table	24
4.4 Archiving	27
4.4.1 Functional	28
4.5 Components Catalog	29
4.5.1 Functional	31
4.5.2 Non-Functional	32
4.5.3 Component Catalog Activity Diagrams	34
4.6 Data Store Access	39
4.6.1 Functional	40
4.6.2 Non-Functional	40
4.7 Error Handling	41
4.7.1 Functional	42
4.7.2 Non-Functional	43
4.8 Logging	44
4.8.1 Functional	45
4.9 Login	46
4.9.1 Functional	46
Login Activity Diagram	48
Fig.1 Activity Diagram that shows a user logging in.	48

4.10 Logout	49
Logout Activity Diagram	51
Fig.1 Activity Diagram detailing how user logouts work.	51
4.11 Most Popular PC Build	52
4.11.1 Functional	54
4.11.2 Non-Functional	55
4.11.3 Most Popular Build Activity Diagrams	56
4.12 Network Communication	59
4.12.1 Functional	60
4.12.2 Non-Functional	61
4.13 PC Builder	62
4.13.1 Functional	65
4.13.2 Non-Functional	65
4.13.3 PC Builder Diagram	66
4.14 PC Upgrader	67
4.14.1 Functional	68
4.14.2 Non-Functional	69
4.14.3 PC Upgrader Diagram	70
4.15 Privacy	71
4.16 Registration	72
4.16.1 Functional	74
4.16.2 Non-Functional	75
4.17 Security	77
4.18 Support Page	79
4.18.1 Functional	81
4.18.2 Non-Functional	83
4.18.3 Support Page Activity Diagrams (to be continued)	83
4.19 User Access Control	87
4.19.1 Functional	88
User Access Controls Diagram	89
4.20 User Analysis Dashboard	90
4.20.1 Functional	91
4.20.2 Non-Functional	92
4.20.3 User Dashboard Activity Diagrams	93
4.21 User Experience	94
4.21.1 Functional	94
4.22 User's Garage	96
4.22.1 Functional	97
4.22.2 Non-Functional	98

4.22.3 User's Garage Activity Diagrams	99
4.23 User Interface	102
4.23.1 Functional	103
4.23.2 Non-Functional	104
4.24 User Management	105
4.24.1 Functional	106
4.25 User Reviews and Ratings	107
4.25.1 Functional	108
4.25.2 Non-Functional	109
4.25.3 User Review and Ratings Activity Diagrams	110
User creates a review on a build	110
User adding a star rating on a build	113
4.26 Vendor Linking	116
4.26.1 Functional	117
4.26.2 Non-Functional	118
4.26.3 Vendor Linking Activity Diagrams	119
5 Appendices	120
5.1 Abbreviations	120
5.2 Glossary	120
5.3 Related Documents	120

1 Document Revisions

Date	Version	Changes
09/29/2020	0.1.0	Initial Draft formatting.
10/4/2020	0.2.0	Added summary, scope, and initial requirements.
10/5/2020	0.3.0	Formatted document, added Abbreviations section, added a related document.
10/6/2020	0.4.0	Added several features and concepts requirements and associated diagrams.
10/8/2020	0.5.0	Introduced Builder and Upgrader.
10/9/2020	0.6.0	Added in all content.
10/10/2020	1.0.0	Finalized and formatted.

2 Saturday Solutions Member Directory

Name	Student ID	Email
Connor Kobel	015036474	crkobel@verizon.net
Daniel Gulland	018648771	dannygulland@gmail.com
Ian Ho-Sing-Loy	026339220	ian.hosingloy@gmial.com
Nick Marshall-Eminger	025567241	hunter3787@gmail.com
Sirage El-Jwhari	018359144	sergejawhari@gmail.com
Zeinab Farhat	017990360	Zeinab.farhat@student.csulb.edu

3 Introduction

3.1 Project Summary

3.1.1 Objectives

- The goal is to create an intuitive website that gives users recommendations for computer components, including peripherals.
- All users will have access to an “Upgrader” and “Builder” that, given certain criteria, will generate a recommended list of components.
- We offer a registration system to support additional functionality on the site, such as the ability to save generated builds (component lists), or the below functionalities.
- Users that register will have access to special storage spaces for “Favorites”, be they posts, builds, or components, as well as the ability to create new threads or respond to existing threads on the support forums.
- Allows users to easily browse a catalog of computer components and peripherals with relevant information and a place to purchase, if available.
- Contains forums to allow for open discussion between registered users about relevant topics.
- Supports search and sort functions on each page for locally relevant information, when appropriate.

3.1.2 Background

The initial proposal was an automatic custom pc builder that took existing benchmarks for components from the web and output an optimal computer build within a price point. This limited our market so we have branched the system out to be more user friendly, more interactive, and more customizable. The new variation of the project contains much more functionality, including the ability to add peripherals to be budgeted for, and supports some laptop upgrades as well.

3.1.3 Business Drivers

- Building a computer, or finding upgrades to existing computer components, can be challenging, customers want an easy to use resource that can help them simplify that choice.

3.2 Project Scope

3.2.1 In Scope Functionality

- Create a product for United States residents, that is usable by those who can understand U.S. English text.
- Create an automated “builder” that recommends optimal components within a selected budget.
- Allow for the builder to account for additional peripherals within the budget that customers may need/want.
- Create an upgrader tool that gives the user a series of recommended parts that perform higher than or equivalent to a selected item.
- Create an upgrader tool that gives a recommended upgrade to an entire system (such as a laptop) or a part that performs equal or better than the selected system/component.
- Allow users to register with the site.
- Create 4 levels of access control: normal, developer, vendor, and admin.
- Allow other users to view profiles, with limited details.
- Allow users to hide their profile from other users.
- Allow registered users to save content to the web system, including generated builds, parts, or posts.
- Allows users to publish builds to the web server for public viewing.
- Create a “Popular Build” section to show builds that have been published to the build section, by users, to be voted on.
- Allows registered users to write reviews and rate published builds.
- Create a reasonably comprehensive catalog of computer components, including peripherals and batteries.
- Provide a support forum section that allows users to share knowledge about computer systems, to discuss problems with parts, and/or get support from others users about how to make a computer.
- Provides vendors with a way to control their listed prices and inventory of products associated with them.
- Allows users to be redirected to the product page of a selected item with a selected vendor.
- This product will not crash or become unresponsive due to coding implementations.
- All errors will be logged, and users will be notified in a clear way.
- All actions, success or failures, will be logged and time stamped, excluding error logs that are being generated by a log that failed (this could cause an infinite loop).
- User information will be recorded anonymously, so as to protect user privacy.

- The user interface will be clean, clear, and not jarring to look at.
- The user experience will be smooth, intuitive, and responsive.
- Stored user information will be stored securely, and guarded vehemently.

3.2.2 Out of Scope Functionality

- Multi-language support.
- Laptop customization.
- Custom parts catalog (such as custom water cooling).
- API to provide to companies that wish to integrate PC Upgrader/Builder.

4 Business Requirements

4.1 Priority Legend

The requirements in this document are prioritized as follows:

Value	Rating	Description
1	Critical	This requirement is critical to the success of the project. The project will not be possible without this requirement.
2	High	This requirement is high priority, but the project can be implemented at a bare minimum without this requirement.
3	Medium	This requirement is somewhat important, as it provides some value but the project can proceed without it.
4	Low	This is a low priority requirement, or a “nice to have” feature, if time and cost allow it.
5	Future	This requirement is out of scope for this project, and has been included here for a possible future release.

4.2 Master Functional Requirements Table

4.2.1 General

Req#	Priority	Description	Rationale
General / Base Functionality			
FR- AC-G-001	1	Archive all errors older than one month	So that developers are able to address errors and fix them in a timely manner, if older errors were to pop up again, we can see how they were resolved,
FR- AC-G-002	1	Archive all user actions older than one month	To track user activity and see common trends over a longer time frame.

FR- CC- G-001	4	Display all products organized into categories.	We want users to be able to search for products easily at a glance.
FR- CC- G-002	2	Each product is represented by a screenshot and product name and average star rating below it.	It is the intuitive flow of information for the user.
FR- CC- G-003	1	Users can apply preset filters to narrow down search for a product	If a user does not know the specific name of the product, preset filters can assist the user by narrowing down the result set to products falling under the filters selected by the user.
FR- CC- G-004	1	Users should be able to search for specific product(s) based on search string.	If a user knows either the full or partial name of a product, he/she can narrow down the result set to product names similar enough to the search string.
FR- CC- G-005	2	Product details should contain a list of vendors accompanied by an “availability” status indicating whether the vendor carries that product. If the vendor carries that product, the user can click on the vendor and be redirected to the product on the vendor’s website.	Since there is no payment system, the user is provided the option to visit the vendor’s website to check out the product if the product is available.
FR- CC- G-006	3	If the vendor does not carry the product, a signup option is provided for the user to receive email notification when the product is available from the vendor.	If the product is not available, the user is provided the option to be notified by email when the vendor carries the product.

FR- CC- G-007	2	Product details also provides account holders the option to save the product to the garage or shelf. Users opting for those options will be prompted to register for a user account.	If the account holder is interested in the product, he/she can either save it in his/her garage or shelf. If a user selects this option, he/she is required to register for a user account.
FR- DSA- G-001	1	Add a record	Add a new record to the datastore
FR- DSA- G-002	1	Delete a record	Enables removal of a record
FR- DSA- G-003	1	Update a record	Whenever an attribute of the record has changed, it is imperative to update the record.
FR- EE- G-001	1	The site will not crash or become unresponsive.	An unresponsive product is not desirable to anyone. Likewise a product that crashes your browser or computer is equally, or moreso, unwanted.
FR- EE- G-002	1	All errors must be handled.	All exceptions must be caught during runtime.
FR- EE- G-003	1	Users will be informed of an error.	Users who understand that an error occurred, and what error occurred, are more able to address, or cope with, the specifics of an error.

FR-LOG-G-001	1	Log all errors	So that developers are able to address errors and fix them in a timely manner.
FR-LOG-G-002	1	Log all user actions	To track user activity and see common trends.
FR-LI-G-001	1	User can log in with credentials	Cannot log in without credentials, there needs to be a login interface to distinguish between accounts and allow access of particular features for accounts.
FR-LO-G-001	1	User must be able to logout	System cannot be secure on the user end if a user can login but not logout.
FR-MPB-G-001	1	The catalog can be filtered by: Gaming, Productivity, and Animation	Users mostly searched for pc builds based on one or more of these criteria.
FR-MPB-G-002	1	There will be a search function for finding specific published builds.	If a user knows the full or partial name of the build, he/she can execute the search based on the search string.
FR-MPB-G-003	1	Account holders can publish their own unique builds	This is what populates the most popular pc builds webpage
FR-MPB-G-004	3	Account holders can post comments on individual builds	This provides feedback to the account holder about the build

FR-MPB-G-005	1	Account holders can save a build to their garage	Allows account holders to save the build and potentially tweak the design
FR-MPB-G-006	3	Published builds that do not gain enough recognition (“likes”, views, saves or favorites) within 30 days will be removed.	This is so that builds that do not carry much traction do not sit in the webpage for long.
FR- NC-G-001	1	HTTP/2	HTTP/2, a persistent connection can be used to service multiple simultaneous requests. This allows for an improvement of efficiency of HTTP over the internet.
FR- PB-G-001	1	Builder algorithm	In order for the upgrader to output a build that uses vendor product data an algorithm needs implementation
FR- PB-G-002	1	Provide filters for users to define build.	Without filters the user will just be given an arbitrary build.
FR- PB-G-003	1	Provide a toggle switch to allow users to use the upgrader instead.	The upgrader and builder will operate in generally the same area, so we put them together with a toggle switch.
FR- PU-G-001	1	Allow users to select either individual components or an entire build to upgrade.	Users will want to customize their build preferences so as to get the most appropriate choice they desire.
FR- PU-G-002	1	Provide detailed specs with comparison.	Comparing multiple products will give the user an idea of where

			this particular product stands, and why to buy it.
FR- PU-G-003	1	Give the users a view of the component and it's availability.	If a component is not available a user should know.
FR-REG-G-001	1	An account will be created.	A user needs an account for certain functionality.
FR- SP-G-001	1	Accounts can create a thread	We want to offer users who have an account more access to the autobuild community.
FR- SP-G-002	1	Forum user formatting	This makes user navigation through the thread easy to follow.
FR-UAC-G-001	1	Admin roles have all privileges	Changes cannot be made to the system files if a role does not have access to edit system files and make updates/changes.
FR-UAC-G-002	1	Vendor roles can edit vendor files, edit their user files and access networks.	Vendors should be able to list product information as well as their products on our site.
FR-UAC-G-003	1	Developer roles can edit UI/UX files.	Developers have front end website access to be able to make changes to UI and implement or update easy functionality for the user.

FR-UAC-G-004	1	Basic roles can edit user files and access the network	Basic roles should have permission to access their user files and make basic changes to their accounts
FR-UAD-G-001	1	Data anonymity	All data represented onto the user analysis dashboard is subject to privacy concerns such as no PII
FR-UX-G-001	1	Users should find the content of the website useful.	Users that do not find the content useful will not return to use it in or promote its use.
FR-UX-G-002	1	The site needs to be easy to navigate.	Users that do not find the site easy to use will not return to use it or promote its use.
FR-UX-G-003	2	Users should find the site attractive, and interesting to use.	Users who enjoy using the site, even for simple fun, will continue to drive and promote future users.
FR-UX-G-004	1	The content on the site must be easy to locate.	The faster a user can locate content the more content that can be used.
FR-UX-G-005	1	Disabled users must have the ability to access all content.	Without supporting those with disabilities our user base takes an immediate loss.
FR-UX-G-006	1	Content must be reliable.	Users are coming to us for recommendations on potentially thousands of dollars in expenditures.
FR-UG-G-001	2	Users will be able to login if not already logged in.	The user garage requires users to be registered and logged in to

			perform any of its functionality, to ease that we will prompt them here.
FR-UG-G-002	2	Users will be able to sort data in the garage.	Being able to sort data is a standard requirement.
FR-UG-G-003	2	Users will be able to filter data in the garage.	Filtering will help users find what they are looking for without a direct search.
FR-UG-G-004	1	Users will be able to search the garage for builds or parts with specific names.	Searching for the exact thing you know you want is always more efficient than browsing.
FR- UI-G-001	1	Interface must be legible.	If the user cannot read the text on the page they will not be able to interact, or may become frustrated with the site.
FR- UI-G-002	1	Users must be informed of actions in progress.	If the users know what is going on, they will be less likely to try to reinitiate the action, or to be frustrated by the lack of system response.
FR- UM-G-001	1	Administrators should be able to add/edit/remove users	Our application needs an administrator(s) that has the power to manage users. Without it, we're taking a huge security risk.
FR- UM-G-002	1	Administrators should be able to manage permissions for each type of user	Permissions must be granted to users so their account has some functionality.

FR-UM-G-003	3	Administrators should be able to group users according to their association with different organisations and groups	Grouping makes it easier for assigning roles.
FR- RR-G-001	1	Accounts can write and publish a review.	The feature will not be accepted without this requirement
FR- RR-G-002	1	Accounts can make a rating of 1-5 stars.	The feature will not be accepted without this requirement
FR-VL-G-001	3	Users can create a vendor account to add/update/remove their own products.	This feature isn't necessary for the user to use our application. Our main use is to assist in building a PC on a budget. This is handy for users/vendors to add price information and link their products.
FR-VL-G-002	2	Vendors will be added if the vendor has an API.	This feature isn't necessary for the user to use our application. However, it's helpful for the user to get a rough estimate of how much top companies sell products for.

4.2.2 Security

Security Requirements			
FR- AC-S-001	1	Archive all login attempts	This will help distinguish potential login attackers and potentially ban them by IP, and find recurrent attackers in the archives.

FR- CC-S-001	1	Account holders must be able to login	There is a need to differentiate users and account holders.
FR- EE-S-001	1	Errors must be handled without revealing secure information.	Information vital to maintaining security, such as the users password, or other compromising data should not be shown or logged.
FR- LOG-S-001	1	Log all login attempts	This will help distinguish potential login attackers and potentially ban them by IP.
FR- LI-S-001	1	Verify credentials and passwords	So that users can not easily be hacked or compromised.
FR- LO-S-001	1	Logout must be successful when the user logs out.	The account must be completely logged out, if the user stays logged in, it could cause security issues.
FR- NC-S-001	1	SSL/ TLS	This verifies ownership of the website, prevent attackers from creating a fake version of the Autobuild site
FR- NC-S-002	1	Authentication	
FR- REG-S-001	1	Since registration will be linked to our database, the text forms that the user submits through must be through a parameterized query.	We are preventing SQL injection attacks. If we don't, anyone can delete information from our database.

FR- SP-S-001	1	Non-registered users can only view	We can think of this as a security measure for prevention of unknown users to have access to edit and post threads.
FR-UAC-S-001	1	Basic role, vendor role and developer role users can not have access to edit System files.	Permissions should not be entangled across accounts, it should be clear which roles get which permissions.
FR-UAD-S-001	1	Only admin account can access	This shows the system as a glance reflected by all the logged information. We don't want anyone to access it.
FR- RR-S-001	1	Image recognition software to verify that images are appropriate for posting.	Features will not work without this requirement. Inappropriate images should not be allowed.
FR-VL-S-001	1	Since vendors will be publishing products, they will need access to our database. Posts to the database must be through a parameterized query.	We are preventing SQL injection attacks. If we don't, anyone can delete information from our database.

4.2.3 Usability

Usability Requirements			
FR- EE-U-001	1	Error handling must be swift and light weight.	We want the user to experience as few errors as possible, but when they occur we want the user to be impacted as little as possible.
FR- LI-U-001	1	User interface will provide an easy and clear-cut way for	So the user does not become confused on what they must do.

		users to sign into their accounts	
FR- LO-U-001	1	Must be an easy and clear way in the user interface for the user to be able to logout.	To minimize user confusion when logging out of a system.
FR- PU-U-001	1	Inform the user of potential upgrades and showing the cost of the new component, as well as several alternatives.	The ability for a customer to compare is always a plus, it may even show them that our recommendations are accurate.
FR- REG-U-001	2	Registration should be smooth and simple.	Account creation is the first step in acquiring a user. Since we want users, we need a smooth UX.
FR- SP-U-001	3	Signing up for thread notification	This enhances the user's experience, if a user sees a thread that is true for him but there is no response made for it yet, he can follow it!
FR- UAC-U-001	3	Signing up for thread notification	This enhances the user's experience, if a user sees a thread that is true for him but there is no response made for it yet, he can follow it!
FR- UAD-U-001	1	Admin can filter out which dashboard view to view.	This allows the admin to update the current system as a whole representation to other filters listed user "display trend of"
FR- UI-U-001	1	The interface needs to be simple.	Simplicity aids in user enjoyment. Users who have to navigate to several sections to find the tidbit

			of information they seek will become agitated quickly.
FR- RR-U-001	1	User interface will allow users to easily input their reviews and make ratings without difficulty.	Without a proper UI, the user will not know how or where to write reviews / make a rating.
FR-VL-U-001	1	User interface will allow users to easily add a vendor account and their products.	UI/UX is important. If a user doesn't like how our website looks or if it's difficult to navigate, they'll look for another website.

4.2.4 Audit

Audit Requirements			
FR- EE-A-001	1	All errors will be logged and dated, excluding log errors.	We want to know about all failures and where they happened, however, if we log a log failure, and that subsequent log fails to log that could create an infinite loop of failure logs.
FR- LI-A-001	1	Logins will be logged	This is to fulfill the logging section of the BRD.
FR- LO-A-001	1	Logouts must be logged	This is a requirement of Logging that must be fulfilled.
FR- NC-A-001	1	Logging Requests to and from the server	This is important to draw out a baseline of how requests to the application are at its best, and on average.

FR-REG-A-001	1	All attempts to create an account will be logged.	We want to log all attempts to create an account. User registrations and logins are great places to attempt SQL injection.
FR-SP-A-001	1	Threads will be locked and archived. + Thread maintainability	This allows the forum page to be less cluttered and allow user readability to be much easier. Therefore, the old thread will be removed from the “active” forum and placed into an archived forum for later viewing.
FR-UI-A-001	1	Each interaction with the interface must be logged.	If a user selects to go to the product catalog we will log, with a date and time stamp, that a user has entered the catalog.
FR-UM-A-001	1	All changes related to user management will be logged.	If an elevation attack were to occur, we could easily locate the cause.
FR-RR-A-001	3	Updates made to reviews will be time-stamped.	To provide users the validity of a review, for example older reviews might hold less credibility when compared to new reviews.
FR-VL-A-001	4	Any changes made to a vendor’s product will be logged.	We will log in order to monitor suspicious activity.

4.3 Master Non-functional Requirements Table

ID	Requirement
NFR-CC-001	The categorization of products should be clear to the user.
NFR-CC-002	The flow of information of the product representation should be intuitive.
NFR-DSA-001	Should not take longer than thirty seconds
NFR-EE-001	Users should not be impacted when an error occurs.
NFR-EE-002	Errors should be caught and handled in less than 5 seconds.
NFR-MPB-001	The categorization of products should be clear to the user.
NFR-MPB-002	The flow of information of the product representation should be intuitive.
NFR-MPB-003	The screenshot should have a resolution of at least 72 DPI.
NFR-MPB-004	Filters should be highlighted when selected.
NFR-MPB-005	Account holders are allowed a maximum of publishing three unique builds per day
NFR-MPB-006	Six unique builds per day when a level of notoriety is achieved

NFR-MPB-007	When a build is published, an immutable copy is made, and saved to the websites database, that can no longer be modified.
NFR-MPB-008	Publications are subject to the Autobuild Reputation system.
NFR-NC-001	Configuration of web server
NFR-PB-001	The time taken by the PC builder to output a build should take no more than 5 sec
NFR-PB-001	The many filtering options in builder shows a great importance in the seamless navigation of all filters and descriptions.
NFR-PU-001	Builds must be computed in under 5 seconds.
NFR-REG-001	All attempts to create an account will be logged.
NFR-REG-002	There is an 8 character minimum which must include a capital letter and at least 1 number.
NFR-SP-001	The time to load a view, or to get from one session to another should be limited. We do not want the time to exceed 5sec to get to the view the user wants to load.
NFR-SP-002	<p>We don't want to make an application that takes time to navigate through, or a considerable amount of clicks to get to the view the user wishes to see. Maximum number of mouse clicks = 5.</p> <p>Time between views should be no more than 5sec.</p>
NFR-SP-003	<p>Scalability: our support page should be able to accommodate for future expansions on forums.</p> <p>Up to 6 new sections the support page should accommodate for.</p>
NFR-SP-004	Account restrictions: Maximum amount is 3 threads for all forums. Limited number of posts that they can make to a thread, maximum 5 posts.

NFR-UAD-001	The user analysis dashboard should be easy to follow and understand. For example, Any user who glances at the dashboard should be able to make out the data.
NFR-UAD-002	Admin should have difficulty in navigating through their account to locate the dashboard.
NFR-UG-001	The list and filters will load in under 2 seconds on a standard internet connection.
NFR-UG-002	Filters will remain highlighted after selection as additional filters are added or unhighlighting as they are removed.
NFR-UG-003	The search bar will save the last 5 searches in the same session for quick searching.
NFR-UG-004	The user's name will be placed in front of the "Garage" in the interface (____'s Garage).
NFR-UG-005	The search bar will contain most recent queries until the user leaves the site.
NFR-UG-006	All sorts will be done in 2 seconds or less.
NFR-UI-001	Colors should not be offensive to the eyes.
NFR-RR-001	The review submission type box will have a limit of 10,000 characters.
NFR-RR-002	The review submission box will allow a maximum photo upload of 5 images.
NFR-VL-001	Vendor data will be refreshed every 6 hours.
NFR-VL-002	Vendor listings that have not been updated within 30 days will be removed.

4.4 Archiving

Objectives:

- Store a backup of all logs that are older than 1 month:
 - Log ins, successful and unsuccessful attempts
 - Logouts
 - All user posts
 - All user actions
 - All errors

Background:

Archiving user activity for actions performed on the web app older than one month will provide data on all previous user activity.

Requirements:

- Archive successful login attempts older than one month
- Archive unsuccessful login attempts older than one month
- Archive logouts older than one month
- Archive all user posts older than one month
- Archive all user actions on the web app older than one month
- Archive all errors older than one month
- Privacy needs to reflect that user actions are being archived and stored.

Non-Functional Requirements:

Success: All user actions and errors will be successfully archived as long as they are older than one month and will be available for viewing and assessment.

Failure: Not all user actions or errors will be archived, leaving a black-area of things that might not be able to be assessed for fixing or addressing, also incorrectly archiving data that is less than one month old.

4.4.1 Functional

Req#	Priority	Description	Rationale
General / Base Functionality			
FR-AC-G-001	1	Archive all errors older than one month	So that developers are able to address errors and fix them in a timely manner, if older errors were to pop up again, we can see how they were resolved,
FR-AC-G-002	1	Archive all user actions older than one month	To track user activity and see common trends over a longer time frame.

Security Requirements			
FR-AC-S-001	1	Archive all login attempts	This will help distinguish potential login attackers and potentially ban them by IP, and find recurrent attackers in the archives.

4.5 Components Catalog

Objectives:

Catalog Webpage

- The component catalog is an organized repository of products where a user can browse for certain computer parts or accessories.
- The user can select filters to narrow down his/her search or use the search bar to request for specific parts.
- The product is represented by a screenshot displaying product name, product image, and the overall average star rating encompassed by third party and Autobuild ratings
- Clicking on the product screenshot pulls up an overlay screen showing product details which include:
 - Product Details
 - Option to return to webpage
 - Third party reviews
 - Vendor List
 - User can apply the PC upgrader to the product
 - User can place the product in the user shelf
 - User can save the product to the user garage

Background:

The idea of the components catalog came up as a user friendly feature for users to find products in the most intuitive way possible. That is accomplished through the use of preset filters or search. Once the user finds a product, options are provided for the user to view the product through the vendor's website.

Requirements:

- Display all products organized into categories.
- Each product is represented by a screenshot and product name and average star rating below it.
- Users can apply preset filters to narrow down search for a product
- Users should be able to search for specific product(s) based on search string.
- Product details should contain a list of vendors accompanied by an "availability" status indicating whether the vendor carries that product. If the vendor carries that product, the user can click on the vendor and be redirected to the product on the vendor's website.
- If the vendor does not carry the product, a signup option is provided for the user to receive email notification when the product is available from the vendor.

- Product details also provides account holders the option to save the product to the garage or shelf. Users opting for those options will be prompted to register for a user account.

Non-Functional Requirements:

- The categorization of products should be clear to the user.
- The flow of information of the product representation should be intuitive.
- The screenshot should have a resolution of at least 72 DPI.
- Filters should be highlighted when selected.
- The search bar should have a suggestions drop down to suggest potential products the user might be requesting.

Success: User successfully is able to locate the product through the use of filters or search. Once there users can visit the product from the vendor's website if available. Account users are provided the option to save the product to the user's garage or shelf.

Failure: The user is not able to search for products due to filters and search not functioning properly. Even with a product found, users will not be able to visit the vendor's website or be notified when the product is available by the vendor. Account users are not able to save the product to their garage or shelf.

4.5.1 Functional

Req#	Priority	Description	Rationale
General / Base Functionality			
FR-CC-G-001	2	Display all products organized into categories.	We want users to be able to search for products easily at a glance.
FR-CC-G-002	2	Each product is represented by a screenshot and product name and average star rating below it.	It is the intuitive flow of information for the user.
FR-CC-G-003	1	Users can apply preset filters to narrow down search for a product	If a user does not know the specific name of the product, preset filters can assist the user by narrowing down the result set to products falling under the filters selected by the user.
FR-CC-G-004	1	Users should be able to search for specific product(s) based on search string.	If a user knows either the full or partial name of a product, he/she can narrow down the result set to product names similar enough to the search string.
FR-CC-G-005	1	Product details should contain a list of vendors accompanied by an “availability” status indicating whether the vendor carries that product. If the vendor carries that product, the user can click on the vendor and be redirected to	Since there is no payment system, the user is provided the option to visit the vendor’s website to check out the product if the product is available.

		the product on the vendor's website.	
FR-CC-G-006	1	If the vendor does not carry the product, a signup option is provided for the user to receive email notification when the product is available from the vendor.	If the product is not available, the user is provided the option to be notified by email when the vendor carries the product.
FR-CC-G-007	1	Product details also provides account holders the option to save the product to the garage or shelf. Users opting for those options will be prompted to register for a user account.	If the account holder is interested in the product, he/she can either save it in his/her garage or shelf. If a user selects this option, he/she is required to register for a user account.

Security Requirements			
FR-CC-S-001	1	Account holders must be able to login	There is a need to differentiate users and account holders.

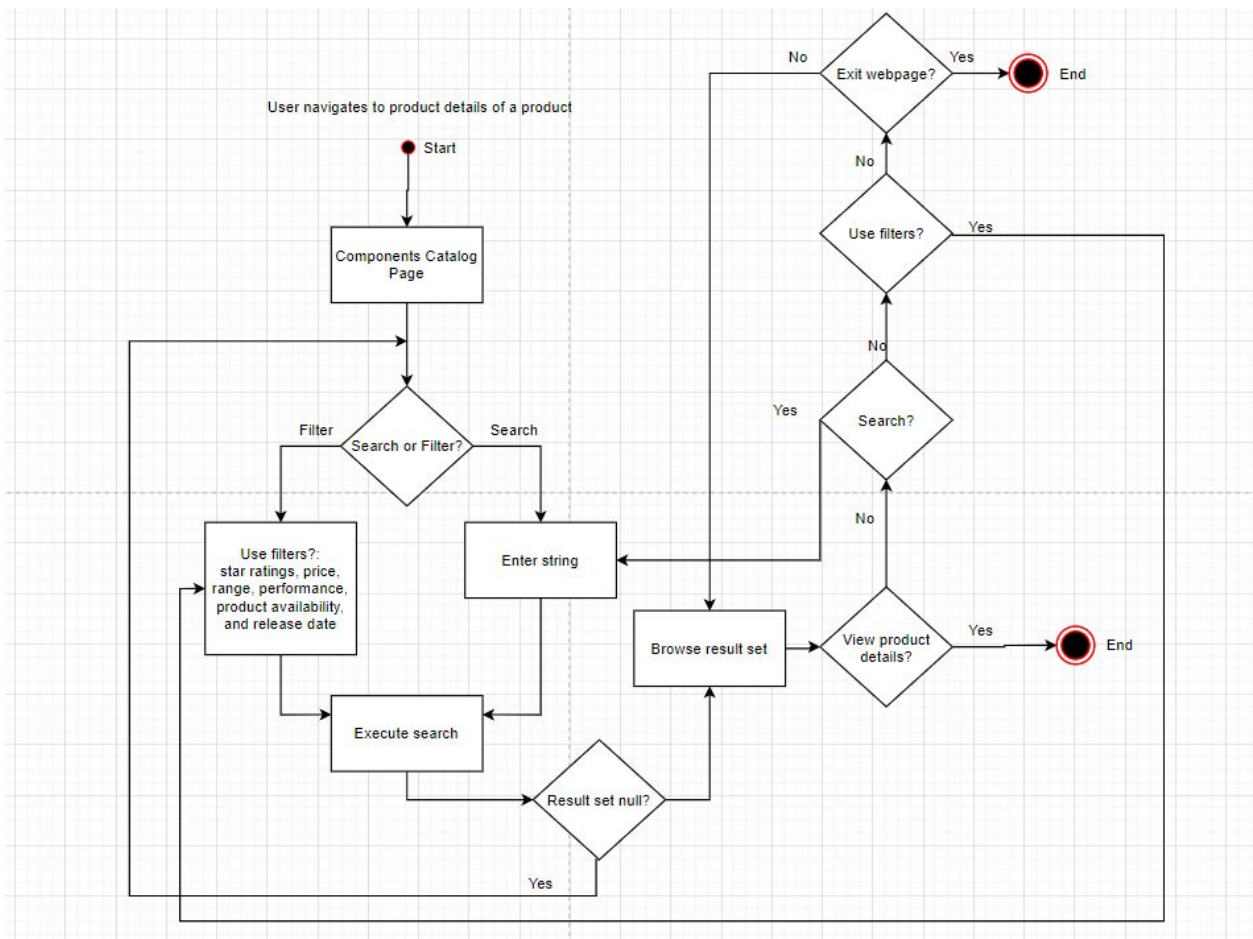
4.5.2 Non-Functional

ID	Requirement
NFR-CC-001	The categorization of products should be clear to the user.
NFR-CC-002	The flow of information of the product representation should be intuitive.

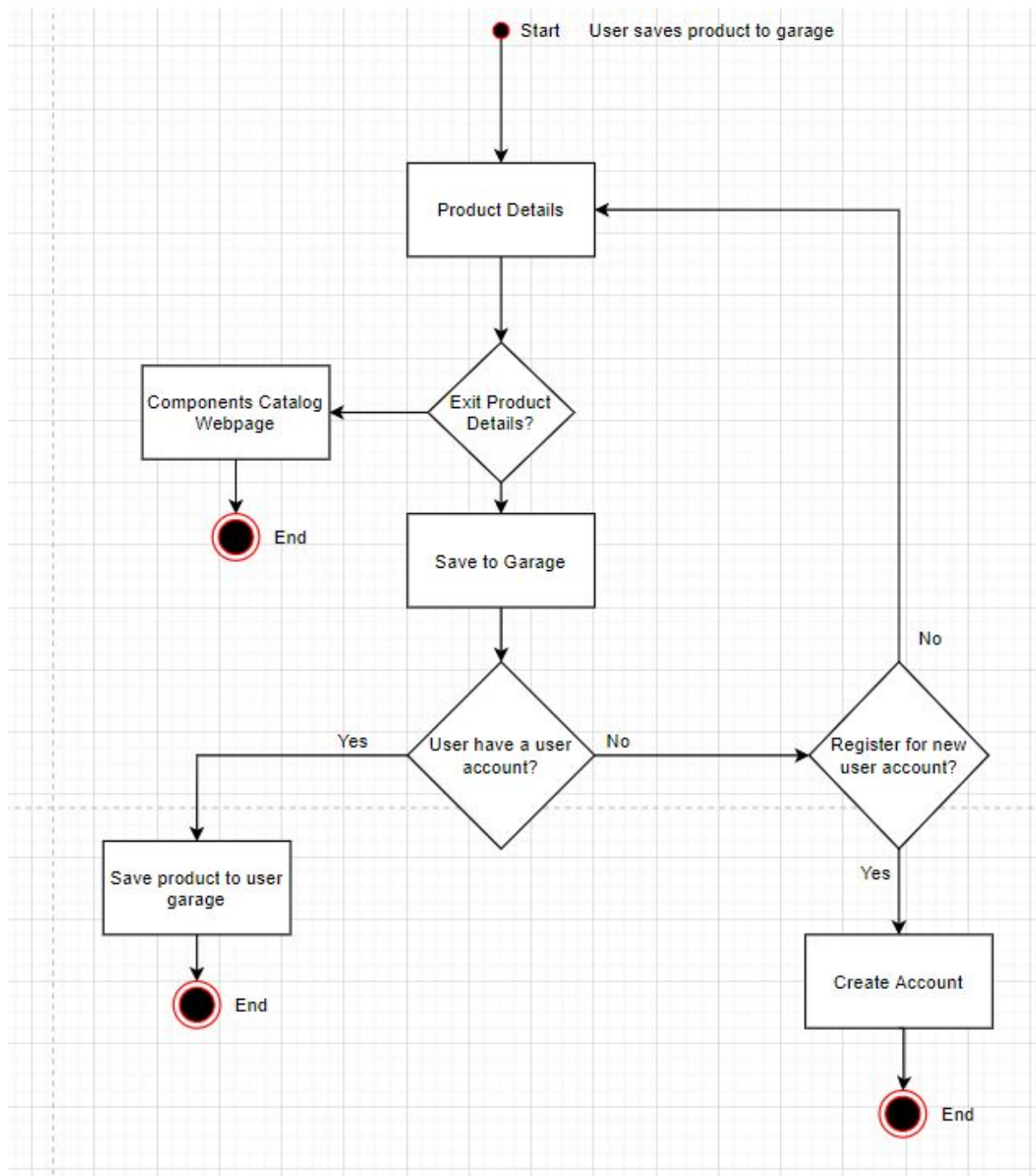
NFR-CC-003	The screenshot should have a resolution of at least 72 DPI.
NFR-CC-004	Filters should be highlighted when selected.
NFR-CC-005	The search bar should have a suggestions drop down to suggest potential products the user might be requesting.

4.5.3 Component Catalog Activity Diagrams

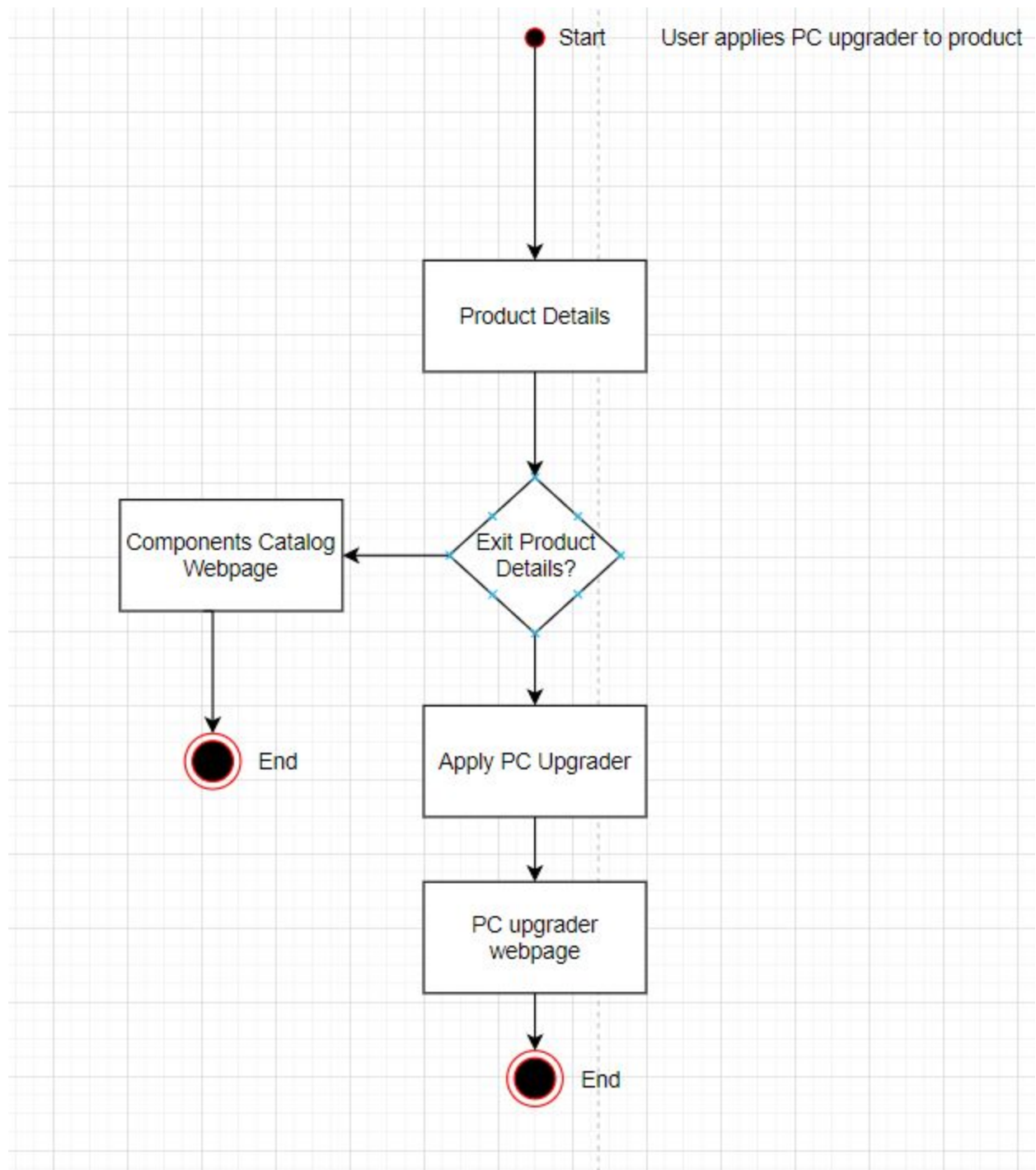
User navigates to product details of a product



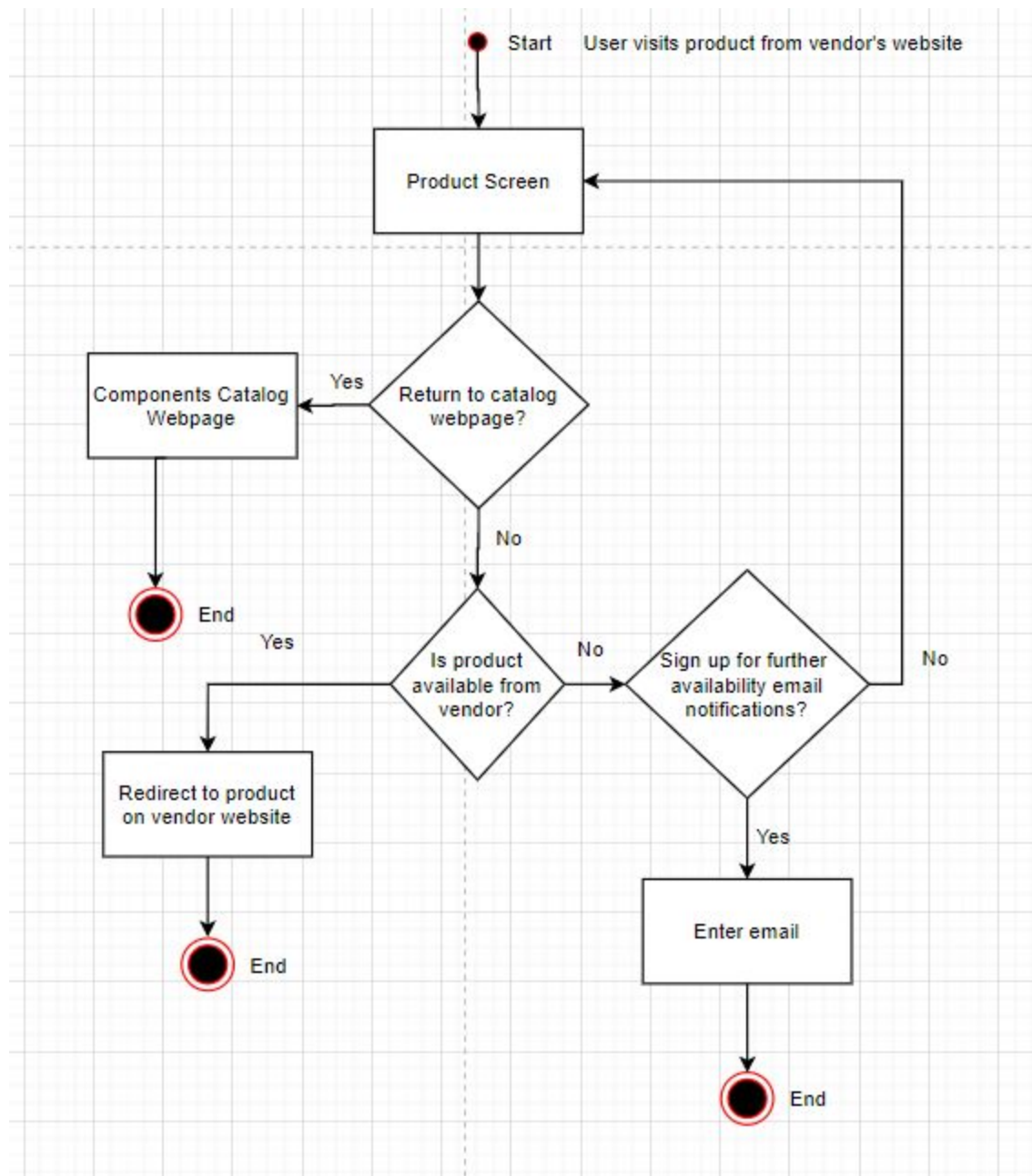
User saves product to garage



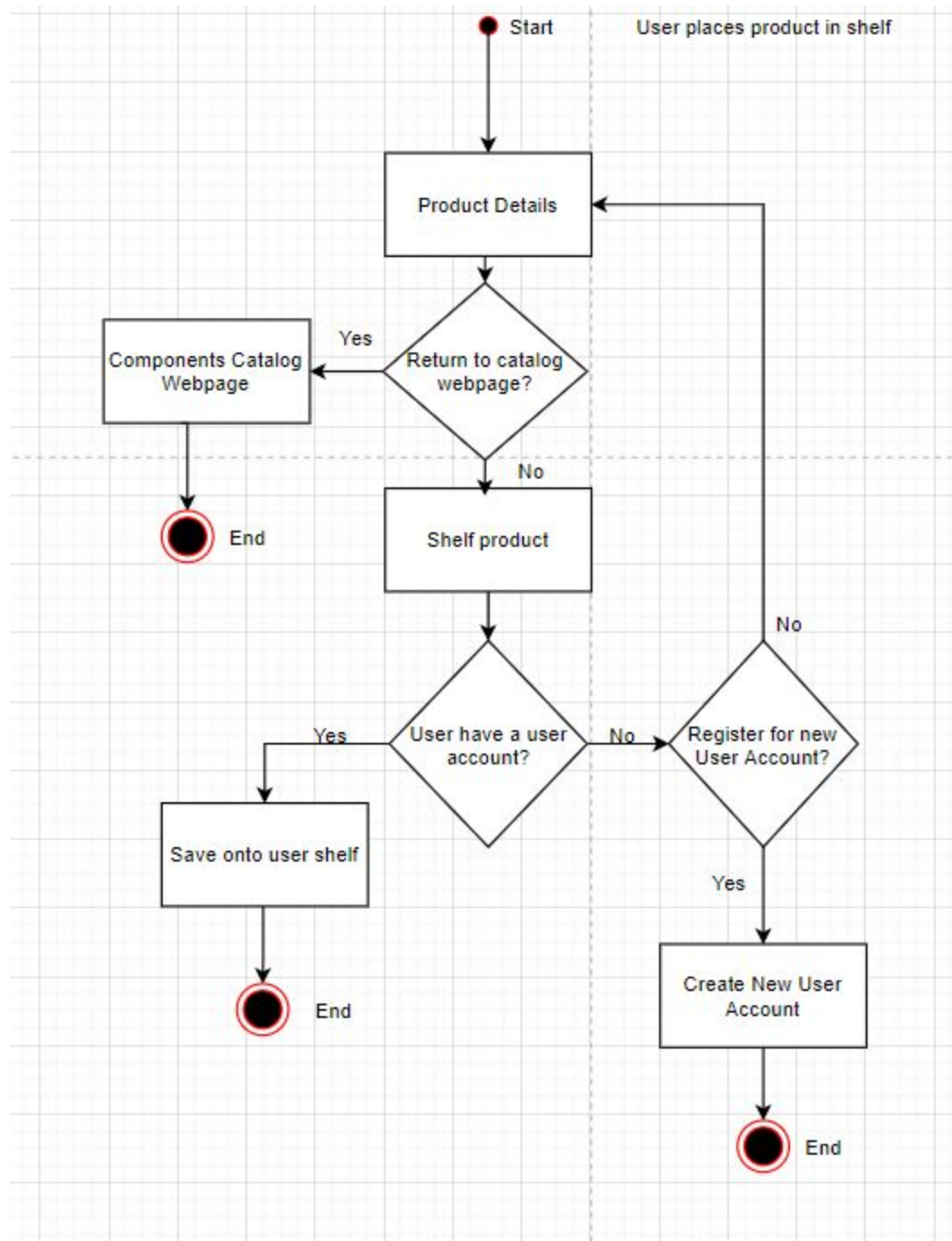
User applies PC upgrader to product



User visits product from vendor's website



User places product in shelf



4.6 Data Store Access

Objectives:

A datastore is a persistent form of storage in which it contains and manipulates records.

Background:

Many operations that are run in the product rely on data for accounts and products from persistent storage. Also, the data such as account data is changed, and those changes have to be reflected in a persistent form. A datastore is utilized to perform the functions of adding, removing, and updating records.

Requirements:

- Be able to access records
- Be able to add records
- Be able to remove records
- Be able to update records

Non-Functional Requirements:

- Should not take longer than thirty seconds

Success:

- Records are dealt with accordingly

Failure:

- Logging and potential timeout

4.6.1 Functional

Req#	Priority	Description	Rationale
General / Base Functionality			
FR-DSA-G-001	1	Add a record	Add a new record to the datastore
FR-DSA-G-002	1	Delete a record	Enables removal of a record
FR-DSA-G-003	1	Update a record	Whenever an attribute of the record has changed, it is imperative to update the record.

4.6.2 Non-Functional

ID	Requirement
NFR-DSA-001	Should not take longer than thirty seconds

4.7 Error Handling

Objectives:

- To prevent users from seeing any failures in the system.
- If failure occurs only relevant systems will be impacted, and for a limited time.
- Recovery must be swift, and users should be impacted as little as possible.

Requirements:

- The site will not crash.
- The site will not become unresponsive.
- No action should cause a system failure. (Closing browser or crashing computer)
- All errors must be logged. (Excluding errors caused by logging)
- All errors must be handled.
- Users will be informed of an error.

Non-functional Requirements:

- Error responses should be lightweight, meaning not much programmatic execution.
- Errors must be handled and reported quickly and concisely to prevent user confusion or frustration.

Success: Fast, responsive, and informative handling that is clear to the user.

Failure: Error handling is slow, confusing, and/or computing intensive, resulting in customer loss and potential computer freezing/crashing.

4.7.1 Functional

Req#	Priority	Description	Rationale
General / Base Functionality			
FR-EE-G-001	1	The site will not crash or become unresponsive.	An unresponsive product is not desirable to anyone. Likewise a product that crashes your browser or computer is equally, or moreso, unwanted.
FR-EE-G-002	1	All errors must be handled.	All exceptions must be caught during runtime.
FR-EE-G-003	1	Users will be informed of an error.	Users who understand that an error occurred, and what error occurred, are more able to address, or cope with, the specifics of an error.

Security Requirements			
FR-EE-S-001	1	Errors must be handled without revealing secure information.	Information vital to maintaining security, such as the users password, or other compromising data should not be shown or logged.

Usability Requirements			
FR-EE-U-001	1	Error handling must be swift and light weight.	We want the user to experience as few errors as possible, but when they occur we want the user to be impacted as little as possible.

Audit Requirements			
FR-EE-A-001	1	All errors will be logged and dated, excluding log errors.	We want to know about all failures and where they happened, however, if we log a log failure, and that subsequent log fails to log that could create an infinite loop of failure logs.

4.7.2 Non-Functional

ID	Requirement
NFR-EE-001	Users should not be impacted when an error occurs.
NFR-EE-002	Errors should be caught and handled in less than 5 seconds.

4.8 Logging

Objectives:

- Log all user activity including:
 - Logins, successful and unsuccessful attempts
 - Logouts
 - All user posts
 - All user actions
 - All errors

Background:

Logging user activity and errors will provide proper feedback and knowledge for improving the web app as well as finding and correcting common errors in code. Logins will be logged to prevent an infinite loop of login attempts as well as potentially blocking a user from having unlimited login attempts. Logging errors is essential for fixing issues as soon as possible.

Requirements:

- Log successful sign-in attempts.
- Log unsuccessful sign-in attempts.
- Log sign-outs.
- Log all user posts.
- Log all user actions on the web app.
- Log all errors.
- Privacy policy needs to reflect that, and which, user actions are being logged.

Non-Functional Requirements:

- All logs will have date and time stamps.

Success: All user actions and errors will be successfully logged and available for viewing and assessment.

Failure: Not all user actions or errors will be logged, leaving a black-area of things that might not be able to be assessed for fixing or addressing.

4.8.1 Functional

Req#	Priority	Description	Rationale
General / Base Functionality			
FR-LOG-G-001	1	Log all errors	So that developers are able to address errors and fix them in a timely manner.
FR-LOG-G-002	1	Log all user actions	To track user activity and see common trends.

Security Requirements			
FR-LOG-S-001	1	Log all login attempts	This will help distinguish potential login attackers and potentially ban them by IP.

4.9 Login

Objectives:

- The user will also be able to logout from a logged in state.
- If there is no activity after 30 minutes a user will be logged out automatically.
- Send logouts to be logged.

Background:

Logging out is an essential component of most web apps, this will allow users to logout for security purposes and protect potentially sensitive information, a login must have a logout to compliment it.

Requirements:

- Users will be able to logout.
- Log all logouts.

Non-Functional Requirements:

- Users will be automatically logged out if idle for longer than 30 minutes.

Success: User is able to logout

Failure: User is unable to login, given the proper credentials.

4.9.1 Functional

Req#	Priority	Description	Rationale
General / Base Functionality			
FR-LI-G-001	1	User can log in with credentials	Cannot log in without credentials, there needs to be a login interface to distinguish between accounts and allow access of particular features for accounts.

Security Requirements

FR-LI-S-001	1	Verify credentials and passwords	So that users can not easily be hacked or compromised.
-------------	---	----------------------------------	--

Usability Requirements

FR-LI-U-001	1	User interface will provide an easy and clear-cut way for users to sign into their accounts	So the user does not become confused on what they must do.
-------------	---	---	--

Audit Requirements

FR-LI-A-001	1	Logins will be logged	This is to fulfill the logging section of the BRD.
-------------	---	-----------------------	--

Login Activity Diagram

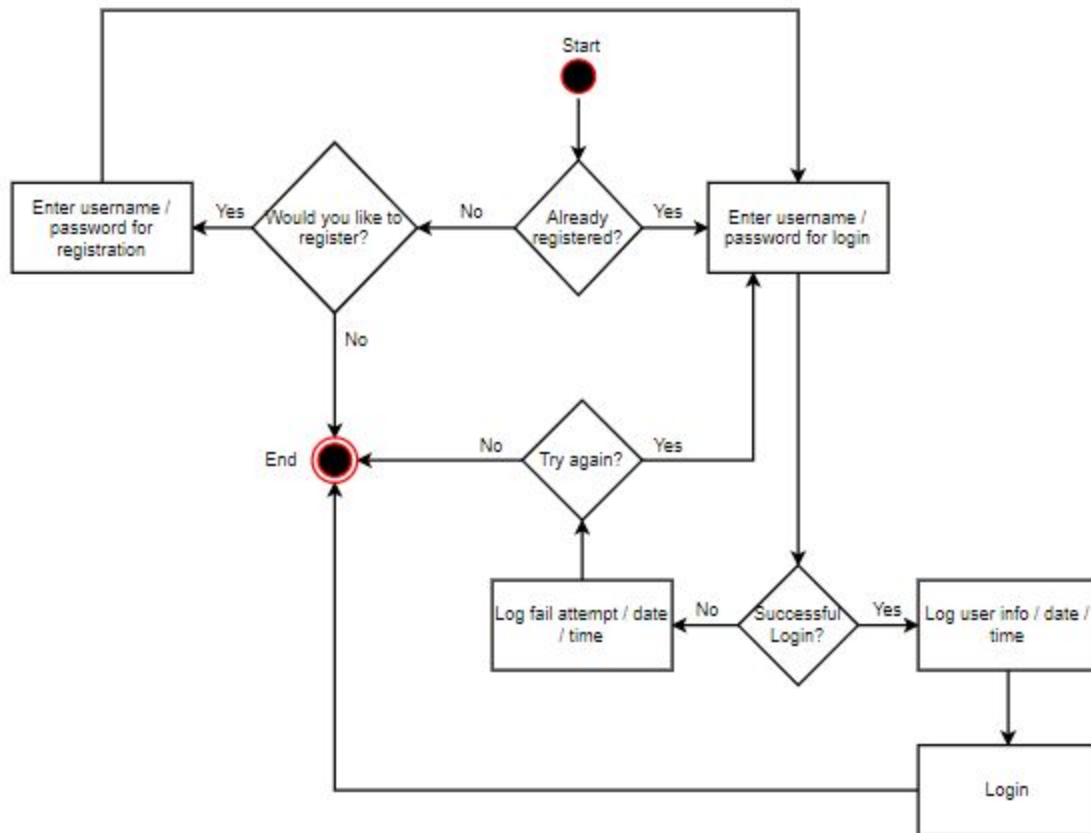


Fig.1 Activity Diagram that shows a user logging in.

4.10 Logout

Objectives:

- The user will also be able to logout from a logged in state.
- Send logouts to be logged.

Background:

Logging out is an essential component of most web apps, this will allow users to logout for security purposes and protect potentially sensitive information, a login must have a logout to compliment it.

Requirements:

- Users will be able to logout.
- Log all logouts.

Non-Functional Requirements:

- Users will be automatically logged out if idle for longer than 30 minutes.

Success: User is able to log out.

Failure: User is unable to logout or there is no option to logout.

4.10.1 Functional

Req#	Priority	Description	Rationale
General / Base Functionality			
FR-LO-G-001	1	User must be able to logout	System cannot be secure on the user end if a user can login but not logout.

Security Requirements			
FR-LO-S-001	1	Logout must be successful when the user logs out.	The account must be completely logged out, if the user stays logged in, it could cause security issues.

Usability Requirements

FR- LO- U-001	1	Must be an easy and clear way in the user interface for the user to be able to logout.	To minimize user confusion when logging out of a system.
---------------------	---	--	--

Audit Requirements

FR- LO- A-001	1	Logouts must be logged	This is a requirement of Logging that must be fulfilled.
---------------------	---	------------------------	--

Logout Activity Diagram

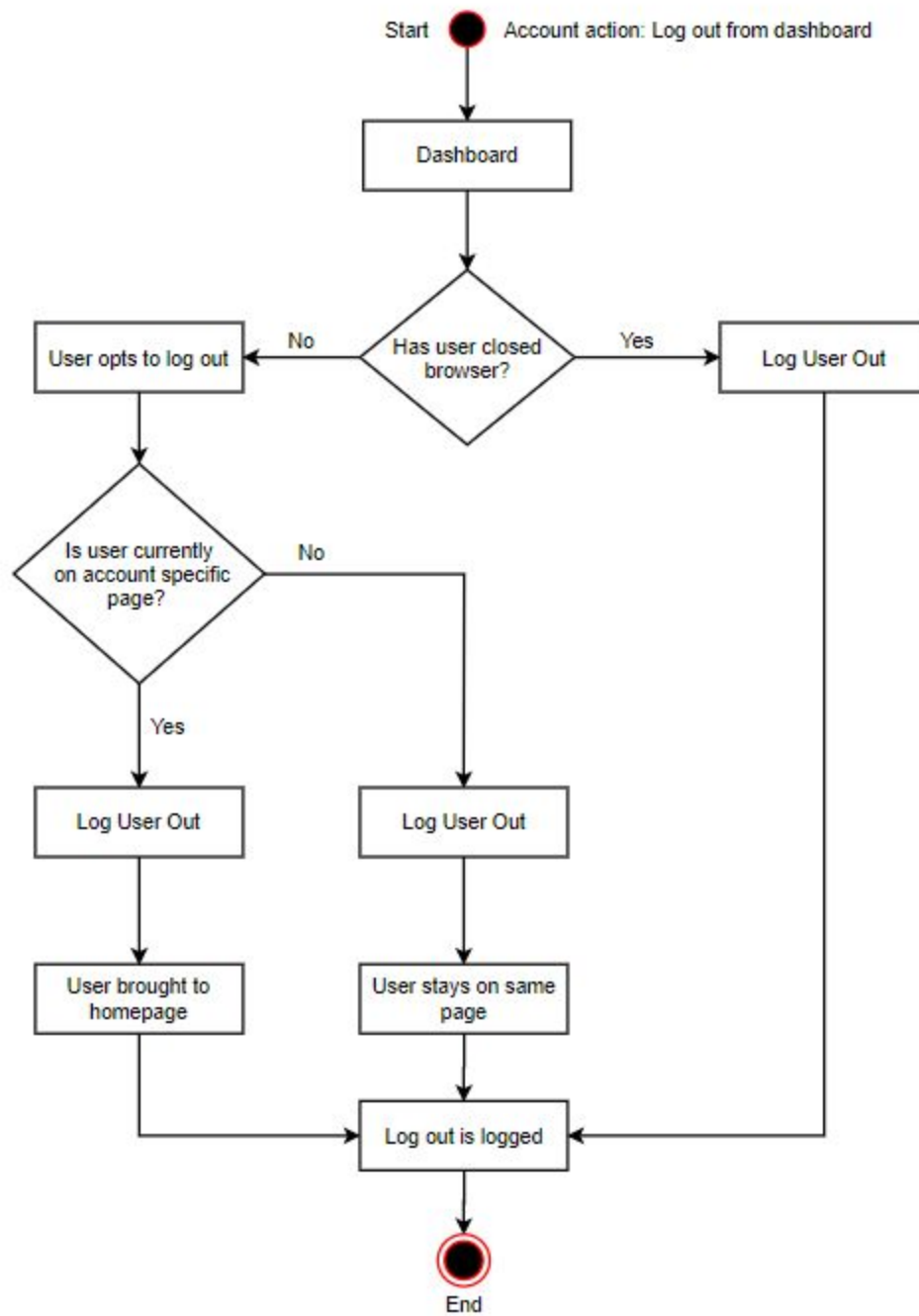


Fig.1 Activity Diagram detailing how user logouts work.

4.11 Most Popular PC Build

Objectives:

- Catalog of the most “popular” builds in the website.
- Users can utilize filters and search to navigate the catalog.
- Users can view the comments made for each build.
- Account holders can publish their own unique builds.
- Account holders can comment on individual builds in the catalog.

Background:

The feature answers the question of how we can get users to use our product. In some cases, users need a starting point to build their pc. We believe that the Most Popular PC Builds is the best answer to this question. By showing users the most popular builds, they have an idea of where to start. The rating and reviews of each popular build would provide information at a glance at optimizing a pc build.

Requirements:

- Provides users a catalog of the most “popular” pc builds.
- Each build is represented by a screenshot and build name.
- Users can view and browse the catalog.
 - Can be filtered by: Gaming, Productivity, and Animation
 - There will be a search function for finding specific published builds.
- Account holders have the option to:
 - Add a build in the catalog to their garage
 - Comment on a build
 - Can publish builds
- “Most Popular Builds” section will be computed to show the most popular builds in descending order by the following:
- Popular builds trending will be organized, in descending or ascending order, and updated according to the number of supporting “likes” the suggested build receives.

Non-Functional Requirements:

- The categorization of the builds should be clear to the user.
- The flow of information of the product representation should be intuitive.
- The screenshot should have a resolution of at least 72 DPI.
- Filters should be highlighted when selected.
- The search bar should have a suggestions drop down to suggest potential products the user might be requesting.
- Starting at three unique builds per day

- Six unique builds per day when a level of notoriety is achieved
- An immutable copy is made, and saved to the websites database, that can no longer be modified.
- Publications are subject to the Autobuild Reputation system.

Success:

Users are able to navigate the popular builds webpage by utilizing filters and search. Account users are able to save a popular build to their garage and comment on builds on the webpage. Also, account users are also able to publish a set number of builds depending on their notoriety to the popular builds webpage.

Failure:

Users are unable to navigate the popular builds webpage through filters and search. Account users are unable to save the desired build to their garage. They are unable to comment on the desired build. Account users are unable to publish their builds, can publish more than the maximum limit, or publish the wrong build from their desired build.

4.11.1 Functional

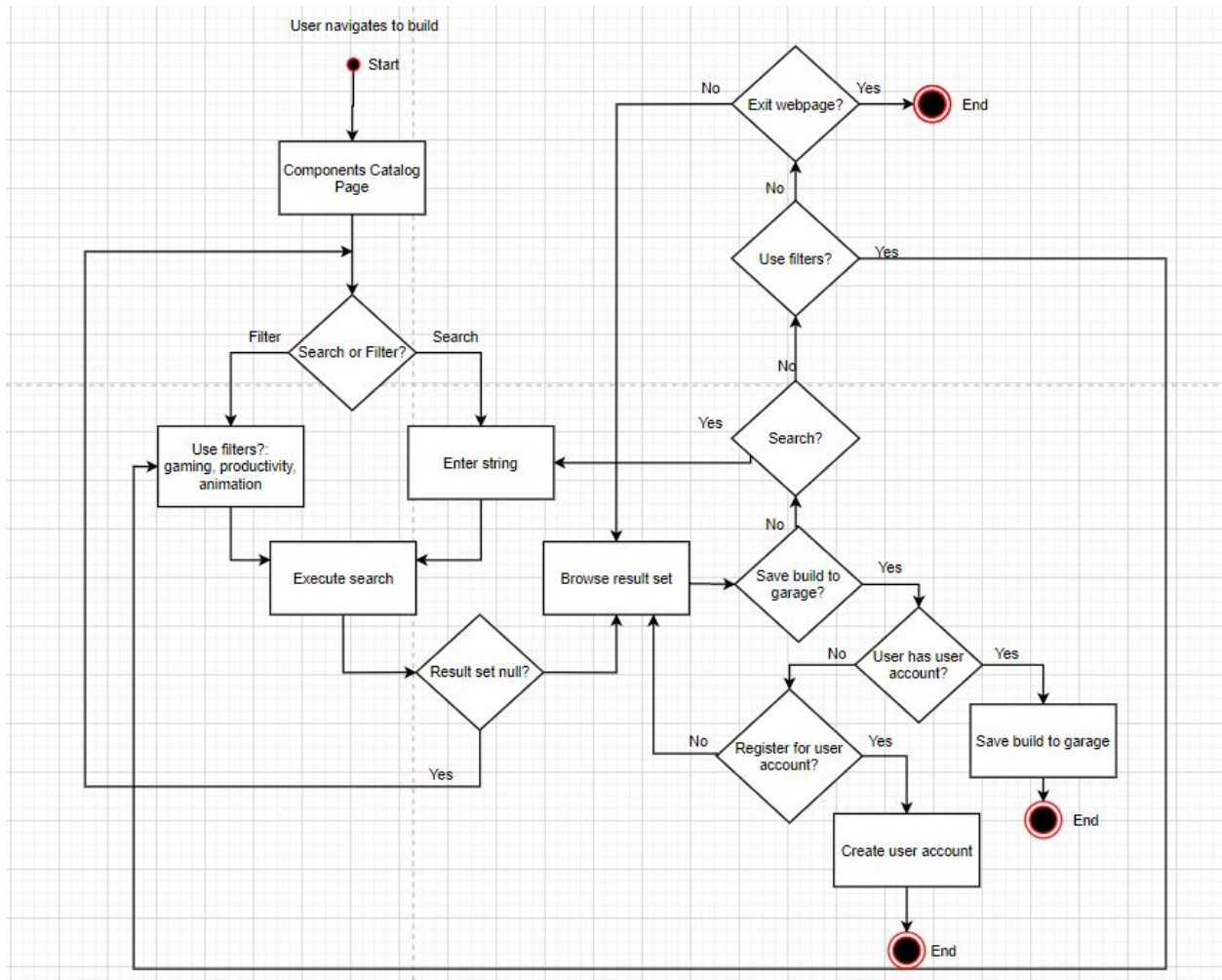
Req#	Priority	Description	Rationale
General / Base Functionality			
FR-MPB-G-001	1	The catalog can be filtered by: Gaming, Productivity, and Animation	Users mostly searched for pc builds based on one or more of these criteria.
FR-MPB-G-002	1	There will be a search function for finding specific published builds.	If a user knows the full or partial name of the build, he/she can execute the search based on the search string.
FR-MPB-G-003	1	Account holders can publish their own unique builds	This is what populates the most popular pc builds webpage
FR-MPB-G-004	3	Account holders can post comments on individual builds	This provides feedback to the account holder about the build
FR-MPB-G-005	1	Account holders can save a build to their garage	Allows account holders to save the build and potentially tweak the design
FR-MPB-G-006	3	Published builds that do not gain enough recognition ("likes", views, saves or favorites) within 30 days will be removed.	This is so that builds that do not carry much traction do not sit in the webpage for long.

4.11.2 Non-Functional

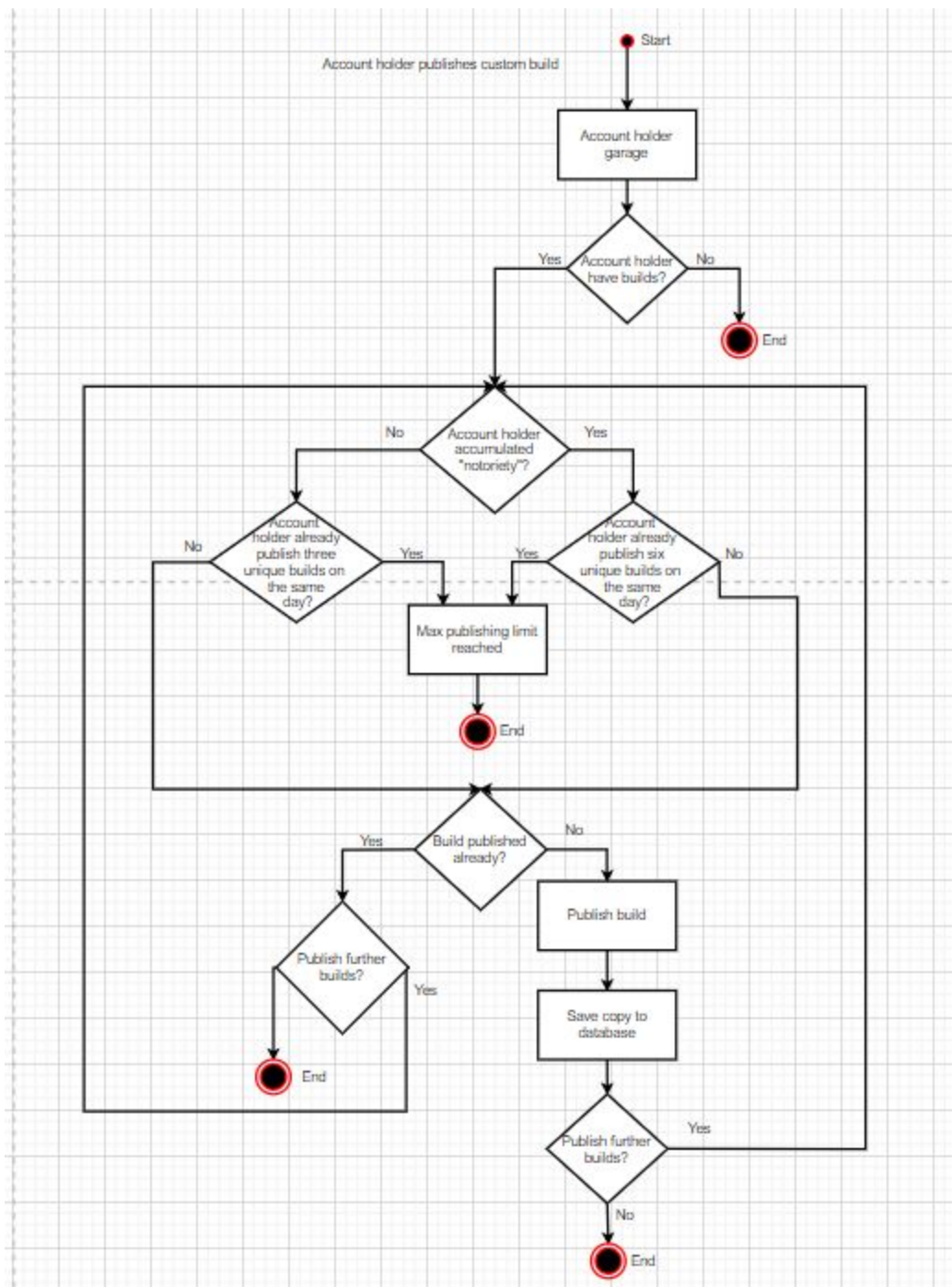
ID	Requirement
NFR-MPB-001	The categorization of products should be clear to the user.
NFR-MPB-002	The flow of information of the product representation should be intuitive.
NFR-MPB-003	The screenshot should have a resolution of at least 72 DPI.
NFR-MPB-004	Filters should be highlighted when selected.
NFR-MPB-005	Account holders are allowed a maximum of publishing three unique builds per day
NFR-MPB-006	Six unique builds per day when a level of notoriety is achieved
NFR-MPB-007	When a build is published, an immutable copy is made, and saved to the websites database, that can no longer be modified.
NFR-MPB-008	Publications are subject to the Autobuild Reputation system.

4.11.3 Most Popular Build Activity Diagrams

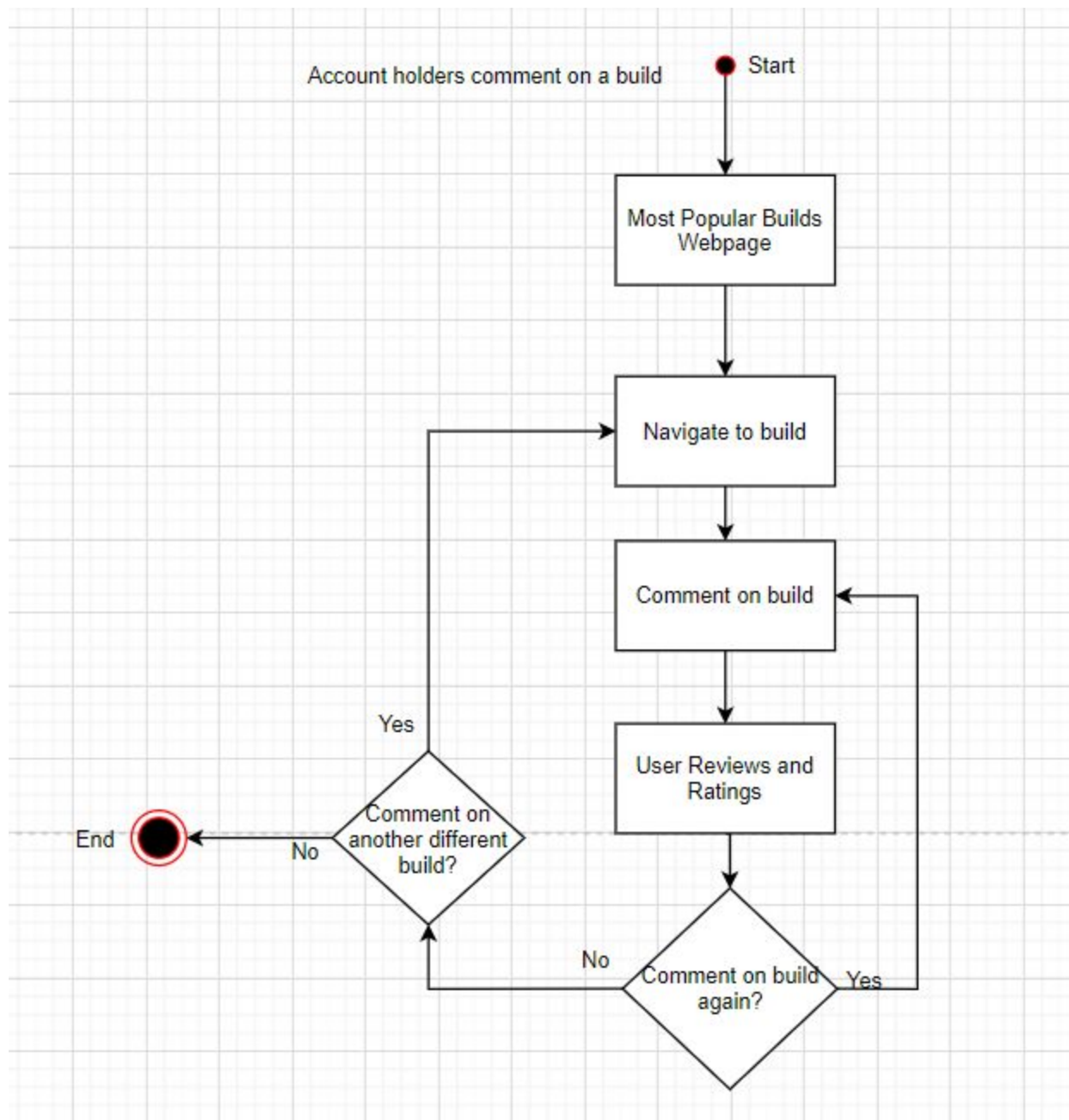
User navigates to build



Account holder publishes custom build



Account holders comment on build



4.12 Network Communication

Objectives

In order to make our application available to the public we need network communication. Since we will be using the out of process hosting model (see Tech document: Web Server section for more reference) we will be using a proxy server between the public internet and our application.

- Proxy server also has SSL certifications which means websites can keep user data secure, verify ownership of the website, prevent attackers from creating a fake version of the site, and gain user trust. TLS/SSL certification is essentially digital proof of a site's identity.
- Our web server will accept HTTPs requests on port 443
- The web server will relay the information on port 80 HTTP/2
- The web server offers security from the public network by:
 - Limiting surface area exposure
- The proxy web server also offers privacy benefits, by keeping the applications IP address and other responses made to incoming traffic hidden.
- The proxy server also supports protocol Simple Mail Transfer Protocol ([SMTP](#)), to send and receive emails.
- Web servers allow the show a logging to view the visitors accessing the web server.
- Web servers also can enable firewalls to ensure that only valid packets are received/ accepted.
- We also want Load balancing across multiple application instances to optimize resource utilization, maximizing throughput, and reducing latency.
- TCP and UDP connections will be utilized.

Background:

Network communication is what allows for seamless communication across a network, accommodating for security concern, high throughput, authorization features, specific ports to have open for use, and etc. Most of these concerns will be attainable via our use of a proxy server that lies between our application and the internet.

HTTPS is a secure communications channel that is used to exchange information between a client computer and a server. It uses Secure Sockets Layer (SSL).

Requirements:

- Enable SSL certification
- Keep logs of requests coming in from the internet via the proxy server.
- HTTP/2
- TCP and UDP
- Authentication

Non-Functional Requirements:

- Configure on NGINX and IIS to use SSL/HTTPS
- Enable the use of HTTP/2
- Configure TCP (or UDP) for load balancing on web server
- Enable Authentication via the web server for application.
 - There are often sections of the site that have restricted access to. Our application can provide our own authentication and authorization methods, but the web server itself can be used to restrict access if these are inadequate or unavailable.

Success: Valid network connection has been established between client and server and is secure.

Failure: Network connection is compromised or not established.

4.12.1 Functional

Req#	Priority	Description	Rationale
General / Base Functionality			
FR-NC-G-001	1	HTTP/2	HTTP/2, a persistent connection can be used to service multiple simultaneous requests. This allows for an improvement of efficiency of HTTP over the internet.

Security Requirements			
FR-NC-S-001	1	SSL/ TLS	This verifies ownership of the website, prevent attackers from creating a fake version of the Autobuild site
FR-NC-S-002	1	Authentication	A web server administrator protects a section of a web site that needs access via authentication.

Audit Requirements			
FR-NC-A-001	1	Logging Requests to and from the server	This is important to draw out a baseline of how requests to the application are at its best, and on average.

4.12.2 Non-Functional

ID	Requirement
NFR-NC-001	Configuration of web server

4.13 PC Builder

Objectives:

- Allow users to select if they would like to use the “Builder” or “Upgrader” in this calculation, through a toggle switch at the top of the page.
- Offers a customizable build based off of user preferences entered within the PC builder.
- Select filters to fine tune build type:
 - Allow users to select between basic, gaming, or graphic intensive. This criteria will determine which components to prioritize
 - Peripherals they want included in their search
 - Type of PSU whether it be non-, fully-, or semi-modular.
 - Hard drive type(s) and how many they want.
 - Input their desired maximum price
- The PC builder uses comparison between data under the following fields in order to provide users with an appropriate choice that best fits their wants/needs:
 - CPU
 - Number of cores
 - Instruction set
 - CPUs base clock speed and its boosted, or overclocked clock speed.
 - CPU cooler: selecting an air cooler with a heat sink or a liquid cooler for their build.
 - Air coolers compare:
 - Fan size
 - RPMs
 - The size of the heatsink
 - dBA output
 - Liquid coolers compare:
 - Fan size
 - RPM
 - dBA output
 - RAM
 - RAM's type (DDR2,DDR3, etc.)
 - Bandwidth
 - Transfer speed
 - Latency
 - Timings
 - Capacity.
 - GPU
 - GPU's base and boosted clock speeds.

- GPU's memory and memory bandwidth. It will also compare the resolution and framerate that it can support.
- Motherboard selection will be based on compatibility
 - (RAM type,CPU support, PCI ports) , chipset, maximum RAM support, and ethernet throughput.
 - comparing number of PCIe slots, GPU slots, and SATA SSD slots available.
- PSU
 - Fully-modular, semi-modular or non-modular
- Hard Drive
 - Spinning hard disk drive (HDD)
 - Solid state drive (SSD)
 - Other SSD types.
 - Comparisons:
 - Storage capacity
 - Read/write speeds
 - Longevity(how many reads/writes it is expected to handle in its lifetime)
 - Disk RPM, if applicable (which will affect access speeds as well as noise).
- Case - The user will be given the choice of whether they want the case to be organized by the inner-workings (ie: number of hard drive slots and motherboard type) or by the physical dimensions of the case.

Background:

The PC Builder is the unique feature that spawned the idea of this website. It will request several parameters from the user to offer them a recommendation on a complete computer system to purchase, made from individual components, that the user will assemble themselves.

Requirements:

- Allows users that choose to use the builder to find a quality recommended system given their desired parameters.
- Create an algorithm that will compare the many benchmarking data (technical aspects of each component and filtering choices made to allow builders to output accurate listings).

Non-Functional Requirements:

- A log will be made when a registered user either builds or upgrades a computer.
- Time taken for the algorithm to output the list of build should take no more than 5 sec

Success: Users will be recommended a credible series of products for purchase and use. Providing users with a simple resource to come up with quick powerful builds within their price limitations.

Failure: Users will receive bad recommendations, resulting in a potential loss of thousands of dollars, and damaging the sites credibility. This would mean the primary feature of the product's original design will be unusable, resulting in a loss of uniqueness, turning our site into a standard catalog of little interest.

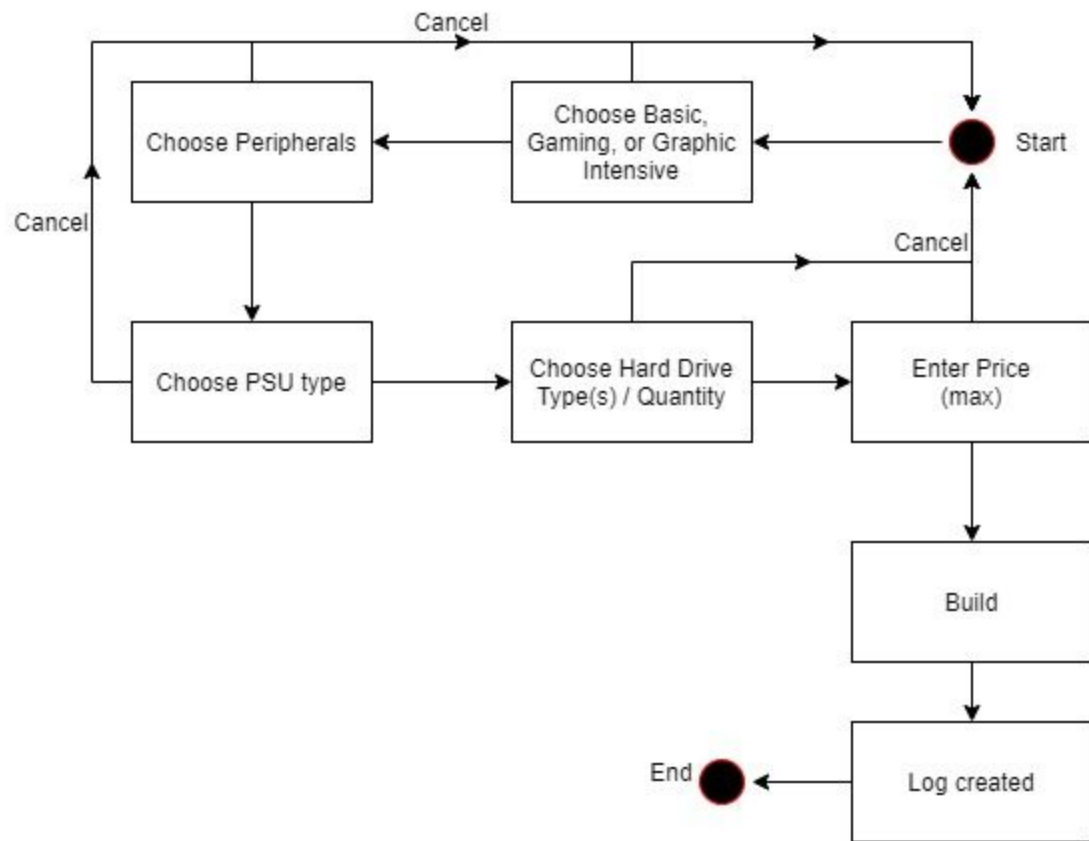
4.13.1 Functional

Req#	Priority	Description	Rationale
General / Base Functionality			
FR-PB-G-001	1	Builder algorithm	In order for the upgrader to output a build that uses vendor product data an algorithm needs implementation
FR-PB-G-002	1	Provide filters for users to define build.	Without filters the user will just be given an arbitrary build.
FR-PB-G-003	1	Provide a toggle switch to allow users to use the upgrader instead.	The upgrader and builder will operate in generally the same area, so we put them together with a toggle switch.

4.13.2 Non-Functional

ID	Requirement
NFR-PB-001	The time taken by the PC builder to output a build should take no more than 5 sec
NFR-PB-001	The many filtering options in builder shows a great importance in the seamless navigation of all filters and descriptions

4.13.3 PC Builder Diagram



4.14 PC Upgrader

Objectives:

- To provide users with the ability to upgrade individual components, or an entire system, in place of using the PC Builder tool.
- To help users find upgrades to computer components, including things such as laptop batteries.
- If the user is registered and logged in, they will be given the option to choose and import a previously saved build from their garage. They can also choose to manually enter the components of their build.
- If it is an unregistered user they will have to manually enter each component of the build.
- The user will select which component(s) they want to upgrade.
 - If the PSU or hard drive is selected they will have to select which type of PSU and or hard drive they are looking for.
 - The user will input their desired maximum price.
- In either case if there are no results that meet the criteria for the price the user will be asked if they want to put in a new price or start over.

Background:

Originally the builder and upgrader were two separate features, however, with the discovery they operate fairly similarly we decided to fold them into one feature. The requirements are slightly different so we separated them in this document.

Requirements:

- Using the same algorithm as the builder, with a slight variation, quickly generates recommendations for users.
- Allow users to select either individual components or an entire build to upgrade.
- Inform the user of potential upgrades and showing the cost of the new component, as well as several alternatives.
- Give the users a view of the component and its availability.
- Provide detailed specs for users who want them, with comparison.

Non-Functional Requirements:

- Builds must be computed in under 5 seconds.

Success: Users will be given several recommendations of viable upgrades and leave satisfied.

Failure: Users will receive bad recommendation on upgrades, potentially leading to a loss of credibility and user loss.

4.14.1 Functional

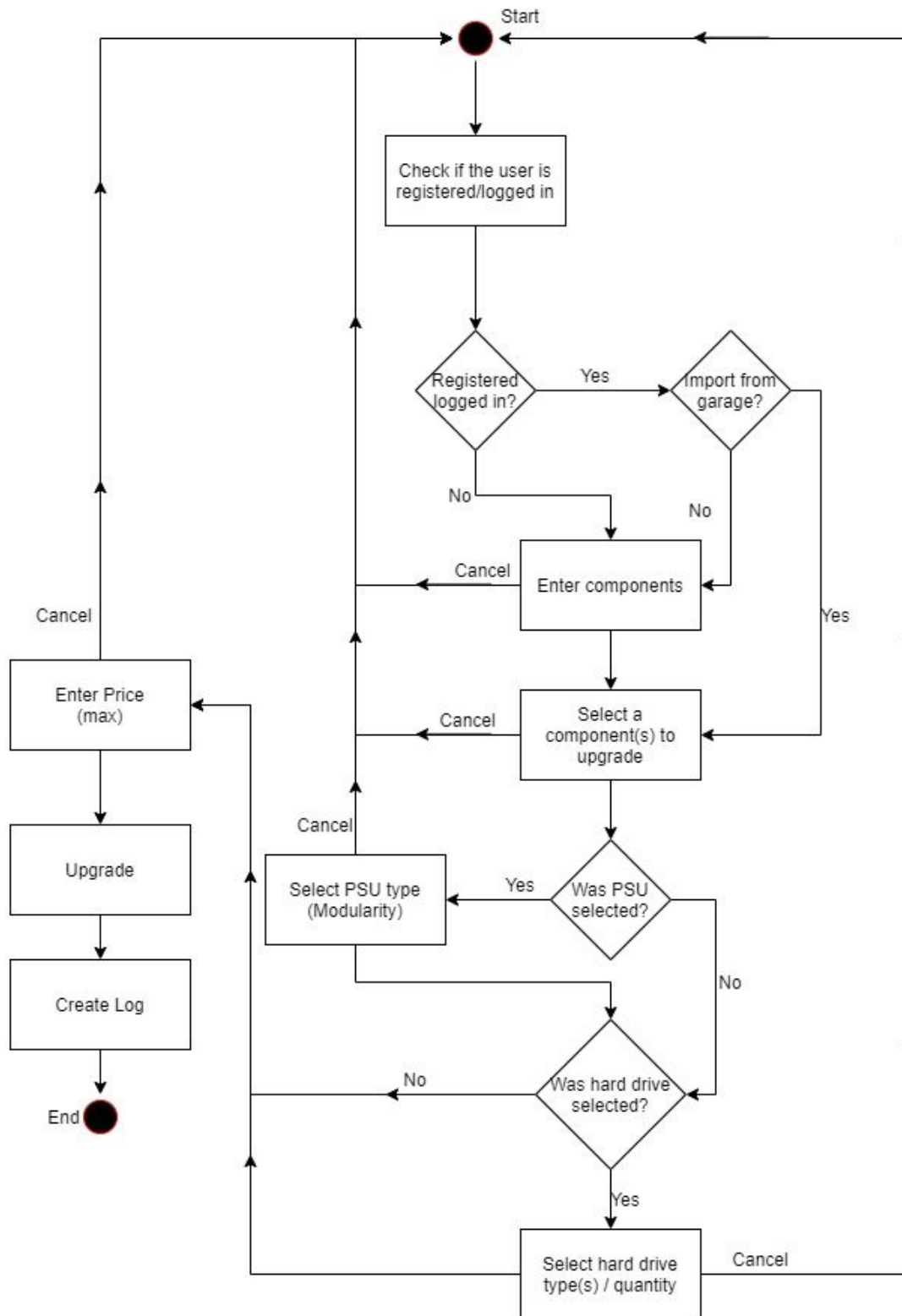
Req#	Priority	Description	Rationale
General / Base Functionality			
FR-PU-G-001	1	Allow users to select either individual components or an entire build to upgrade.	Users will want to customize their build preferences so as to get the most appropriate choice they desire.
FR-PU-G-002	1	Provide detailed specs with comparison.	Comparing multiple products will give the user an idea of where this particular product stands, and why to buy it.
FR-PU-G-003	1	Give the users a view of the component and it's availability.	If a component is not available a user should know.

Usability Requirements			
FR-PU-U-001	1	Inform the user of potential upgrades and showing the cost of the new component, as well as several alternatives.	The ability for a customer to compare is always a plus, it may even show them that our recommendations are accurate.

4.14.2 Non-Functional

ID	Requirement
NFR-PU-001	Builds must be computed in under 5 seconds.

4.14.3 PC Upgrader Diagram



4.15 Privacy

Objectives:

- To adhere to the legal constraints set forth by governments on the gathering of information from our users.
- To anonymize collected data.
- To protect all data about our users as vehemently as possible.
- Offering users the ability to opt-out of data gathering, if user specific data is gathered.
- No personally identifiable information (PII) will be gathered.

Background:

Our product, as a web based product, is subject to many different laws that are constantly changing and being updated. With that in mind we must protect our product, ourselves, and our users by adhering to the laws to the best of our ability, and ensuring users are given the right to the privacy they wish to have.

Requirements:

- Data gathered must be anonymous.
- Users must be able to opt-out of cookies and select which they wish to allow.
- Users must be able to view the privacy policy, displaying what data we collect, what we will do with that information, and who we will share it with (if anyone).
- Follow the 7 principles of the EU Data Protection Directive.
 - **Notice:** users should be given notice when their data is being collected
 - **Purpose:** data should only be used for what you say you will use it for
 - **Consent:** user data should not be shared without your users' consent
 - **Security:** collected data should be kept secure
 - **Disclosure:** users should be informed about who is collecting their data
 - **Access:** users should be allowed to access their data and make corrections to any inaccurate data
 - **Accountability:** users should have a method available to them to hold data collectors accountable for not following the above principles
- Provide a clear privacy policy that will be displayed as a standalone document, as per CalOPPA.

Non-Functional Requirements:

- Users will only be prompted on each visit if they have not granted or denied permissions.

Success: Adherence to legal requirements, as well as building trust with our users.

Failure: Legal action could be taken against the developers, resulting in fines or other legal repercussions.

4.16 Registration

Objectives:

- To register an account

Background:

The website is functional without an account; however, functionality is limited. There are features where we require an account. One example is creating a post or commenting on someone's post. We aren't allowing anonymous users to post, so they need an account. Our website has two different accounts for users to choose from: a basic user account and a vendor account. The vendor account is a basic user account; however, they have the vendor functionality, which allows them to add/update/remove items that they want to list.

Requirements:

- To create an account for the user.
- For a basic user account, verification email will be sent to the email provided.

Non-Functional Requirements:

- There is an 8 character minimum which must include a capital letter and at least 1 number.
- All attempts to create an account will be logged.

Success: An account was successfully created.

Failure: The account registration process is confusing or some part of our system breaks which frustrates the user and causes them to leave our website.

4.16.1 Functional

Req#	Priority	Description	Rationale
General / Base Functionality			
FR-REG-G-001	1	An account will be created.	A user needs an account for certain functionality.

Security Requirements			
FR-REG-S-001	1	Since registration will be linked to our database, the text forms that the user submits through must be through a parameterized query.	We are preventing SQL injection attacks. If we don't, anyone can delete information from our database.

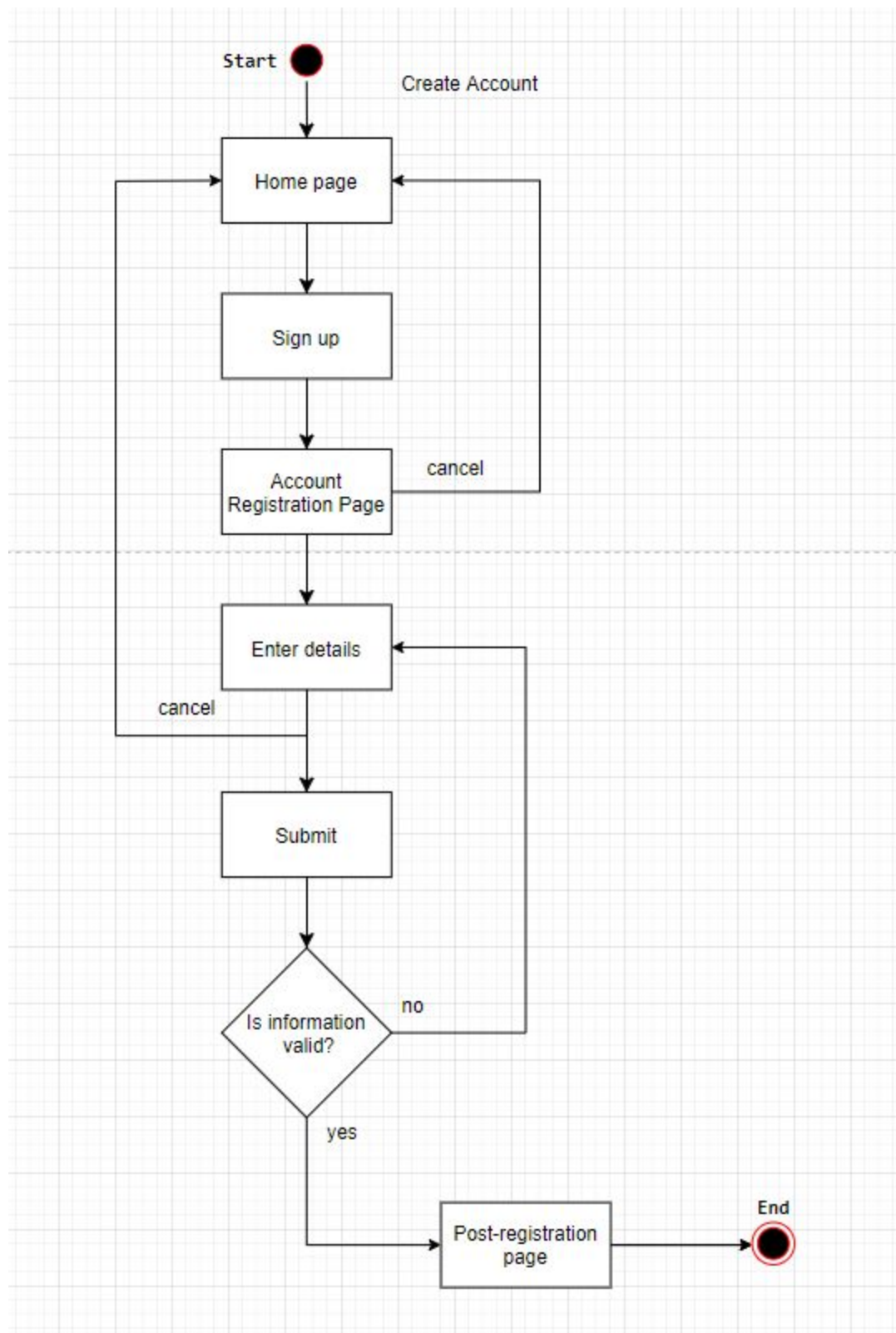
Usability Requirements			
FR-REG-U-001	2	Registration should be smooth and simple.	Account creation is the first step in acquiring a user. Since we want users, we need a smooth UX.

Audit Requirements			
--------------------	--	--	--

FR-REG-A-001	1	All attempts to create an account will be logged.	We want to log all attempts to create an account. User registrations and logins are great places to attempt SQL injection.
--------------	---	---	--

4.16.2 Non-Functional

ID	Requirement
NFR-REG-001	All attempts to create an account will be logged.
NFR-REG-002	There is an 8 character minimum which must include a capital letter and at least 1 number.



4.17 Security

Objectives:

- Follow the NIST Framework
 - Identify risks by analysing:
 - Asset Management
 - Business Environment
 - Governance
 - Risk Assessment
 - Risk Management Strategy
 - Supply Chain Risk Management
 - Protect our organization against threats through:
 - Identity Management and Access Control
 - Awareness and Training
 - Data Security
 - Information Protection Processes and Procedures
 - Maintenance
 - Protective Technology
 - Detect events as they occur by:
 - Anomalies and Events
 - Security Continuous Monitoring
 - Detection Processes
 - Respond rapidly through:
 - Response Planning
 - Communications
 - Analysis
 - Mitigation
 - Improvements
 - Recover lost data:
 - Recovery Planning
 - Improvements
 - Communications
- Be mindful of the OWASP top 10 Web Application Security Risks
 - SQL Injection
 - Broken Authentication
 - Sensitive Data Exposure
 - XML External Entities (XXE)
 - Broken Access Control
 - Security Misconfiguration
 - Cross-Site Scripting XSS
 - Insecure Deserialization

- Using Components with Known Vulnerabilities
- Insufficient Logging & Monitoring
- Use other OWASP and NIST utilities to ward off compromising attacks.

Background:

We want our users to feel secure, and be secure, while using our website. So we will follow be using several security standard to complete that goal, in the hopes of avoiding

Requirements:

- Make sure our website is NIST compliant.
- Observe all OWASP guidelines.

Success: The user information will be secure and the websites

Failure: Website security is compromised, users may lose private information, data from vendors could be modified making them lose credibility, users or vendors could be deleted from the server (if admin account is lost). Full site compromise requiring down time to repair.

4.18 Support Page

4.18.1 Objectives:

- The Q&A forum section will allow account holders to create new threads.
- The Q&A forum section will allow users to choose how many threads to view on each page by the increments (20,50 or 100).
- Accounts with Autobuild can respond to the threads posted by other accounts, therefore creating reply posts to that initial thread.
- Threads posted must be locked and archived after a 30-day time frame has passed with no new additional Responses made.
- Accounts can follow threads made by other accounts (vendors, administrators, developers) and be notified of new posts to that thread through email notification
- The support page will be limited to only the “Q&A forum” section.
- The “Q&A forum section” will have a subsection section for threads.
- The “Q&A Archived forum section” will have a subsection section for archived threads.

4.18.2 Background:

The Autobuild support page is a platform for accounts to post questions and concerns to be viewed by other accounts where they can offer feedback, called threads. This support page consists of a “Q&A” forum where the threads are created.

4.18.3 Requirements: **DO not edit (still subject to change?)**

- A user must have an account registered with Autobuild to be able to create threads onto the “Q&A forum” and post replies to threads.
- Users with no account can only view threads.
- Creating a thread
 - Accounts should be able to create a thread in the forum section
 - Accounts should be able to modify/ edit their threads they created to a forum after posting it.
- Signing up for thread notification
 - Accounts should receive an email notification to threads they followed
 - Accounts can choose a preference. For example: to be notified when a Developer, Admin, Vendor, or other basic users respond/ post to a thread.
- Locked/ archived Threads:
 - Archived threads should be stored for later viewing
 - Accounts should be able to view previously archived threads.
 - Archived threads should be moved to the archived forum section.
- Forum user formatting
 - Accounts should be able to modify the number of threads viewed on the “Q&A forum” per each page by choosing from the specified incrementations of 20,50 or 100 per page

- Accounts can search for specific threads by title in the Q&A forum section
- posting a thread:
 - Character limit for threads posted is 10,000 characters long.
- Thread maintainability
 - Threads in the forum page will be maintained by weekly checks for threads whose date posted have been greater or equal to 30 days and have had no posts made to it in the past three days. Threads that fit this criteria will be archived and saved for later viewing

4.18.4 Non-Functional Requirements:

- User Navigation Per UI/UX
 - The support page should take no more than 5 sec to navigate through to the Q&A section for viewing the threads posted under that section.
 - The threads in the Q&A forums page should be readable preferably “Lato - 12” text
 - Thread titles should overview what the thread will be about
 - Accounts can access a thread for viewing in no more than 4 clicks.
- Scalability
 - Additional functionality will be in place for adding further sections to the support page, available to the administrators and developers should there be a need to expand sections in the support page (“general forum”, “Help forum” etc.).
 - Up to 5 new sections can be added as sub-sections to the support page
- Time to load
 - New threads within a forum should be viewable to all users in no more than 10 seconds
 - Time taken to open and load a thread for viewing should be less than 5 seconds.
- Thread notification
 - Accounts should be notified by email as soon as a post is made to a thread they follow no later than 10 seconds.
- Account restrictions:
 - Accounts have a limited number of threads that they can create to a forum section. Maximum amount is 3 threads for all forums.
 - Accounts have a limited number of posts that they can make to a thread as well, maximum 5 posts.

4.18.5 Success: Functional requirements are implemented as needed are success states unless otherwise which is a failure state.

Failure:

- Creating a thread
 - Non-registered Users cannot create a thread to a forum.
- Signing up for thread notification
 - Non-registered Users cannot sign up to be notified for a thread notification
- posting a thread:
 - Account posts cannot exceed the character limit in the provided text box.
 - Accounts cannot post an empty thread.

4.18.1 Functional

Req#	Priority	Description	Rationale (why?)
General / Base Functionality			
FR-SP-G-001	1	Accounts can create a thread	We want to offer users who have an account more access to the autobuild community.
FR-SP-G-002	1	Forum user formatting	This makes user navigation through the thread easy to follow.

Security Requirements			
FR-SP-S-001	1	Non-registered users can only view	We can think of this as a security measure for prevention of unknown users to have access to edit and post threads.

Usability Requirements			
FR-SP-U-001	3	Signing up for thread notification	This enhances the user's experience, if a user sees a thread that is true for him but there is no response made for it yet, he can follow it!

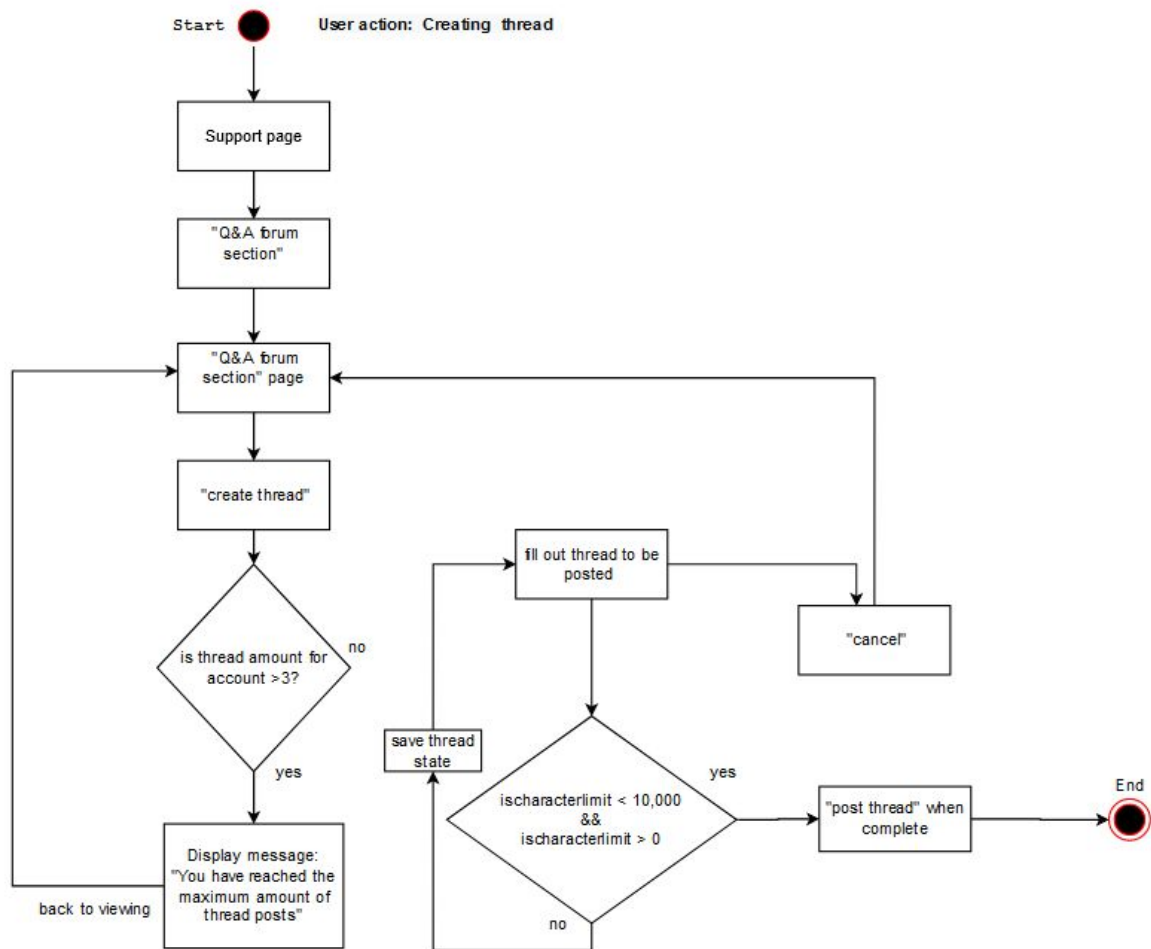
Audit Requirements			
FR-SP-A-001	1	Threads will be locked and archived. + Thread maintainability	This allows the forum page to be less cluttered and allow user readability to be much easier. Therefore, the old thread will be removed from the "active" forum and placed into an archived forum for later viewing.

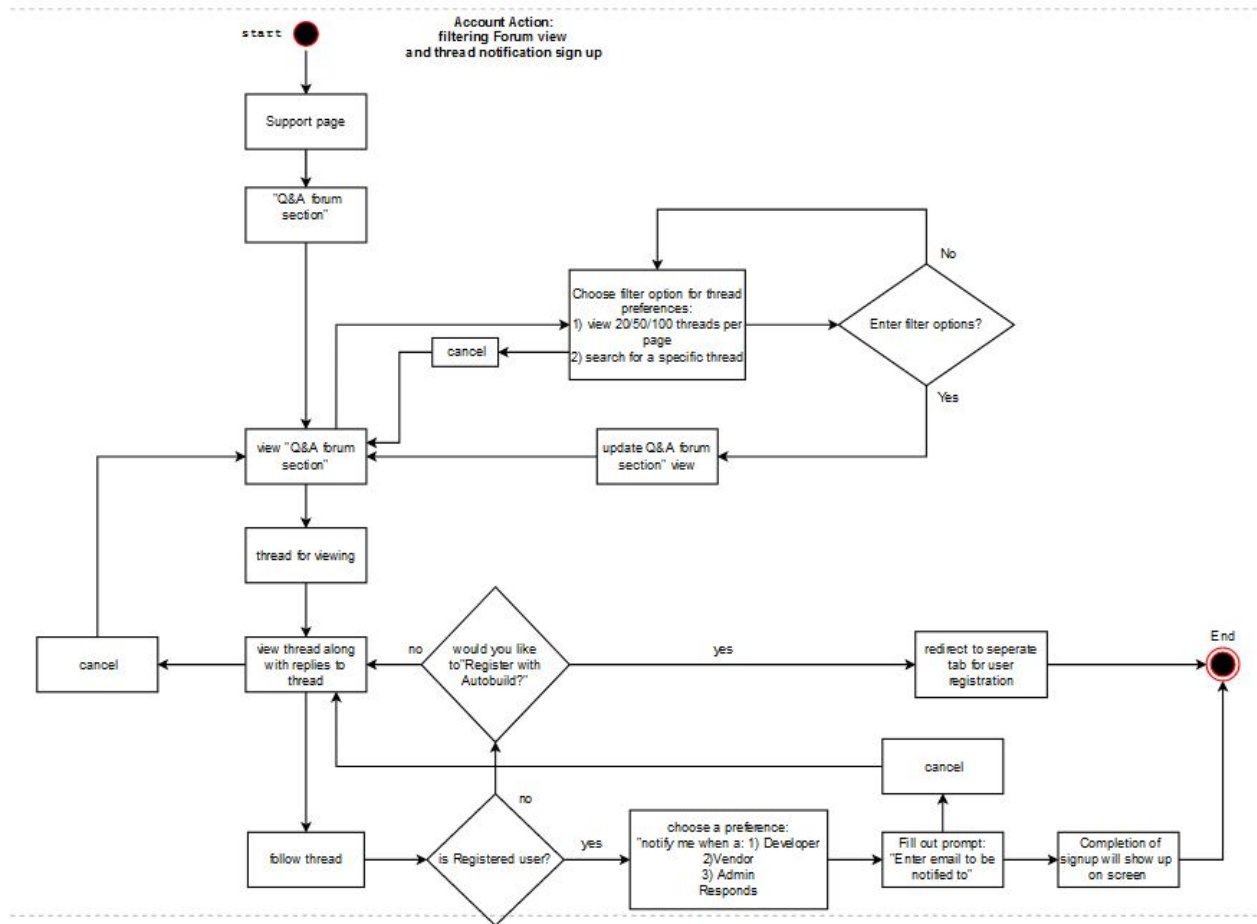
4.18.2 Non-Functional

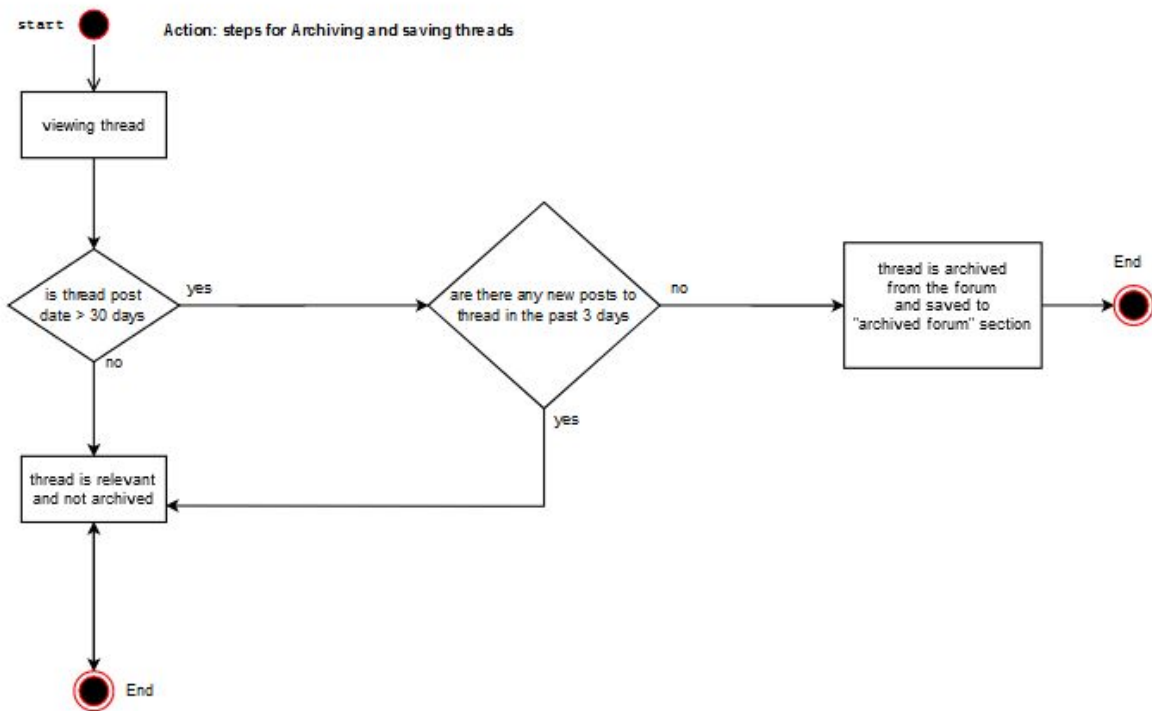
ID	Requirement
NFR-SP-001	The time to load a view, or to get from one session to another should be limited. We do not want the time to exceed 5sec to get to the view the user wants to load.
NFR-SP-002	We don't want to make an application that takes time to navigate through, or a considerable amount of clicks to get to the view the user wishes to see. Maximum number of mouse clicks = 5. Time between views should be no more than 5sec.
NFR-SP-003	Scalability: our support page should be able to accommodate for future expansions on forums. Up to 6 new sections the support page should accommodate for.
NFR-SP-004	Account restrictions: Maximum amount is 3 threads for all forums. limited number of posts that they can make to a thread, maximum 5 posts.

4.18.3 Support Page Activity Diagrams (to be continued)

- Account creates a thread
 - Failure criteria:
 - Account tries to post an empty thread
 - Account tries to exceed number character limit
 - Account follows a forum or thread
 - Account searches or formatted page view







4.19 User Access Control

Objectives:

- Establish a hierarchy of power that grants users different levels of power.
- Give Admin roles master control of the system.
- Give Vendor roles access over their vendor files and information.
- Give Developer roles access to edit web UI and UX files.

Background:

User Access Controls will add a layer of security to our webapp. Establishing a hierarchy of power allows certain users such as admins and developers to have certain powers over other users. This will allow high-level roles to manage system contents as well as enhance user experience by keeping back-end designs hidden for lower level users such as vendors and normal users.

Requirements:

- Give all roles network access.
- Give admin roles all available privileges.
- Give vendor roles access to edit vendor files and user files.
- Give normal user roles access to edit user files.
- Give developer roles access to edit UI and UX files.

Non-Functional Requirements:

Success: An efficient hierarchy will be set in place, that assigns the proper permissions to their respective roles.

Failure: An ineffective hierarchy will be set in place that will clutter and entangle the permissions and their roles.

4.19.1 Functional

Req#	Priority	Description	Rationale
General / Base Functionality			
FR-UAC-G-001	1	Admin roles have all privileges	Changes cannot be made to the system files if a role does not have access to edit system files and make updates/changes.
FR-UAC-G-002	1	Vendor roles can edit vendor files, edit their user files and access networks.	Vendors should be able to list product information as well as their products on out site.
FR-UAC-G-003	1	Developer roles can edit UI/UX files.	Developers have front end website access to be able to make changes to UI and implement or update easy functionality for the user.
FR-UAC-G-004	1	Basic roles can edit user files and access the network	Basic roles should have permission to access their user files and make basic changes to their accounts

Security Requirements			
FR-UAC-S-001	1	Basic role, vendor role and developer role users can not have access to edit System files.	Permissions should not be entangled across accounts, it should be clear which roles get which permissions.

Usability Requirements			
FR-UAC-U-001	3	Signing up for thread notification	This enhances the user's experience, if a user sees a thread that is true for him but there is no response made for it yet, he can follow it!

User Access Controls Diagram

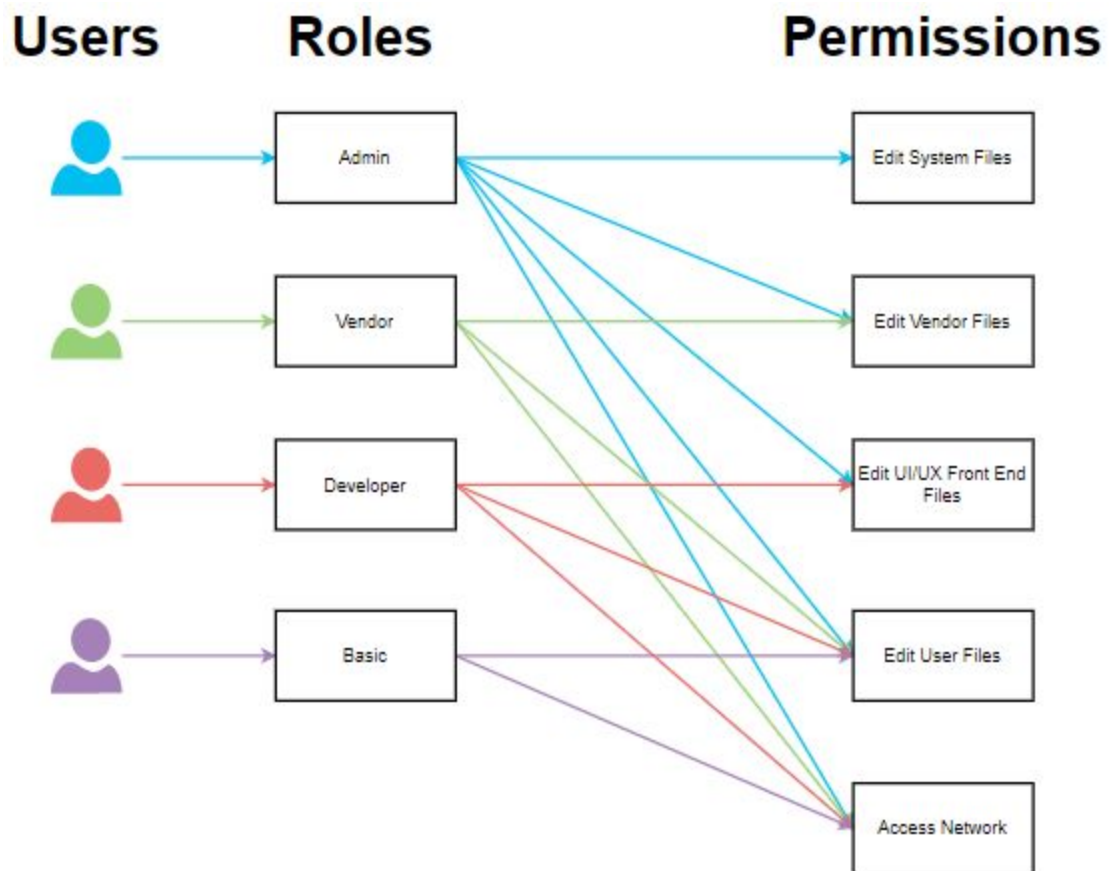


Fig.1 Role-Based Access Control

4.20 User Analysis Dashboard

Objectives:

- Track every user interaction on the system
- a trend of the last 6 months to year
- Show which component was used the most
- How many are registered
 - The user analytics should be affected when the user deletes their account
- Show which Autobuild view was visited most
- How long was the average session of the user (login in to log out})
- Keep anonymity (prevent from identifying the user)
- NO Personal identifiable information PII will be displayed on the dashboard

Background:

The User analysis dashboard offers an admin account an easy to read summary of information regarding Autobuild. It is essentially an overview of the system as a glance.

Requirements:

- The Dashboard is only accessible to the admin account.
- Display the trend of:
 - Show which component was used the most
 - Show which view was visited the most
 - How long was the average session of the user (login in to log out})
 - How many accounts does autobuild hold
 - How many registrations took place date wise
 - All user interaction (“clicks”) will be logged.
- Will not hold any PII

Non-Functional Requirements:

- Admin should have difficulty in navigating through their account to locate the dashboard.
- Data Intuitive graphical understanding of the data.
 - The user analysis dashboard should be easy to follow and understand. For example, Any user who glances at the dashboard should be able to make out the data.
 -

Success: Every interaction and user session is tracked.

Failure: Some user time or sessions slips through the crack, resulting in a faulty analysis.

4.20.1 Functional

Req#	Priority	Description	Rationale
General / Base Functionality			
FR-UAD-G-001	1	Data anonymity	All data represented onto the user analysis dashboard is subject to privacy concerns such as no PII

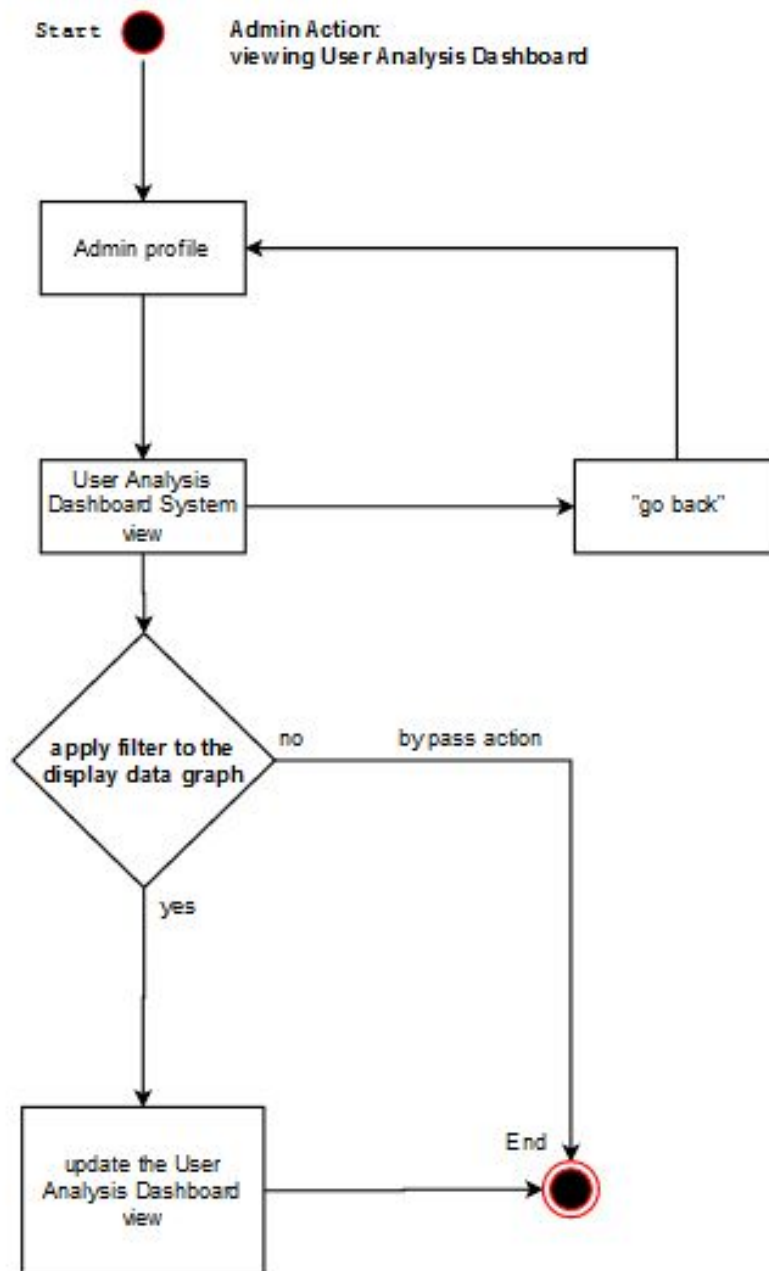
Security Requirements			
FR-UAD-S-001	1	Only admin account can access	This shows the system as a glance reflected by all the logged information. We don't want anyone to access it.

Usability Requirements			
FR-UAD-U-001	1	Admin can filter out which dashboard view to view.	This allows the admin to update the current system as a whole representation to other filters listed user "display trend of"

4.20.2 Non-Functional

ID	Requirement
NFR-UAD-001	The user analysis dashboard should be easy to follow and understand. For example, Any user who glances at the dashboard should be able to make out the data.
NFR-UAD-002	Admin should have difficulty in navigating through their account to locate the dashboard.

4.20.3 User Dashboard Activity Diagrams



4.21 User Experience

Requirements:

- **Useful:** Your content should be original and fulfill a need.
- **Usable:** Site must be easy to use.
- **Desirable:** Image, identity, brand, and other design elements are used to evoke emotion and appreciation.
- **Findable:** Content needs to be navigable and locatable onsite and offsite.
- **Accessible:** Content needs to be accessible to people with disabilities.
- **Credible:** Users must trust and believe what you tell them.

Success: A pleasant experience for users.

Failure: Disinterest in further use of the product. Revisions will be required.

4.21.1 Functional

Req#	Priority	Description	Rationale
General / Base Functionality			
FR-UX-G-001	1	Users should find the content of the website useful.	Users that do not find the content useful will not return to use it in or promote its use.
FR-UX-G-002	1	The site needs to be easy to navigate.	Users that do not find the site easy to use will not return to use it or promote its use.
FR-UX-G-003	2	Users should find the site attractive, and interesting to use.	Users who enjoy using the site, even for simple fun, will continue to drive and promote future users.
FR-UX-G-004	1	The content on the site must be easy to locate.	The faster a user can locate content the more content that can be used.

FR-UX-G-005	1	Disabled users must have the ability to access all content.	Without supporting those with disabilities our user base takes an immediate loss.
FR-UX-G-006	1	Content must be reliable.	Users are coming to us for recommendations on potentially thousands of dollars in expenditures.

4.22 User's Garage

Objectives:

- Create limited storage space for only registered users, lovingly titled after the account holder, e.g. "Nick's Garage".
- Provides account holders with a gateway to products or builds they "save" to their for later viewing.
- Allow users to search through those saved products or builds through filtering and/or a search box.
- Displays builds names and partial details about those builds.
- Has multiple storage areas for components referred to as a "Shelf", these shelves function like wishlists, where users can browse and store items they wish to examine further.
- Shows users a list of all saved builds.
- Shows a list of all saved individual items that are on the shelf.

Background:

The User Garage will function as the users storage space, hence the name "Garage". This storage space will allow users to put together and store build ideas, for later viewing or sharing. These builds can be published for further review by the members of the site, giving users some level of renown if their builds become popular enough. This storage space can be hidden from all users and used as a private space to store ideas, or linked to to show off all the users great ideas.

Users will be able to duplicate other builds and save them to their own space for modification. Allowing for minor tweaks, or even the use of the upgrade tool to find the next best things for an old build that needs some upgrades.

Requirements:

- If the user is not logged in, after selecting the User's Garage, the Garage's usual screen will be replaced by a login prompt that will, after successful login or registration, show the Garage.
- On the left hand side of the screen will be a search bar, filters, and a list of "Shelves" to help find or organize the builds.
- Some of the filters will include:
 - Stars rating (A published or a saved build)
 - Date created
 - Date last modified
 - A check box to show only builds that have images
 - A check box for "My creations" which will only show builds generated or modified by the logged in user.
- The ability to sort builds or parts in ascending or descending order by each relevant criteria.

Non-Functional Requirements:

- The list and filters will load in under 2 seconds on a standard internet connection.
- Filters will remain highlighted after selection as additional filters are added or unhighlighting as they are removed.
- The search bar will save the last 5 searches in the same session for quick searching.
- The user's name will be placed in front of the "Garage" in the interface (____'s Garage).
- The search bar will contain most recent queries until the user leaves the site.
- All sorts will be done in 2 seconds or less.

Success: A comfortable experience with a spacious storage space that invokes the user with a feeling of working in their own garage or shop.

Failure: Users will become frustrated and not desire to login to use the feature, resulting in fewer registered users using the feature and less generated content for the community, since only account holders can publish builds to the web page.

4.22.1 Functional

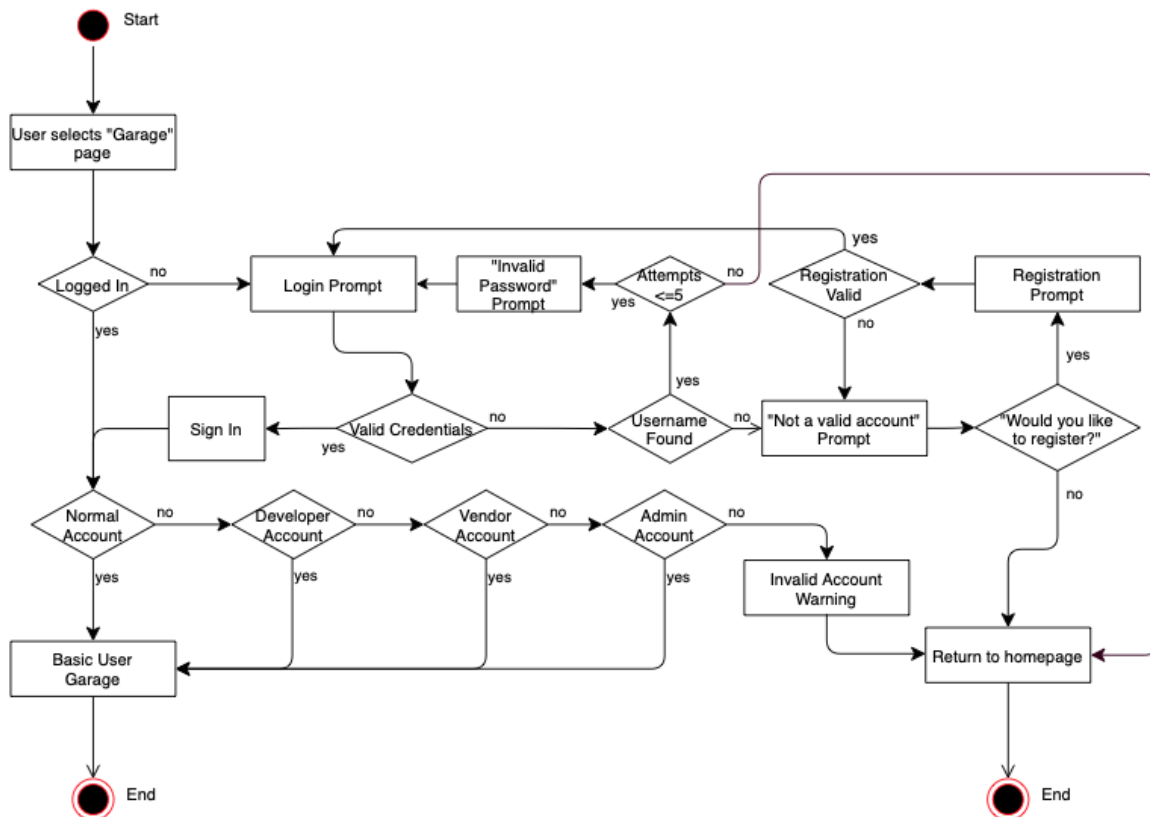
Req#	Priority	Description	Rationale
General / Base Functionality			
FR-UG-G-001	2	Users will be able to login if not already logged in.	The user garage requires users to be registered and logged in to perform any of its functionality, to ease that we will prompt them here.
FR-UG-G-002	2	Users will be able to sort data in the garage.	Being able to sort data is a standard requirement.
FR-UG-G-003	2	Users will be able to filter data in the garage.	Filtering will help users find what they are looking for without a direct search.
FR-UG-G-004	1	Users will be able to search the garage for builds or parts with specific names.	Searching for the exact thing you know you want is always more efficient than browsing.

4.22.2 Non-Functional

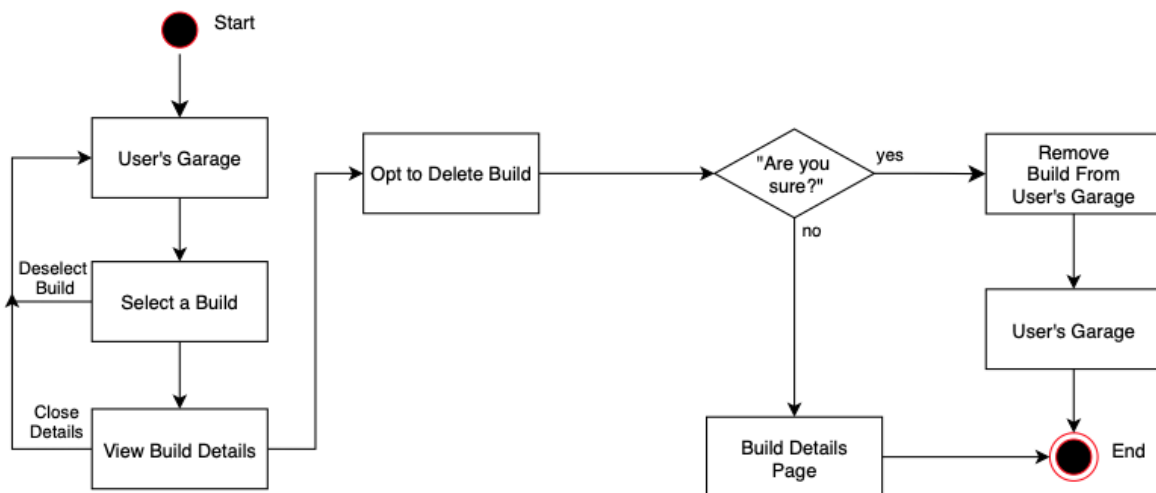
ID	Requirement
NFR-UG-001	The list and filters will load in under 2 seconds on a standard internet connection.
NFR-UG-002	Filters will remain highlighted after selection as additional filters are added or unhighlighting as they are removed.
NFR-UG-003	The search bar will save the last 5 searches in the same session for quick searching.
NFR-UG-004	The user's name will be placed in front of the "Garage" in the interface (____'s Garage).
NFR-UG-005	The search bar will contain most recent queries until the user leaves the site.
NFR-UG-006	All sorts will be done in 2 seconds or less.

4.22.3 User's Garage Activity Diagrams

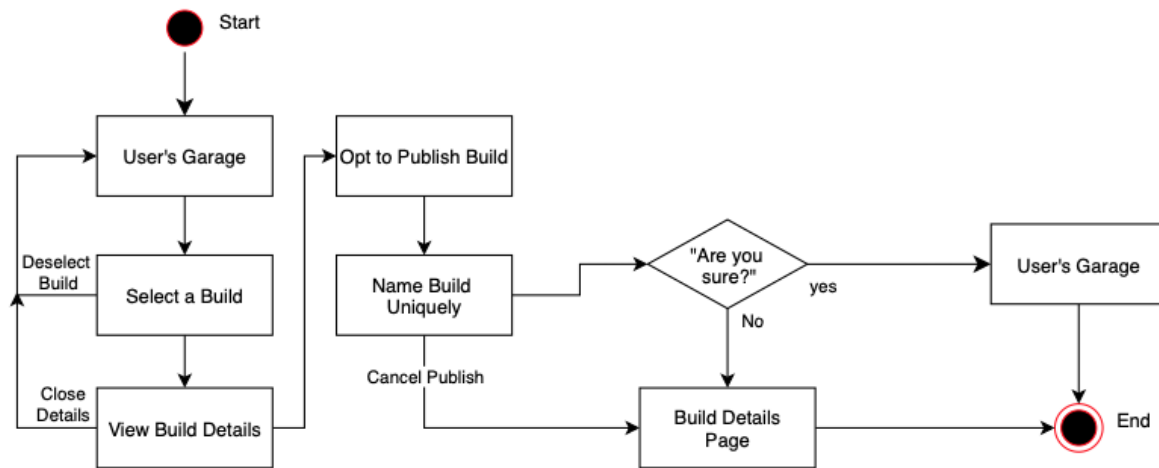
Initially Access User's Garage



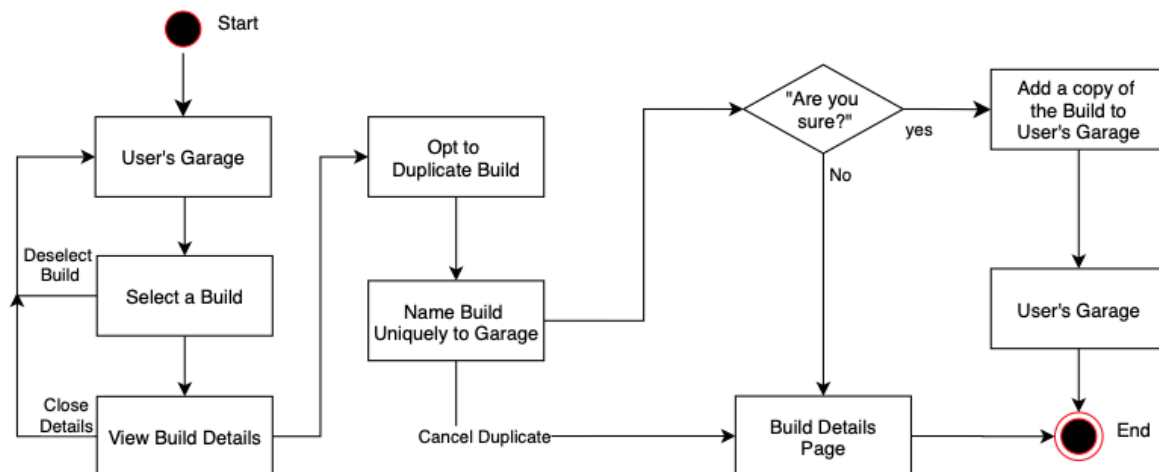
Delete a Build From the Garage



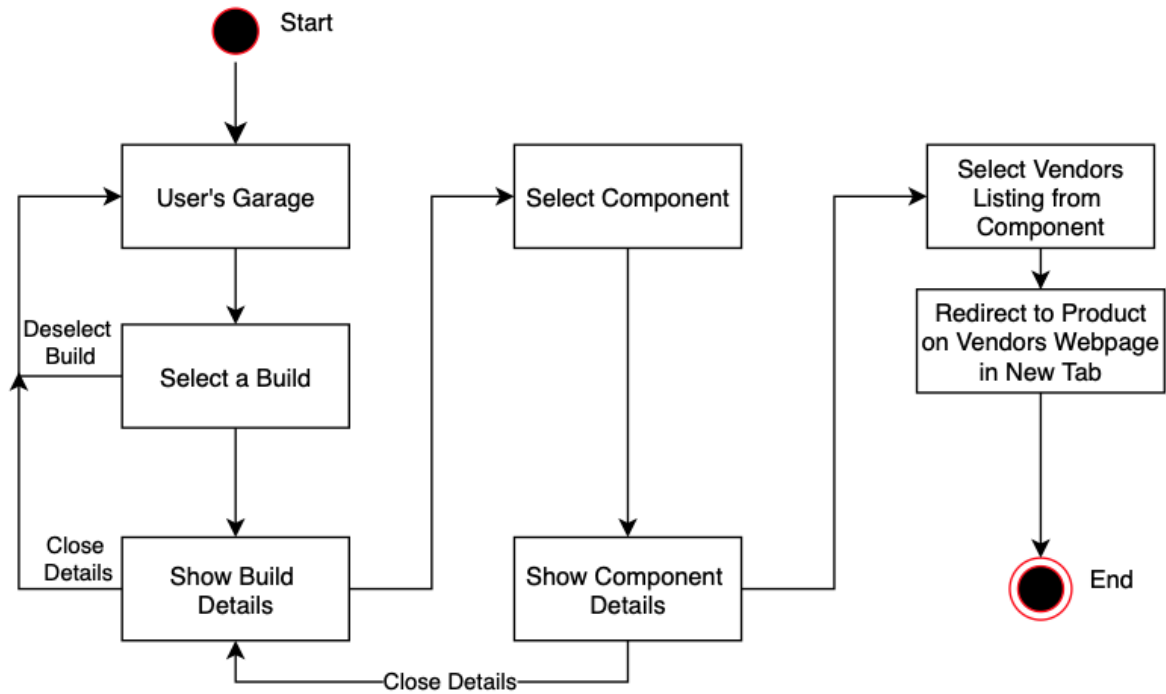
Publish a Build From the Garage



Duplicate a Build in the Garage



View Component on Vendors Website from a Saved Build:



This activity shows the flow through the garage, viewing builds, components, and being redirected to a vendors webpage in a separate tab.

4.23 User Interface

Objectives:

- Offer a simple, understandable, and intuitive environment in which to interact with our product.
- To entice users with aesthetically pleasing and not overly flashy, or active, screens.
- To allow users to interact with different tools without having to load new pages.

Background:

The user interface is intended to be simple and intuitive, users should be able to navigate through several sections without needing to reload the interface entirely, which should aid in user enjoyment.

Requirements:

- Interface must be simple.
- Elements must be consistent.
- Inform the user of actions in progress.
- Interface must be legible.
- Users should be able to change text size without affecting placement of elements.
- Text, buttons, and labels must be clear. (No weird fonts)
- Buttons must either redirect to the logical page (link name) or perform the expected task.

Non-functional Requirements:

- Interfaces must be intuitive.
- Colors should not be offensive to the eyes.
- Text and backgrounds must be different colors, preferably high contrast to aid in clarity.
- Users must be able to see they are logged in somewhere on the page.

Success: Users can see, understand, and interact with the content provided with no frustration, and can easily navigate through the website.

Failure: The system may be unusable to some, many, or all users. Design changes are required.

4.23.1 Functional

Req#	Priority	Description	Rationale
General / Base Functionality			
FR-UI-G-001	1	Interface must be legible.	If the user cannot read the text on the page they will not be able to interact, or may become frustrated with the site.
FR-UI-G-002	1	Users must be informed of actions in progress.	If the users know what is going on, they will be less likely to try to reinitiate the action, or to be frustrated by the lack of system response.

Usability Requirements			
FR-UI-U-001	1	The interface needs to be simple.	Simplicity aids in user enjoyment. Users who have to navigate to several sections to find the tidbit of information they seek will become agitated quickly.

Audit Requirements			
FR-UI-A-001	1	Each interaction with the interface must be logged.	If a user selects to go to the product catalog we will log, with a date and time stamp, that a user has entered the catalog.

4.23.2 Non-Functional

ID	Requirement
NFR-UI-001	Colors should not be offensive to the eyes.

4.24 User Management

Objectives:

- To view all users that are using the website
- To manage permissions of users.
- To add/edit/remove users.

Background:

User management enables administrators to control user access. A group of users may not need access to a different group of users' resources. By limiting access, we greatly increase our security and reduce the risk of data loss.

Requirements:

- Administrators should be able to add/edit/remove users
- Administrators should be able to manage permissions for each type of user
- Administrators should be able to group users according to their association with different organisations and groups

Success: Roles are given to the right users and no user has more power than necessary.

Failure: A user has elevated permissions which could potentially remove users.

4.24.1 Functional

Req#	Priority	Description	Rationale
General / Base Functionality			
FR-UM-G-001	1	Administrators should be able to add/edit/remove users	Our application needs an administrator(s) that has the power to manage users. Without it, we're taking a huge security risk.
FR-UM-G-002	1	Administrators should be able to manage permissions for each type of user	Permissions must be granted to users so their account has some functionality.
FR-UM-G-003	3	Administrators should be able to group users according to their association with different organisations and groups	Grouping makes it easier for assigning roles.

Audit Requirements			
FR-UM-A-001	1	All changes related to user management will be logged.	If an elevation attack were to occur, we could easily locate the cause.

4.25 User Reviews and Ratings

Objectives:

- Offer a system that allows accounts to write reviews and make ratings across the webapp.
- Not allow accounts from rating the same item more than once
- Support image recognition software to make sure written reviews are appropriate.
- Allow accounts to delete reviews
- Allow users to update reviews

Background:

Reviews and Ratings will add a level of credibility to our webapp, as well as make developers aware of user feedback, by allowing users to write their own reviews on PCs that are recommended by our application. This allows users to take advice and information from other accounts, rather than an authoritarian system that acts as the know-all, only one way that works.

Requirements:

- Allow the creation of account reviews and ratings.
- Refuse users from rating the same item again.
- Ability to delete reviews.
- Support image recognition software.
- Allow users to delete reviews.
- Allow users to update reviews.
- Creating reviews and ratings will be used in other web-app features.
 - Most Popular PC builds.

Non-Functional Requirements:

- The review submission type box will have a limit of 10,000 characters.
- The review submission box will allow a maximum photo upload of 5 images.

Success: Account holders will be able to inform other users, in potentially verbose manner, what they think of the shared system design, and assign a rating they feel the design deserves.

Failure: Users will be unable to post or rate builds, lowering interest and detracting from future use.

4.25.1 Functional

Req#	Priority	Description	Rationale
General / Base Functionality			
FR-RR-G-001	1	Accounts can write and publish a review.	The feature will not be accepted without this requirement
FR-RR-G-002	1	Accounts can make a rating of 1-5 stars.	The feature will not be accepted without this requirement

Security Requirements			
FR-RR-S-001	1	Image recognition software to verify that images are appropriate for posting.	Features will not work without this requirement. Inappropriate images should not be allowed.

Usability Requirements			
FR-RR-U-001	1	User interface will allow users to easily input their reviews and make ratings without difficulty.	Without a proper UI, the user will not know how or where to write reviews / make a rating.

Audit Requirements			
FR-RR-A-001	3	Updates made to reviews will be time-stamped.	To provide users the validity of a review, for example older reviews might hold less credibility when compared to new reviews.

4.25.2 Non-Functional

ID	Requirement
NFR-RR-001	The review submission type box will have a limit of 10,000 characters.
NFR-RR-002	The review submission box will allow a maximum photo upload of 5 images.

4.25.3 User Review and Ratings Activity Diagrams

- User creates a review on a build

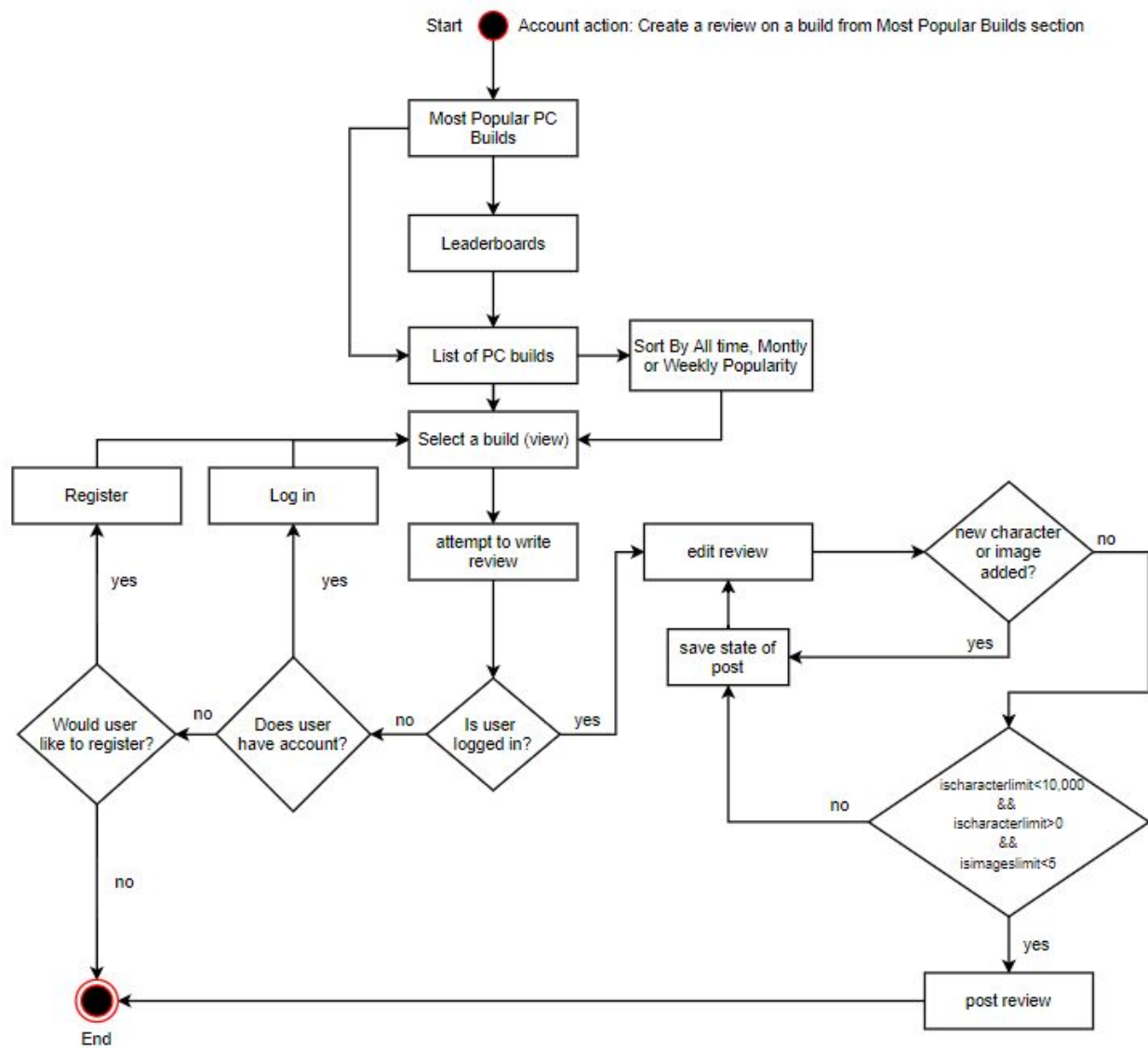


Fig. 1 Most Popular Builds

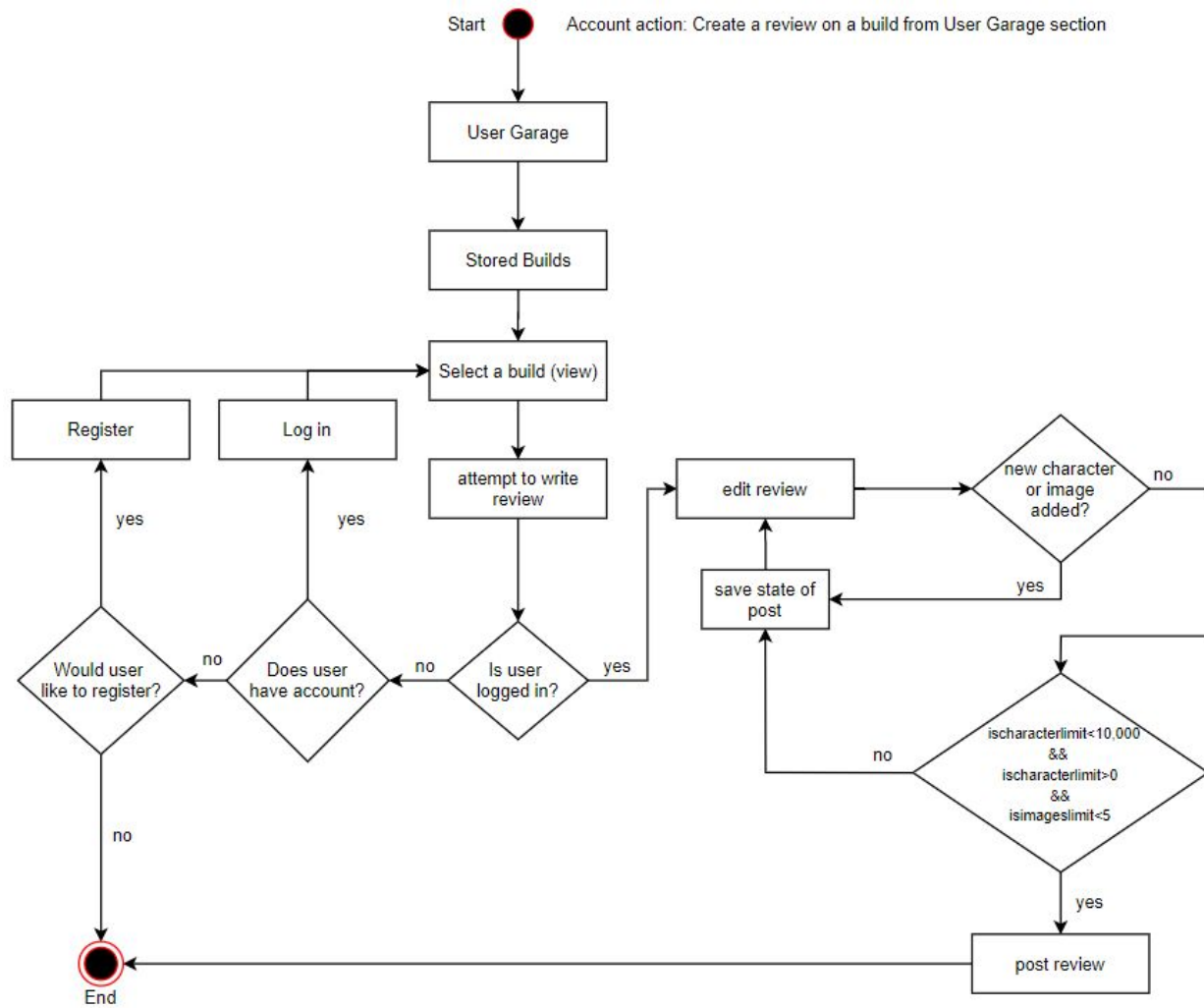


Fig.2 User Garage

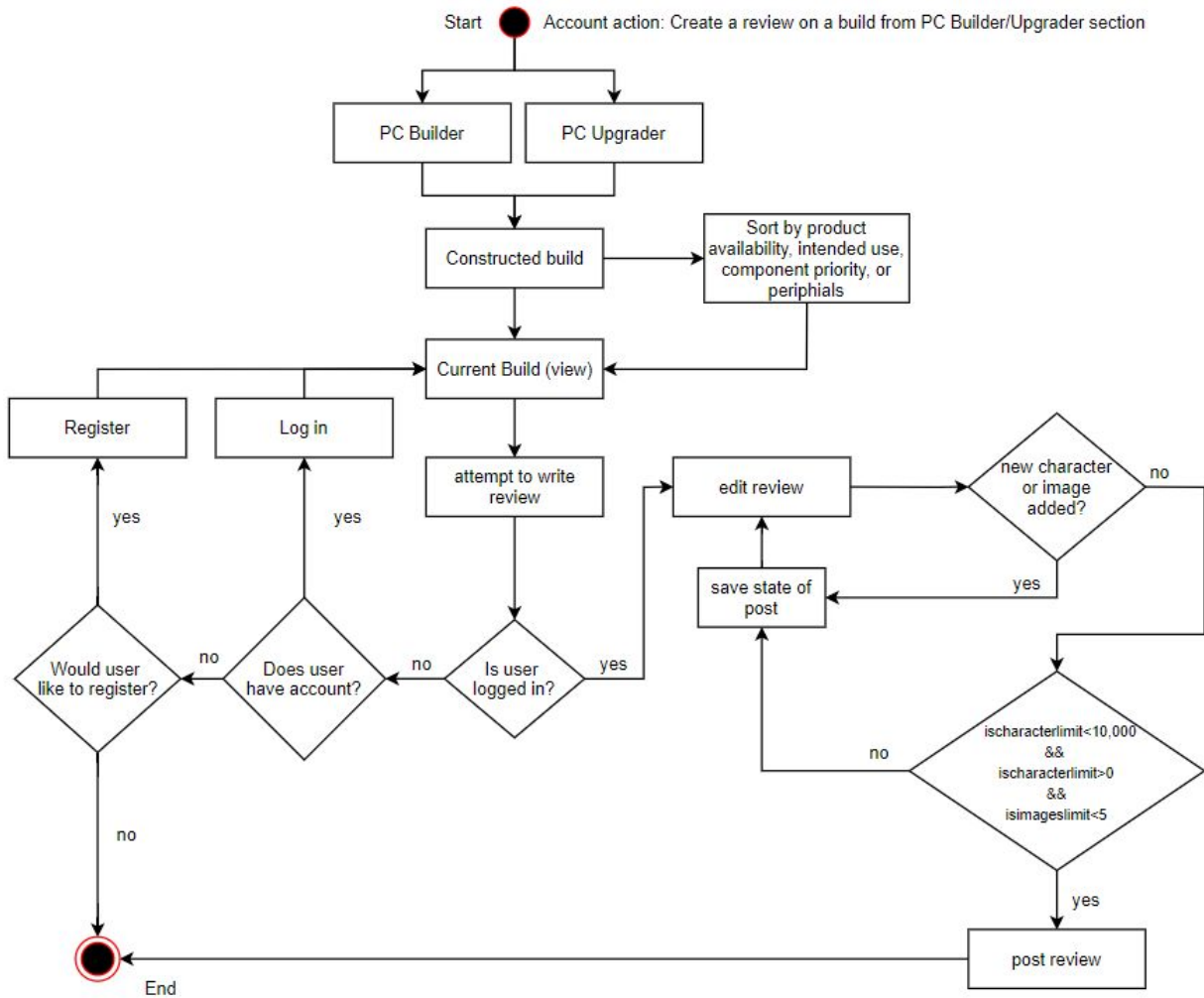


Fig.3 PC Builder/Upgrader

- User adding a star rating on a build

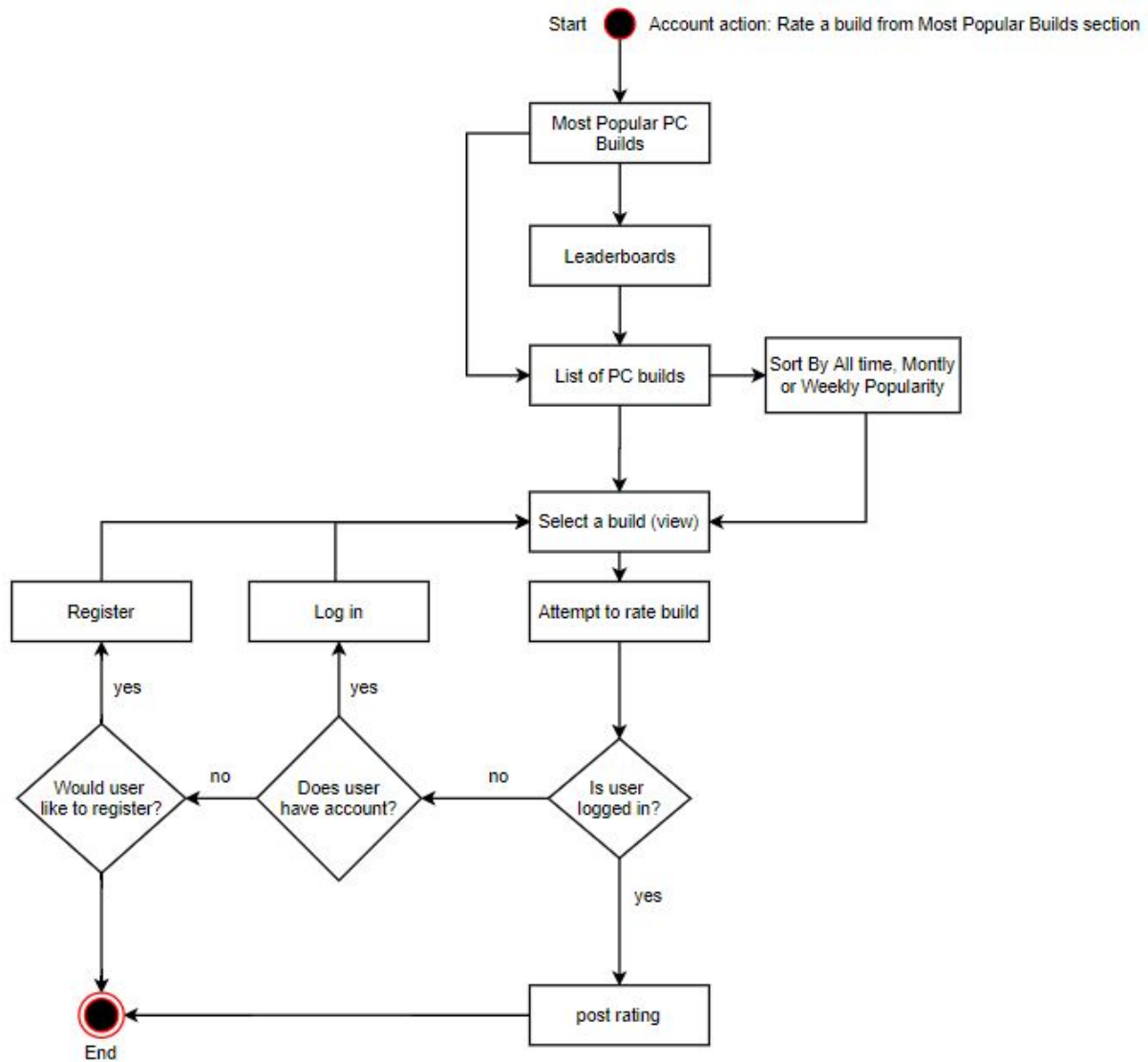


Fig.1 Most Popular Builds

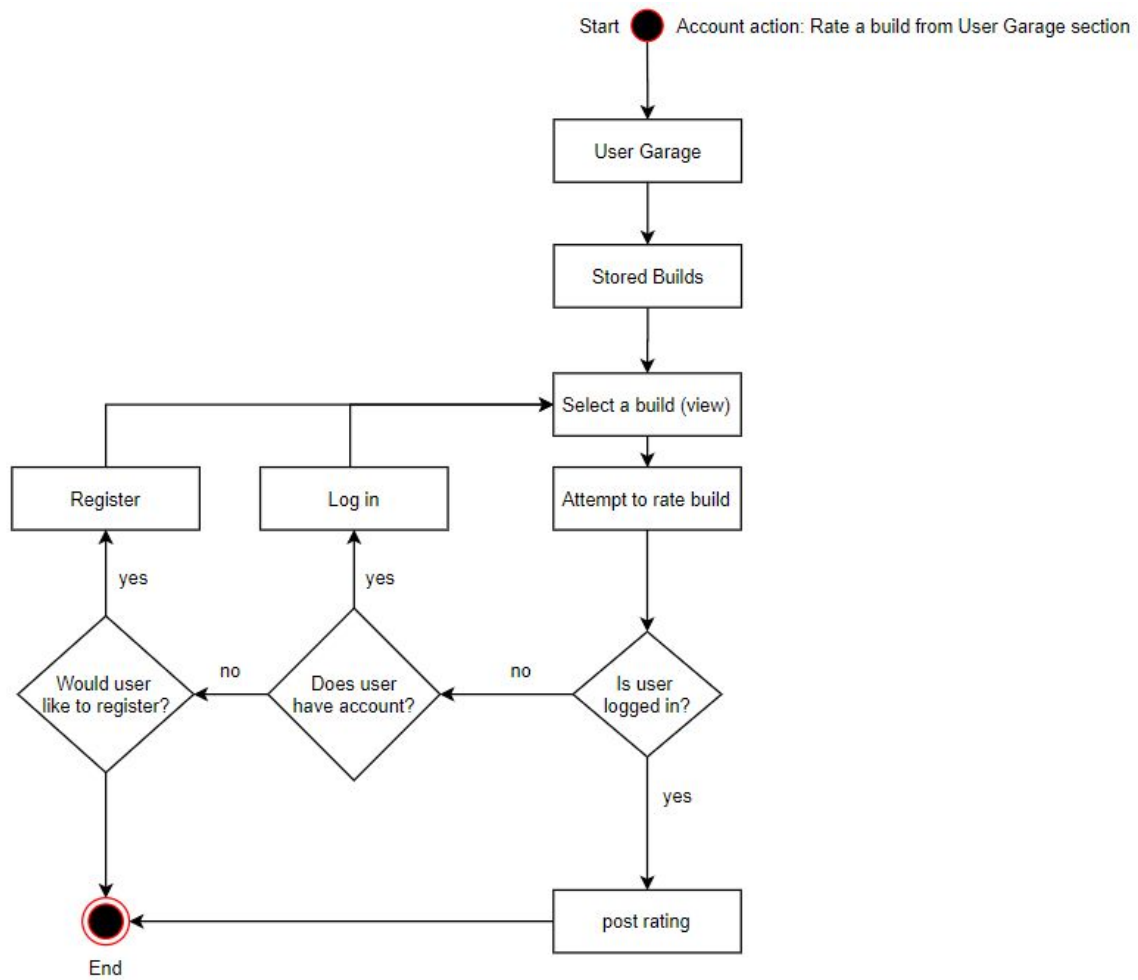


Fig.2 User Garage

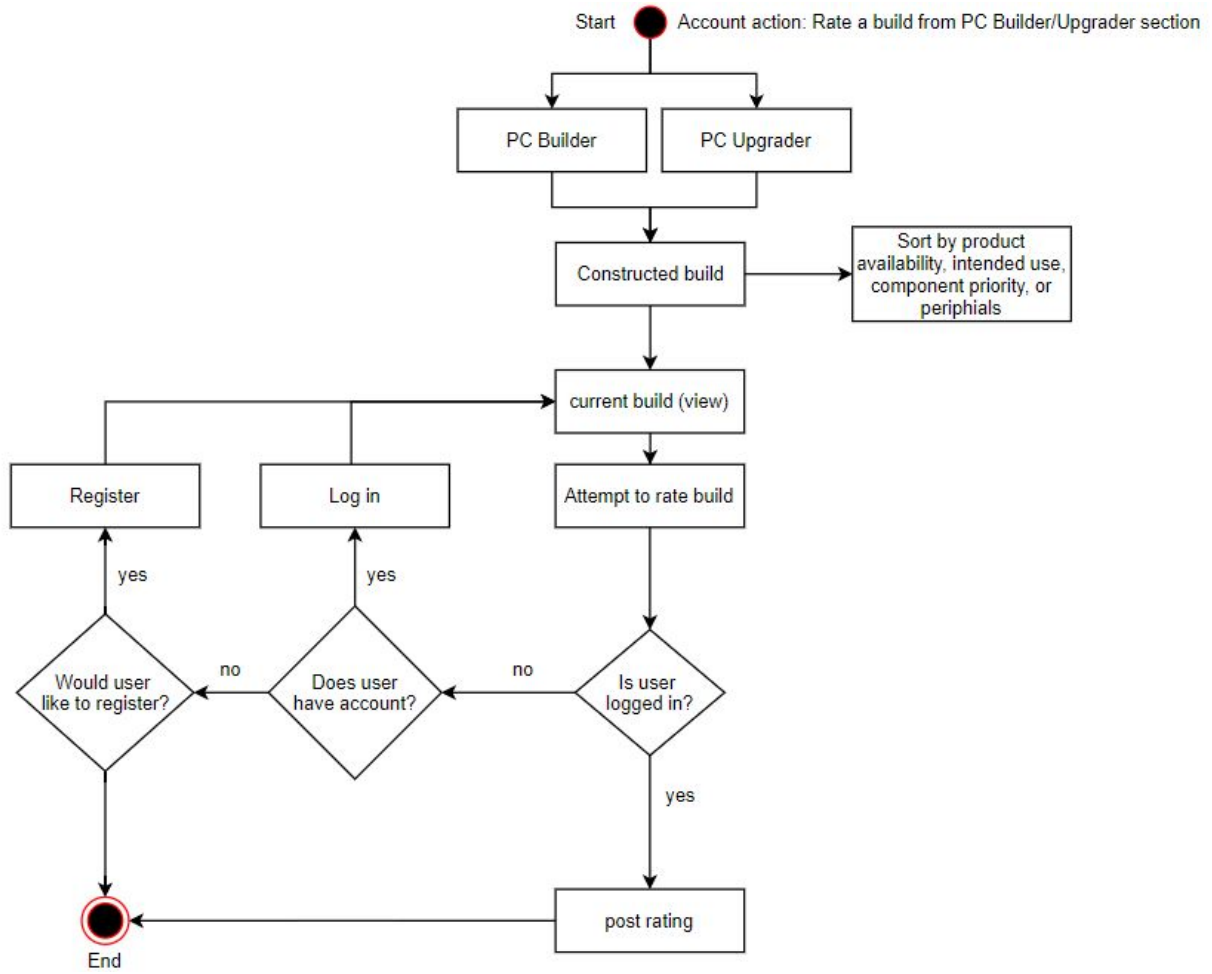


Fig.3 PC Builder/Upgrader

4.26 Vendor Linking

Objectives:

- To link a vendor to our website.
- Allow vendors to create a vendor account to link their products.
- Allow users to view vendor products.

Background:

Vendor linking was first brought up when we were talking about individual components once a build is chosen. Since our algorithm optimizes a build based on a budget, we thought to add links to purchase those individual items from various vendors. Issues that came with this feature involve how the linking happens. There are two options: using vendors' APIs and web crawling. Since some vendors don't offer an API, web crawling would be our next course of action in retrieving component information. In the case where a vendor wants to link with our website and doesn't have an API, we offer a vendor account which will allow a vendor to add/update/remove their own items.

Requirements:

- Allow vendor accounts to link their own products.
- Allow vendor accounts to edit or remove their own products.
- Link a vendor to our website if they offer an API and add and update products automatically.

Non-Functional Requirements:

- Vendor data will be refreshed every 6 hours.
- Vendor listings that have not been updated within 30 days will be removed.

Success:

Vendors will be successfully linked allowing full controls to managing their own products on their vendor page.

Failure:

Failure: Vendors are unable to link their business and will then search for a different website that offers such.

4.26.1 Functional

Req#	Priority	Description	Rationale
General / Base Functionality			
FR-V L-G-0 01	3	Users can create a vendor account to add/update/remove their own products.	This feature isn't necessary for the user to use our application. Our main use is to assist in building a PC on a budget. This is handy for users/vendors to add price information and link their products.
FR-V L-G-0 02	2	Vendors will be added if the vendor has an API.	This feature isn't necessary for the user to use our application. However, it's helpful for the user to get a rough estimate of how much top companies sell products for.
Security Requirements			
FR-V L-S-0 01	1	Since vendors will be publishing products, they will need access to our database. Posts to the database must be through a parameterized query.	We are preventing SQL injection attacks. If we don't, anyone can delete information from our database.
Usability Requirements			
FR-V L-U-0 01	1	User interface will allow users to easily add a vendor account and their products.	UI/UX is important. If a user doesn't like how our website looks or if it's difficult to navigate, they'll look for another website.

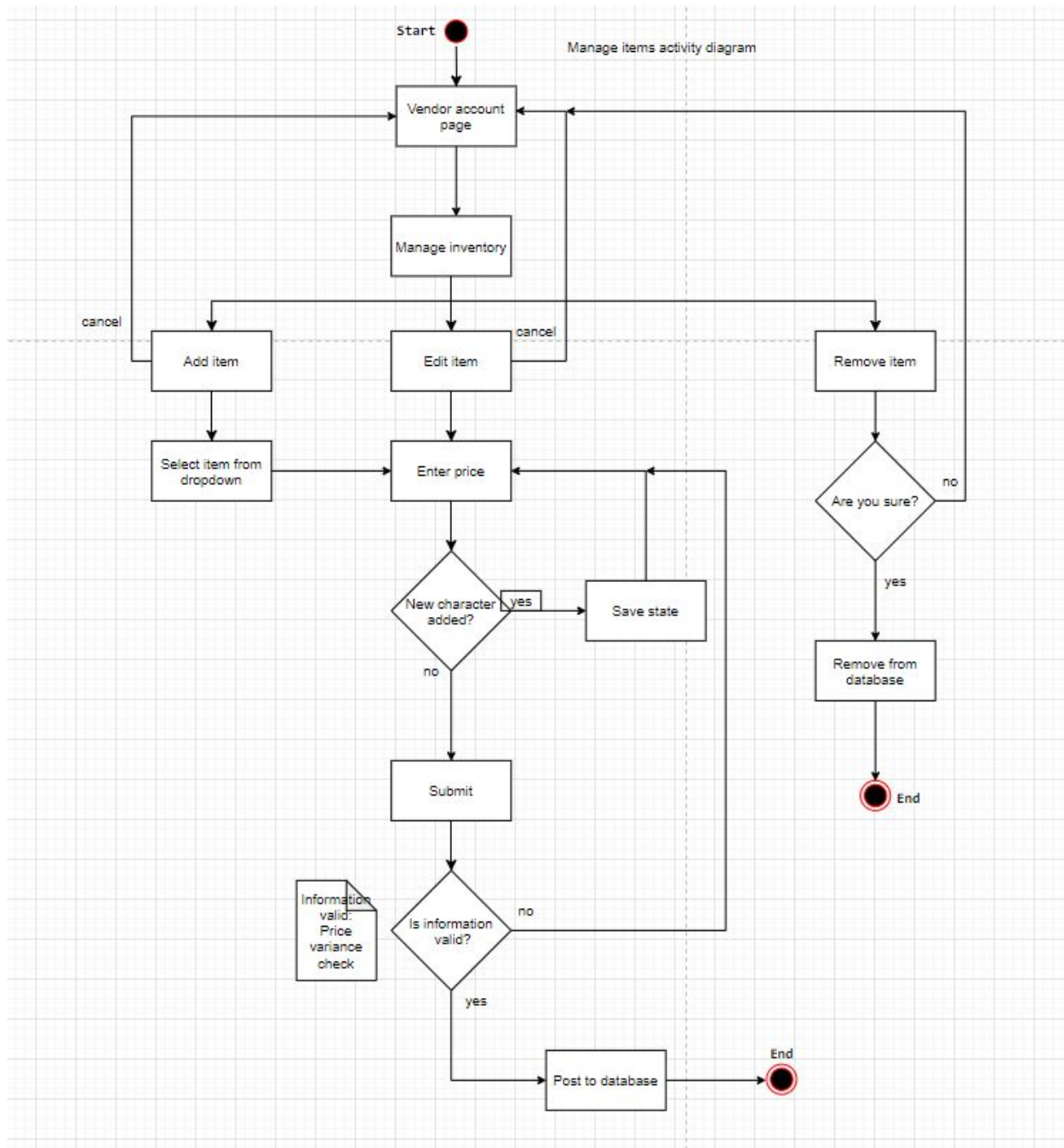
Audit Requirements

FR-V L-A-0 01	4	Any changes made to a vendor's product will be logged.	We will log in order to monitor suspicious activity.
---------------------	---	--	--

4.26.2 Non-Functional

ID	Requirement
NFR-VL-001	Vendor data will be refreshed every 6 hours.
NFR-VL-002	Vendor listings that have not been updated within 30 days will be removed.

4.26.3 Vendor Linking Activity Diagrams



5 Appendices

5.1 Abbreviations

AC - Archiving	REG - Registration
API - Application Programming Interface	SC - Security
CC - Components Catalog	SP - Support Page
DSA - Data Store Access	UAC - User Access Control
EH - Error Handling	UAD - User Analysis Dashboard
LOG - Logging	UX - User Experience
LI - Log In	UG - User's Garage
LO - Log Out	UI - User Interface
MPB - Most Popular PC Build	UM - User Management
NC - Network Communication	RR - User Review and Ratings
PB - PC Builder	VL - Vendor Linking
PU - PC Upgrader	FR - Functional Requirements
PR - Privacy	NFR - Non-functional Requirements
PSU - Power Supply Unit	

5.2 Glossary

Account - A registered user.

Application programming interface - an interface that defines the kinds of calls or requests that can be made, how to make them, the data formats that should be used, the conventions to follow, etc.

Image recognition software - A program that will be used to make sure that images to be uploaded are checked and verified for inappropriate material.

Modular - Modularity refers to how many wires can be removed or are built into the PSU box. Fully modular means each wire has to be individually plugged into the PSU and non modular means the wires are built into the PSU and cannot be unplugged.

Parameterized query - A pre-compiled SQL statement where you only need to provide a parameter. This is mainly used to prevent SQL injection attacks.

5.3 Related Documents

- **User Experience:** <https://www.usability.gov/what-and-why/user-experience.html>