
CS958 PROJECT

COURSEWORK ASSIGNMENT

LEWIS W. BRITTON
202194412
UNIVERSITY OF STRATHCLYDE
Glasgow City, Scotland

BURNING ROOTS:

WRITTEN & DIRECTED BY
John Hughes
EXECUTIVE PRODUCER
Michael Mann
CREATED BY
Anthony Yerkovich
MUSIC COMPOSED & PERFORMED BY
Jan Hammer

... WORDS

DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE SOFTWARE DEVELOPMENT AT THE
UNIVERSITY OF STRATHCLYDE

ACADEMIC YEAR 2020/2021

ABSTRACT

INDEX TERMS:

DECLARATION & INFORMATION

This dissertation is submitted in partial fulfilment of the requirements for the degree of Master of Science Software Development at the University of Strathclyde. It accords with the University’s regulations for the programme as detailed in the University Calendar.

MAIL TO: wi.lbritton@yahoo.com	TELEPHONE: 07415 212 ***
WEBSITE: lewisbritton.com	GITHUB: FedeRog1977

This document’s presentation reflects the use of L^AT_EX typesetting (Figure B7), using Computer Modern Unicode (Figure B8) (I haven’t reached my GNU Troff phase yet). This escapes the inane formatting requirements of my institution. References are presented using B_IB_TE_X, favouring *oblique* over *italic*, in-line with Donald E. Knuth’s preference (Knuth, 2020). The process is executed in command line using Vim, which is a very powerful editor that has many commands, too many to explain in a tutor such as this. For maximum optical pleasure, the use of M μ PDF is vigorously advised with [-I]. Navigate this document using h(\leftarrow), j(\downarrow), k(\uparrow), l(\rightarrow), ensuring that the Caps-Lock, Super-Key ‘mod’, or any other command key is not depressed. Note that the Oxford Serial Comma is favoured throughout this text. This study’s sentence structure focuses on pragmatics and syntax, disregarding bloated filler content. Arguments are coherent, logical, definitive and straight-to-the-point. Nugatory theory is ignored. If you are curious about any of the mathematical, operational, logical, etc., symbols or notation used in this report, a comprehensive L^AT_EX-syntax-based symbolist will be available from my [website library](#) from approximately summer 2021.

The word count of this piece reflects relevant content from titles, heading classes 1, 2 and 3, paragraphs, footnotes, tables (excluding [results] tables 4.7, 5.10 \rightarrow 5.20), table titles, figures, and figure titles in *Chapters 1 \rightarrow 6*. Word count excludes any pre/succeeding content from *Abstract*, *Declaration & Information*, *Acknowledgements*, *Table of Contents*, *Appendices*, and *Bibliography*.

I declare that this document embodies the results of my own work and that it has been composed by myself. Following normal academic conventions, I have made due acknowledgement of the work of others.

Signed:

Date:

ACKNOWLEDGEMENTS

I would like to thank my dissertation supervisor, Dr Devraj Basu, for his approach with regards belief that students must be self-disciplined, organised, structured and punctual to their own degree. This closely relates to my own personally practiced work ethic and philosophy of the ASAP standard.

I would like to give credit for the computational aspect of this study to one of my biggest inspirations, John “The Tzar” Kelly. He inspired my love for everything bare-bones computational, from simple arrays (of hope), through Hyperthreading-enabled, all the way to x86 Assembly. I would also like to accredit Luke Smith for the foundation of my knowledge of Bram Moolenaar’s Vim and L^AT_EX. This study’s presentation would not have optimal without Smith (2015).

This piece would not have been as efficient without the aid of the only acceptable Linux distribution, ‘distro’ if you will, Arch Linux. I would like to thank Judd Vinet for his eye-opening and life-altering contribution to the development and computer-system enthusiast community. ‘The Arch Principal’ is certainly out in high force. Finally, for making use of this software mechanically efficient, I would like to thank IBM for the creation of the ThinkPad T23, X30, T42, R50e, T60, X60, X200, X220 and T420 neoVimPads, the UltraDock, and the 1987 Model M *Catastrophically Buckling Compression Column Switch and Actuator* typehorse (US369 9296A, 1972). For your convenience, one of my [blog posts](#) can satisfy your interest in this.

TABLE OF CONTENTS

1	INTRODUCTION	1
1.1	PURPOSE & INDUSTRY	1
1.2	SYSTEM STRUCTURE	1
1.3	USERS & PLATFORMS	2
1.4	DEVELOPMENT PROCESS	3
1.5	DISPOSITION	4
2	RESEARCH & EVIDENTIAL BACKGROUND	7
2.1	AREAS OF EXPLORATION	7
2.1.1	HIKING ROUTES	7
2.1.2	HIKING EQUIPMENT	7
2.1.3	PERSONAL FITNESS & ABILITY	8
2.1.4	GEOGRAPHY	8
2.2	MATERIAL INVESTIGATION & HEURISTIC EVALUATIONS	8
2.2.1	WALKHIGHLANDS	10
2.2.2	STRAVA	10
3	DATA PROCESSING & METHODOLOGY	11
3.1	DATA ACQUISITION	11
3.1.1	LOCATION	11
3.1.2	SELF-DECLARED ATTRIBUTES – ABILITY	11
3.1.3	SELF-DECLARED ATTRIBUTES – EQUIPMENT	11
3.1.4	WEATHER	11
3.1.5	ROUTE DATA	11
3.1.6	REGIONAL DATA	11
3.1.7	LANDMASS DATA	11
3.2	DATA MANIPULATION & OUTPUT	11
3.2.1	ROUTE OVERVIEW	11
3.2.2	WEATHER SYSTEM	12
3.2.3	‘RANGER’ CALCULATOR	12
3.2.4	REGIONAL & LANDMASS GENERAL	12
3.3	INFORMATION ACQUISITION & OUTPUT	12
3.3.1	PURPOSE	12
3.3.2	KEYS & SUGGESTED READING	12

4	SYSTEM REQUIREMENTS	13
4.1	SCALE & SCOPE	13
4.2	GATHERING & PRIORITIZATION METHODOLOGY	14
4.3	REQUIREMENTS	16
4.3.1	FUNCTIONAL	16
4.3.2	NON-FUNCTIONAL	17
4.3.3	USER STORIES	17
4.4	USER ACCESSIBILITY	18
5	SYSTEM DESIGN	19
5.1	SYSTEM ARCHITECTURE	19
5.2	USER INTERFACE	19
5.2.1	LOGICAL DESIGN	19
5.2.2	GRAPHIC DESIGN & COMMUNICATION	19
5.2.3	INTERFACE TREE	19
5.2.4	USE CASE DIAGRAM	19
5.2.5	PRIMARY USE CASE EXAMPLES	19
5.3	DATA STRUCTURE	19
6	SYSTEM CONSTRUCTION & IMPLEMENTATION	20
6.1	DEVELOPMENT & LANGUAGES	20
6.1.1	DEVELOPMENT ENVIRONMENT	20
6.1.2	FRONT-END & USER INTERFACE	20
6.1.3	SEMI-REAR-END & SUPPORTING DATA	20
6.2	APPLICATION PROGRAMMING INTERFACES (APIs) & LIBRARIES	20
6.2.1	GEOLOCATION	20
6.2.2	ORDNANCE SURVEY	20
6.2.3	OPEN WEATHER	20
6.2.4	LEAFLET	20
6.2.5	MAPBOX	20
6.2.6	CHART.JS	20
6.3	PROJECT MANAGEMENT	20
6.3.1	LIFE-CYCLE & TIMING	20
6.3.2	SUPPORTING TOOLS	20
7	EVALUATION	21
7.1	REQUIREMENTS	21

7.2	DESIGN, CONSTRUCTION & IMPLEMENTATION	21
7.2.1	HEURISTIC EVALUATION	21
7.2.2	USER ACCEPTANCE (UAT), ACCESSIBILITY & USABILITY . . .	21
7.2.3	‘PROTOTYPING’ & CONSTANT EVALUATION	21
7.3	ADDITIONAL TESTING	21
7.3.1	‘TEST-DRIVEN’ DEVELOPMENT	21
7.3.2	UNIT TESTING	21
7.4	CONSTRICTIONS	21
8	DEVELOPMENT CONCLUSIONS	22

LIST OF TABLES

2.1 Heuristics & Severity Rating 9

8.2 Heuristic Analysis I

8.3 Heuristic Analysis II

LIST OF FIGURES

1 INTRODUCTION

1.1 PURPOSE & INDUSTRY

The system developed throughout this project is a functional web application based on providing user-location-based and external GPS data. User-based GPS services are based on data relevant to the user's machine and external services are based on context-relevant data. The system, which so forth may be referred to as 'the system', 'the application', 'the site', provides detailed information, guidance and recommendations relevant to sport and leisure, particularly hiking and its associated practices, in Scotland's fine rural outdoors and [*these mist covered mountains, which are home now for me*]. Therefore, upon a hypothetical release of a full version of this system, it would be a direct competitor of services such as Walkhighlands, Strava, Garmin Connect, All-Trials, etc. Due to the autistic and comprehensive nature of its development, it would be in the market not in the competition-driven business, but in the [\[empire business\]](#).

1.2 SYSTEM STRUCTURE

The site upon which the system is spread is static, not dynamic, meaning any possible requests made by the user are based on existing data. The site does not reference an external database for any purpose. Thus, no PHP or SQL-relevant content. The system is segmented into four, approximately equivalent, parts with the metric(s) determining their weight being algorithmic volume, number of services, etc.

The first, 'home'/'drafting room', page allows the user to view a comprehensive overview of the site's offerings. It allows the user to quick-view activities in the 'overview' section; analyse their projected personal ability and their projected gear performance (upon their input(s)) in the 'conditioning' section under 'ability' and 'equipment cache'; and, view a coordinate-based weather briefing in the 'weather' section. All of these sections include 'key's and 'suggested reading's. Yes, this site is that obnoxious.

The second, 'conquest map', page allows the user to view and interact with various GPS and mapping features including their location and location seeking. It also includes functionality which allows pinpointing of particular features upon interaction, such as Munros, Munro Tops, Corbetts, Corbett Tops, etc. Further it includes aspects which allow the user to seek, be recommended, select and print GPX routes on the map.

The third segment is the ‘ranger calculator’ which is used purely for analytical purposes and allows the user to input data relevant to their ability, equipment, routes, etc. and will deliver output in the form of statistics tables and charts based on various computations. This element does not implement any GPS functionality in-line with the system’s primary focus however, is extremely relevant.

The final segment is the ‘general search’ function which allows the user to input or select search criteria which returns a comprehensive overview of all statistics relevant to the match(es). Unlike the other sections, this is more subjective and informative, as opposed to being logical / statistics-oriented. That is, it exists more so for the users understanding of what they’re doing and how to actually interpret some of the statistics they’re being delivered in other elements. It is open to their use and interpretation.

The service is called ‘*Burning Roots*’. No, the rhetorical use of satiric misspelling is not unintentional malapropism; it is in fact deliberate. In harmony with the feeling you’d experience when [racing south-west to Lone Stallion Ranch], the term references that certain ‘*burning* desire’ for freedom in the sweet country air, the one you only experience when digging to your deepest ‘*roots*’ to achieve a new personal record or firmly assert your dominance over your inferiors. This service uses and computes data to encourage a user to take to the trails with motivation to be the fastest, most efficient, most prepared and most endured athlete on their *routes*.

1.3 USERS & PLATFORMS

All users of this system will be sport/leisure oriented and as this is focused on a specific group of enthusiasts who have a firm set of beliefs and a strong pre-developed relationship with their sport (lifestyle), it will likely only receive traffic from athletes who are already hiking-inclined. It may encourage new hikers due to the comprehensive nature and customization opportunity of the learning and planning material however, it is unlikely. The most efficient empires dominate only one type of market. This market does however expand to: walkers, hillwalking enthusiasts, scrambling-inclined 4x4s, [T6 vanlife] climbers, and Scottish mountaineers / ice climbers.

Upon original briefing, this system was planned to include a road cycling section which would essentially mirror the hiking part with cycling-specific data. However, upon reflection this element is irrelevant for two reasons. The first being all functional areas

are covered by the computational processes involved with hiking data. And second, road cycling has little association with hiking and therefore the adjacent sport may be seen as irrelevant by a specific user. If it were to be included, it would only be logical to implement a wider array of sports for example, excluding road cycling and including mountain biking and (fell) running, which are actually relevant to hiking. Or an even wider array if road cycling were to blend in seamlessly. This is unnecessarily [time-consuming] and only duplicates processes and would not deliver additional benefit, only diminishing returns, upon the project marking process. I'm not [the Zuck'], not only do I not have the time or resources for this, I do not have the relevant background knowledge.

As far as platforms go, the user arrives at this site through a web browser. This system is deployed as a website and is therefore extremely versatile and usable on any device. Browser caching of script elements allows a user to view and interact with the relevant data offline, provided they receive GPS signal. For example, they can still view their location and route on the 'conquest map'. Of course, [this means that] the site is constructed using HTML, CSS and JavaScript. Elements of JavaScript allow this site to dynamically scale to various device sizes tailor relevant content to these devices.

1.4 DEVELOPMENT PROCESS

As this is a [solo project], it's one man, his [ThinkPad X220], [Artix Linux] and his [neoVim] setup. There is little requirement for extensive use of any formal [inane] project management methods such as team-based allocations or associated time-based or progress management coordination frameworks. Therefore, any adherence to processes aimed at mapping management of this project are / have been more logic-oriented and variable, allowing creative freedom. I work on an ASAP basis so one creative day of thinking may be followed by a [5am – 11pm] of implementation, which may then be followed by 2 days of idling. Any formal micro-level plan would be redundant.

Succeeding acquisition of user requirements, the most important part of gaining an understanding of how this system would look and operate is determining how the user interacts with the various aspects of their sport. That is, what data/inputs must the user provide, how will this be used, and what will it be used for to satisfy the requirements. Therefore, the mapping of the functional structure of the user interface (UI) leads this in the sense that it demonstrates the logic and process of interaction

relevant to this data. Thus, this creates a valid starting point. And so forth, the development process of this project reflects what follows:

Requirement analysis
 → Structural design
 → Usable data construction
 → Graphical user interface functionality
 → Usable data implementation
 → Graphical user interface graphic design
 → Testing

Following the structural mapping, there is little sense in proceeding without any data to work with as incremental testing of site functionality would be challenging to impossible. So, it's at this point which the acquisition of relevant data takes place. In this case, this data accounts for non-user-centered data such as GPS coordinates, map regions, landmasses and their attributes, etc., which are essential for the majority of computations. Therefore, not only is [JavaBloat] implemented to manage site dynamics, it also computes based on data from these discussed files, in JSON format. It is only after this when the graphic design of the site can be allocated more focus, however of course much of it comes instinctively along the way also. After this, and frequent incremental developer tests, the system is ready for more expansive developer and user testing.

For the natural ease in workflow, for the developer's mental state, and for the minimization of [nugatory] methodologies and [bloated] task flow [cargo donkeys] such as IDEs and [froymeworks], all files (including 'code', data files, notes and write-up) are [composed and performed by Lewis Britton] in [Bram Moolenaar's Neo Vi Improved], in the command line of a pragmatic dwm setup on Artix. HTML, CSS and JavaScript is written from scratch in plain text format, therefore using no environment prompts or assistance, in order to keep the process practical. All write-up documentation is transcribed using [Donald 'Don' E. Knuth's T_EX]. Or as some [neomoderinists] like to use, L^AT_EX. Due to time constraint, there is unfortunately no mastering the fine art of [GNU Troff], so transcriptions may not appear [optically optimal] without the famous Groff-PostScript [multi-kill].

1.5 DISPOSITION

So forth, the following elements of this project are responsible for...

2 RESEARCH & EVIDENTIAL BACKGROUND

Explores the areas of research and data gathering including hiking routes, hiking equipment, personal fitness and ability, geography and geology. Furthermore, presenting and examining results and conclusions to evaluations of currently existing competitors' services. This section acts as a literature review would in a paper based on, say, an empirical piece investigation; providing the foundational material upon which development aims to further succeed and 'develop'.

3 DATA PROCESSING & METHODOLOGY

Mapping and justifying the data selected for use in the system. This is broken into three segments, first being data acquisition which explains how and why data is sources from third parties and inputted from users. The second section explains how this data is manipulated and the third; statistical and informative output.

4 SYSTEM REQUIREMENTS

Describing the scale and scope of the users and their requirements for this system, and mapping how these are prioritized at the beginning of and throughout the project. This is in the context of functional and non-functional requirements.

5 SYSTEM DESIGN

Displaying the structure of the system architecture and how the logic aligns with the requirements of the system. Also, describing the various aspects of the user interface's functional and graphical communication and design process. It's apparent at this point how the data structure is made relevant to the design of the system using the requirements.

6 SYSTEM CONSTRUCTION & IMPLEMENTATION

Providing a closer look at and justification of the development environment, languages and protocols selected for the creation of this system and exploring the various APIs and JavaScript libraries and other supporting tools used to enhance the system and allow it to function in harmony. Also, Providing an overview of how these elements

were implemented from a project management point of view.

7 EVALUATION

An evaluation of requirements gathering and the feasibility and tangibility of their implementation, an review of self-testing methods and additional tests, demonstrations of prototyping and various other aspects of developer' and user-centered testing.

8 DEVELOPMENT CONCLUSIONS

Summaries of development conclusions which are presented pragmatically as objective, critical notes and possible segues.

2 RESEARCH & EVIDENTIAL BACKGROUND

2.1 AREAS OF EXPLORATION

This system is designed to combine and present aspects of the different types hiking routes and their attributes relative to their conditions; the recommended and available equipment for users to investigate and explore expansive opportunities within; the personal fitness and ability level of users and therefore, their ability to interact with different routes and opportunities; and, the geography and geology of various aspects of hiking routes which contributes to various other factors within user ability how users may interact with the routes themselves.

2.1.1 HIKING ROUTES

The hiking routes are the foundation of this system. They provide the purpose and reason to the GPS aspect of the system. There are various demonstrations of how hiking routes are implemented in different ways across slightly different platforms. Walkhighlands for example, presents very static use of these; displaying a page per routes listing manually expressed data and literature. Each route cannot be interacted with and had no dynamic attributes. They are simply listed for user interpretation. In this system, GPS routes are made relevant to the particular user interacting with them. Routes are not only selected through subjective choice, they are dynamically relevant to both user conscious and subconscious attributes.

2.1.2 HIKING EQUIPMENT

Hiking equipment is arguably half the battle when it comes to most effectively tackling projects. Although I've had my fair share of 15 mile proj's with approach shoes and one litre of water, alongside [Griff] in shredded boots and MTB tee-shirt and shorts; it's still pretty important. To the [Maddie Owens] of the industry. Regardless, many routes require particular components and combinations of equipment, including the appropriate knowledge of such. This means it is completely necessary that, especially under-experienced hikers, are as aware of the precautions and hazards present on selected and suggested routes. Including a metric which accounts for the user's equipment, alongside relevant literature, ensures that this site takes the implements the correct protocol to see that the user does not make any unrealistic inference regarding routes. Once again, in other services such as Strava and Garmin Connect, there are not metrics which account for these attributes. Within Walkhighlands, there is plenty

of literature available however, this information is not quantified and translated into data input so therefore leaves routes static, relative to equipment.

2.1.3 PERSONAL FITNESS & ABILITY

Fitness is undoubtedly the single most important factor in any sport. Skill, knowledge, technique, understanding of kinematics and dynamics of the human body, and things alike all contribute to the degree to which you excel at a sport. However, without raw fitness you might as well sit on the bench. Keeping the heart rate regulated, understanding which parts of your body to engage and not to engage, correctly distributing force and converting torque are all closely related to personal fitness and therefore must be quantified in such a way which reflects a user's expected effort and ability to complete a route. This effects results such as elapsed time, breaks required, fatigue and estimated recovery time, etc. Again, route planners such as Walkhighlands do not offer any form of input using these metrics. Strava and Garmin Connect do however make estimates following completion of activities however, do not allow these statistics to be re-used and inputted as variables determining results of estimates of future activities.

2.1.4 GEOGRAPHY

The use of geography within this system is fairly static and informational. That is, it is not quantified and it's attributes cannot be used as inputs which determine future estimates and results. As physical geography is out of the control of user's however, there wouldn't be much use in quantifying it. It is however useful if users have an understanding of what geographical features are and how they can have an impact on their routes. Of course this particular feature is irrelevant to much of Strava and Garmin Connect's functionality and is therefore not included in any form. Walkhighlands does include excellent educational sources however there is not much continuity and consistency to their presence. Therefore, relevance is often unaligned.

2.2 MATERIAL INVESTIGATION & HEURISTIC EVALUATIONS

As discussed, improving upon various features and aspects of Walkhighlands, Strava and Garmin Connect is a relevant step in developing requirements and informal desires from this system. It goes without saying that as an inexperienced sole developer, these improvements are not based on functionality and code efficiency etc. This would be intangible. Improvements are primarily focused on making particular features more relevant, accessible and usable. The three services under examination are significantly

more advanced and expansive than this system is at the end of development. Therefore, features which these services include but this system does not will not be examined. So forth, the services will be examined purely under the scope of this system. That is, basic services (overview, ability, equipment) integration; map services (OS map, GPX file and map feature) integration; and, statistical processing (route and mountain information, and personal data processing).

As an alternative to formal empirical methodologies, Nielsen (1994) proposes a critique-based method based on a heuristic evaluation which involves analysis based on areas of expertise. A heuristic evaluation of one's own system is also argued to be a useful method of allocating time to minor issues before final user testing. Nielsen also claims that the optimal number of 'experts' assigned to an evaluation is three-to-five in order to find the 'optimal' number of issues relative to the cost-benefit analysis. In this case however, one examiner is used for obvious reasons. Each issue is individually listed and valued against the set of ten heuristic factors and assigned a severity rating, as seen in Table 2.1.

The analyses conducted subsequently are not exhaustive however, are relevant to the context and features in this system. There is no heuristic evaluation for Garmin Connect as any differing functionality from Strava is more advanced than that which this system accounts for and therefore, does not need to be evaluated.

Heuristics
H ₁ : Visibility of System Status
H ₂ : System-Real-World Match
H ₃ : User Control & Freedom
H ₄ : Consistency & Standards
H ₅ : Error Prevention
H ₆ : Recognition Rather than Recall
H ₇ : Flexibility & Efficiency of Use
H ₈ : Aesthetic & Minimalist Design
H ₉ : User Recognition, Diagnostic & Recovery from Error
H ₁₀ : Help & Documentation
Severity Ratings
S ₀ : Don't think it is a usability problem
S ₁ : Cosmetic issue; repair in additional time
S ₂ : Minor usability problem; allocate low priority to repair
S ₃ : Major usability problem; allocate high priority to repair
S ₄ : Critical error; repair immediately

TABLE 2.1: HEURISTICS & SEVERITY RATING

2.2.1 WALKHIGHLANDS

The heuristic evaluation for Walkhighlands is listed under Figure A1 in Appendix 1.

2.2.2 STRAVA

The heuristic evaluation for Strava is listed under Figure A2 in Appendix 1.

3 DATA PROCESSING & METHODOLOGY

3.1 DATA ACQUISITION

3.1.1 LOCATION

3.1.2 SELF-DECLARED ATTRIBUTES – ABILITY

Body statistics: height, body mass, body fat, muscle mass

Basics: average resting rate, 5k run average rate, 30 mile ride average rate, 10 mile hike average rate, 20 mile hike average rate

3.1.3 SELF-DECLARED ATTRIBUTES – EQUIPMENT

3.1.4 WEATHER

3.1.5 ROUTE DATA

Overview no longer gives an overview of user stats on routes, it now breaks down the routes.

3.1.6 REGIONAL DATA

3.1.7 LANDMASS DATA

3.2 DATA MANIPULATION & OUTPUT

3.2.1 ROUTE OVERVIEW

Basics: distance, elevation gain, estimated time (based on ability, equipment), estimates energy output (based on user stats, ability)

Difficulty rating: playground, normie, enthusiast, trad, gigachad

CONQUER ROUTE

3.2.2 WEATHER SYSTEM

3.2.3 'RANGER' CALCULATOR

Route recommendations and suggestions

Risk indicators: lack of visibility (based on weather), misdirection (based on weather and terrain), falling off cliffs and ridges (based on weather and terrain), rock fall (based on terrain), rock kick-back (based on terrain), dehydration (based on weather and duration etc.), sunburn (based on weather and duration etc.), confrontation with livestock etc. (based on terrain and terrain type), plants and allergens like moss and ferns etc. (based on terrain and terrain type)

Winter risk indicators: lack of visibility - white-out (based on weather and terrain), cornices (based on weather and terrain), avalanche (based on weather), ice fall (based on weather), hypothermia (based on weather and ability)

Recommended gear: based on all route factors

Recommended climbs (based on terrain and equipment): trad, sport, top rope, bouldering, free tool, free solo (suggested reading table)

Ranger graphs: elevation profile, speed input, power input, heart rate input; Select: Constant Speed (Max); Output: Required Time and Power; Select: Constant Speed (Average); Output: Required Time and Power; Select: Constant Power Output; Output: Required Speed and Time; Input: Target Speed; Output: Required Time and Power; Input: Target Power Output; Output: Required Time and Power; Input: Target Energy Output; Output: Required Time and Power

3.2.4 REGIONAL & LANDMASS GENERAL

3.3 INFORMATION ACQUISITION & OUTPUT

3.3.1 PURPOSE

3.3.2 KEYS & SUGGESTED READING

4 SYSTEM REQUIREMENTS

System requirements gathering is essentially the stage at which the concept of a system's functionality is aligned with real-world user desires (requirements). This stage provides a context for creativity and a blueprint upon which this can be mapped. Although the founder of a system or concept may have a clear vision of the intended outcome of their development, understanding what final users need and want allows the creator to constantly tailor development. This may extend to how they and/or the software could/should gather information, store data, transact data, and output data/information.

4.1 SCALE & SCOPE

I repeat, this is a [solo project] so there are some restrictions regarding the overall scalability of the project as a whole. If this system were to be designed by a team of professionals, it would be very large-scale and offer much expansible functionality. However, due to the number of personnel assigned to the task (a.k.a. me), the time constraint, and budget constraint, the system finds itself with two major general down-scales: [1] there is no user-data back-end, meaning user accounts are unavailable on this system, which is acceptable as the static functionality of the site is most relevant; and, [2] the sample region for data collection is significantly smaller than that offered by other services, which is also acceptable as adding a wider scope of data (to the master JSON in this case) would only consist of repeating the same patterns perpetually. As this system does not aggregate this data for any form of cross-sectional statistical analysis, larger sample sizes become irrelevant after a certain point. Overall, this implies these factors are not directly related to any implemented requirements. Many of the specifics of these restrictions, and others, are discussed latterly in the *Evaluation* section.

In an ideal world, users would be able to expand their scope of interactions within the system (i.e. different sports with a wider and varying array of attributes), and the individual scale of these. [This means that] as sports differ, attributes and statistics differ, and information and guidance differs; offering a better-rounded service. Ideally, users should have the opportunity to fully customize their experience however, to do this on such a scale feasible with the development team available (myself) would be a significant over-effort for an under-achievement. This is why user accounts have been disregarded. This decision helps keep the system and implementation of requirements

more manageable and makes it easier to achieve a polished product within the time constraint.

4.2 GATHERING & PRIORITIZATION METHODOLOGY

In this scenario, there are three methods of requirements gathering and inference. The first is the purely user-centered method. As I am surrounded by people who share an interest in the form of this system, including professional developers, student developers and various other [NPCs] of the sort, they act as an accurate representation of market users as they share the same attributes. The second mode is alike however, is argued to be subject to various aspects of contextual bias. This consists of creative direction explicitly from the creator, me. In this context, these inputs will generally align with those of the formerly discussed however, due to the bias, is not considered a viable user-centered method unless used in conjunction with others affirmative of said criteria. The final method involves basic inference from the *Research & Evidential Background* section. That is, much of the material investigation of Walkhighlands, Strava and Garmin Connect in this section highlights areas for linear development. These refer to aspects which generally do not require user-centered input and must be developed purely functionally.

Many aspects of this system are purely functional and exist to serve an objective purpose. For example, relaying GPX and JSON data related to components such as route and weather information. [This means that] the primary functionality of the system is majorly accounted for in the latter of the three requirements gathering methods, in that the goal of the system is to create wider-scoped versions of much of the existing content in the explored areas. Therefore, the two former methods generally account for improvements which can be implemented upon these predecessors throughout development, which are primarily focussed on enhancing the user's experience. Myself and the discussed group of experienced others are a credible source for this, considering the scale of the project.

Randomly assigning requirements to development would be irresponsible. To help better-address the importance of the components of the required functionality of a system, requirements should be analysed using some form of hierarchical tool which highlights a clearer path for the development process. In this case, the Must-Have/Should-Have/Could-Have/Won't-Have (MoSCoW) methodology is selected for this purpose.

This helps differentiate between what functionality is essential to make the system run as intended, what functionality is required for optimization of the system, and what is required for additional enhancements. This approach generally shows functional requirements to average at the top-priority end and non-functional requirements to be distributed further down the hierarchy. With regards to the framework itself, note that:

- *Must-Have* implementations refer to aspects of the system which must be present in order to make it basically functional and behave as it is intended and as the user desires;
- *Should-Have* implementations refer to features of the system which should be implemented in order to make the essential features of the system more accessible and useable to the masses. They may also exist to improve efficiency, but [fly Under the Radar] and therefore go unnoticed by the user. They may offer additional functionality which makes the system more unique (or something of the sort) and therefore, more ‘creative’ and attractive to users.
- *Could-Have* implementations refer to aspects which may become present or relevant during the development of the former two. They may add additional functionality or usability to existing aspects or simply add final touches to the system overall. They are sometimes more contemporary.
- *Won't Have* non-implementations refer to aspects of the system which aren't necessarily impossible or are of a nature which the system ‘can't have’ but, which will probably be omitted or postponed due to constraints such as time, manpower, technical ability, finance, etc.

There is no formal client base for this system which means that no face-to-face client-oriented interviews or surveys can take place with regards to determining specific user desires. However, the aforementioned three-method protocol leads to the subsequently discussed requirements. To reiterate, the following list of requirements is generated through discussion of the desires of [1] an existing group of users of systems alike (some of whom are developers), [2] myself, another existing user of the sort, and [3] analysis of other services. This list is of course not exhaustive, as there is always room for perpetual development. However, it does account for every currently visible desire and possibility given the constraints.

4.3 REQUIREMENTS

As discussed, it's important at this stage to clearly differentiate between essential functionality for the foundations of the system, generally accounted for in *functional requirements*; and functionality which more contemporary, generally accounted for in *non-functional* requirements. This is often considered 'making things *work* and making things *relevant*'. It is also a useful method from which to infer associations between elements of the design and construction stages of development. Retaining these requirements, their position, and hierarchy at the center of development throughout the process allows accurate amendment of software and/or requirements along the way as more functionalities and possibilities become apparent and tangible or alternatively, further from reach.

4.3.1 FUNCTIONAL

Requirements here are presented in a numeric format. This does not refer to any hierarchical order, it simply creates a reference point for user stories. Hierarchy remains determined by the MoSCoW methodology. So forth, as required by the three discussed groups, users and the system *must have* the ability to:

1. Access the user's current location upon various types of request
2. View an 'overview' of recommended routes
3. Accept inputs related to user 'abilities' and compute and display results based on them
4. Accept inputs related to user 'equipment' and compute and display results based on them
5. View a weather forecast breakdown using real weather data
6. Display an Ordnance Survey map

Additionally, users and the system *should have* the ability to:

1. On OS Map, alternate between 'OS Leisure' (primary/default), 'OS Road', and 'OS Outdoor' topographical structures
2. On OS Map, display grid markers at center-pan

Additionally, given the various constraints, users and the system *could have* the ability to:

1. Create an account and have their data stored
2. Protect user data using an appropriate authentication and security system
3. Store and re-use user attributes such as the aforementioned ‘ability’ and ‘equipment cache’
4. Choose ‘priority attributes’ relevant to their routes which are stored and used to generate more relevant route recommendations etc.

Additionally, users and the system *won't have* the ability to:

1. Store route-relevant data and display them as ‘historically completed routes’, or something of the sort, under the aforementioned ‘overview’

4.3.2 NON-FUNCTIONAL

To ensure a stable, trustworthy, useable and relevant system; users and the system will have the ability to / have the capacity to:

1. Be written in such a manner which allows full compatibility with the majority of web browsers (this could effect markup languages, font packages, JavaScript libraries, etc.)
2. Be written in such a manner which allows full scalability between desktop and mobile use
3. Offer the user various descriptive background [pieces], otherwise referred to as ‘key’s and ‘suggested reading’s which provide aid to users’ background knowledge, understanding and decision making
4. Offer the appropriate combination of relevant graphic design and actual functionality
5. Adhere to the appropriate accessibility standards, with reference primarily to graphic design and system structure

4.3.3 USER STORIES

‘User stories’ are often an effective method of contextualizing requirements and presenting how they can and will be interpreted and used by the final user. Putting yourself in the hypothetical context of a user is a good method of delivering basic feedback on how to apply a solution to a requirement. Additionally, it helps identify possible negations or irrelevant content. So forth, the user stories found in Figure A3 in Appendix

1 follow the syntax: “I wish to be able to <interact_with_feature> in anticipation of <returned_result> which will provide me with <payoff>”. Development of the solution to these user stories listed in the figure are prioritized using the metric seen in the *Priority* column.

4.4 USER ACCESSIBILITY

This system is available as a website application, it could be accessed and used by anyone with access to the internet. Therefore, it is to some degree essential that users of all types are welcomed to the site, even if they have no prior knowledge of the industry or field. More importantly, the graphic design and methods of interfacing in the user interface should be accommodating of users with possible imparities, such as a [severe mental imparity], by taking a user-centered approach which considers visible accessibility controls which allow these users to interface with the system more easily. Accessibility does not however only refer to disability. Ensuring optimal accessibility also accounts for factors such as implementing clean and concise CSS and graphic design for simple and intuitive navigation, stimulation, relevance, etc. These approaches are widely considered and implemented.

In context, whether using the desktop or scaled mobile version of the site, the user interface implements many ‘friendly’ universal standards. For example, it primarily uses the readable and elegant Sans-Serif font Audi AG and Serif font Garamond, with suitable font-sizing and color etc. Hue, saturation and luminance are appropriately considered/assigned to create [maximal optical pleasure] and proper readability with the correct contrast between elements. The well respected font-family Font Awesome, which is universally recognised, to display various standard symbols. In theory, these considerations lead to more efficient understanding and transitioning between elements of the site.

5 SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

5.2 USER INTERFACE

5.2.1 LOGICAL DESIGN

5.2.2 GRAPHIC DESIGN & COMMUNICATION

5.2.3 INTERFACE TREE

5.2.4 USE CASE DIAGRAM

5.2.5 PRIMARY USE CASE EXAMPLES

5.3 DATA STRUCTURE

6 SYSTEM CONSTRUCTION & IMPLEMENTATION

6.1 DEVELOPMENT & LANGUAGES

6.1.1 DEVELOPMENT ENVIRONMENT

6.1.2 FRONT-END & USER INTERFACE

6.1.3 SEMI-REAR-END & SUPPORTING DATA

6.2 APPLICATION PROGRAMMING INTERFACES (APIs) & LIBRARIES

6.2.1 GEOLOCATION

6.2.2 ORDNANCE SURVEY

OS Road (1 : 250 000), OS Landranger (1 : 50 000), OS Explorer (1 : 25 000)

6.2.3 OPEN WEATHER

6.2.4 LEAFLET

Leaflet JavaScript library for maps

6.2.5 MAPBOX

Mapbox script for GPX-GeoJSON conversion

6.2.6 CHART.JS

Chart.js JavaScript library for ranger

6.3 PROJECT MANAGEMENT

6.3.1 LIFE-CYCLE & TIMING

6.3.2 SUPPORTING TOOLS

7 EVALUATION

7.1 REQUIREMENTS

7.2 DESIGN, CONSTRUCTION & IMPLEMENTATION

7.2.1 HEURISTIC EVALUATION

7.2.2 USER ACCEPTANCE (UAT), ACCESSIBILITY & USABILITY

7.2.3 'PROTOTYPING' & CONSTANT EVALUATION

7.3 ADDITIONAL TESTING

7.3.1 'TEST-DRIVEN' DEVELOPMENT

7.3.2 UNIT TESTING

7.4 CONSTRICTIONS

Restricted to Glen Coe and Glen Etive

8 DEVELOPMENT CONCLUSIONS

A MICHAEL MANN PRODUCTION
©1984 ORION PICTURES CORPORATION
ALL RIGHTS RESERVED
DOLBY STEREO™ IN SELECTED THEATRES

APPENDICES

Testing questionnaires, test cards (exp. result etc.), heuristic evaluation of system

APPENDIX 1: FOUNDATIONAL MATERIAL

FIGURE A1: HEURISTIC EVALUATION – WALKHIGHALNDS

DESCRIPTION	VIO.	SEV.	PROPOSED SOLUTION
BASIC FEATURES			
Sparse Navigation: There is not much order to the site navigation and in places there is a lack of relevance and hierarchy to the options. For example, some sub-sets of pages/features are listed in navigation on the same 'tier'; whether that being represented by the link having the same alignment as its parent, or something of the sort	H ₃ , H ₄	S ₃	Correctly use HTML and CSS to display the hierarchy of pages and features in this system
Difficult Navigation: Some essential literature relating to components such as ability and equipment, as featured in this system, is difficult to find and requires a long path of navigation through pages and their children. Often some literature is simply linked through an in-body hyperlink (i.e. in a paragraph) and not even allocated its own title etc.	H ₃ , H ₄	S ₃	Correctly use HTML and CSS to display the hierarchy of pages and features in this system. Ensure no relevance is lost in masses of text
MAP FEATURES			
Feature Separation: The OS Maps displaying Munros and Corbetts etc. are displayed on different pages of the site, with navigation links between them. This means that interaction with features is limited to the point where all a user can do is view the information provided about it upon a click and follow further links from there. They cannot for example, toggle on/off different features while interacting with the map meaning isolating different features on a GPS route a user is viewing is impossible.	H ₃ , H ₇	S ₃	Implement a toggle feature for access to different map feature when the user desires, as listed in requirements
STATISTICAL FEATURES			
...			
TOTAL VIOLATIONS: 3			
EVALUATOR: Lewis Britton			
PLATFORM(S): Brave Browser (Desktop), Brave Browser (Mobile)			

TABLE 8.2: HEURISTIC ANALYSIS

FIGURE A2: HEURISTIC EVALUATION – STRAVA

DESCRIPTION	VIO.	SEV.	PROPOSED SOLUTION
BASIC FEATURES			
Weather Misuse: Weather data is only available as part of complete routes. I.e. activities which are complete and have been saved. Strava use a weather API so it is hard to understand why they would not integrate this as a basic summary feature on the home page or on a profile etc. Historic weather data, i.e. that printed on complete routes, is the most irrelevant use of such as, come on, the user already knows what the weather was like on their routes and if they want to show off a 20hr hike in a snow storm or something, upload a picture	H ₃	S ₃	Include a comprehensive weather system, as listed in requirements
MAP FEATURES			
Map Layers: Although the Mapbox map seen throughout Strava is used very effectively in hosting many dynamic layers such as customizable GPX routes with terrain' and athlete-specific attributes, it is just a basic map with contours, not a full topographical map. This significantly limits interaction with map features and interpretation as it simply doesn't show many of the relevant aspects to hiking	H ₂ , H ₃	S ₃	Utilize the OS Map API instead of a more basic one to ensure full access to topographical mapping, as listed in requirements
STATISTICAL FEATURES			
...			
TOTAL VIOLATIONS: 2			
EVALUATOR: Lewis Britton			
PLATFORM(S): Brave Browser (Desktop), Brave Browser (Mobile)			

TABLE 8.3: HEURISTIC ANALYSIS

FIGURE A3: USER STORIES, PRIORITIZATION & SOLUTIONS

SCENARIO	REQ.	RELEVANCE	PRIORITY*	SOLUTION
FUNCTIONAL – MUST-HAVE				
“I wish to be able to [use the current location of my device] in anticipation of [seeing my location printed on a map] which would allow me to [navigate towards map features relative to my location]”	1	UI OS Maps	1	Utilize <i>Geolocation</i> API to fetch location of user’s device
“I wish to be able to [use the current location of my device] in anticipation of [weather forecast results for my location] which would allow me to [quick-access my weather forecast, likely while on-route]”	1, 5	UI Weather Service	1	Utilize <i>Geolocation</i> API to fetch location of user’s device and place in weather function
“I wish to be able to [quickly view a summary of suggested routes] which would allow me to [see all the relevant attributes of a route on one return] and therefore, [select routes with less thought, time and effort]”	2	UI Route Summary	1	Include a section on the home page with this/these information/statistics
“I wish to be able to [include attributes related to my ability] which contribute to my [suggested routes] therefore, making them [more relevant to my needs]”	3	UI Ability Service	1	Include a section which accepts relevant user inputs and returns route suggestions as required
“I wish to be able to [include attributes related to the equipment I own] which contribute to my [suggested routes] therefore, making them [more relevant to my needs]”	4	UI Equipment Service	1	Include a section which accepts relevant user inputs and returns route suggestions as required
“I wish to be able to [seek locations by their name] in anticipation of the return of [weather forecast results related to their coordinates] which would allow me to [plan my routes more effectively]”	5	UI Weather Service	1	Include a section which accepts location-based inputs, whether this be pre-defined coordinates of locations or the user’s current location as discussed previously, to display weather forecast results for these locations

“I wish to be able to [view an Ordnance Survey map with OS Road (1 : 250 000), OS Landranger (1 : 50 000), and OS Explorer (1 : 25 000) layers simultaneously, alternative upon zoom] in anticipation of [viewing initially basic features such as my location as a symbol on the map and center-pan coordinates adjacent to the map] therefore, [making location seeking more interactive and understandable]”	1, 6	UI OS Maps	1	Utilize <i>Ordnance Survey</i> API to host the map Utilize <i>Leaflet</i> JavaScript library to initialize and interact with the map
“I wish to have the option to [interact with the OS map] which allows me to [toggle on/off display of features], allowing me to [view only specific features I wish to include on my routes]”	6	UI OS Map	1	Utilize <i>Leaflet</i> JavaScript library to display and toggle on/off symbols representing the location of various geographical features such as Munros, Munro Tops, Corbetts, and Corbett Tops
“I wish to have the option to [interact with the OS map] which allows me to [toggle on/off GPS routes based on search criteria] therefore, [making route results relevant to my subjective choice], as opposed to other factors such as recommendations based on my ability and equipment, or raw search”	6	UI OS Map	1	Utilize <i>Leaflet</i> JavaScript library to display and toggle on/off GPS route (GPX data) layers over the map, relative to various search criteria Utilize <i>Mapbox GL-JS</i> JavaScript library to convert GeoJSON files to GPX
“I wish to be able to [search for routes close to my current location] which returns the [closest relevant GPS overlay] to my location, making it easy for me to [make a fast decision if I just want a close project]”	6, 1	UI OS Map	1	Utilize <i>Geolocation</i> API to fetch location of user’s device and compare to the location of all listed projects
FUNCTIONAL – SHOULD-HAVE				
“I wish to be able to [interact with the OS Map] in order to [change its layers] which will allow me to [interpret the features and contours of the land in different ways]”	1	UI OS Map	2	Include some form of input which <i>Leaflet</i> JavaScript library to allow the user to change primary map layer

“I wish to [view grid markers at the center of the OS Map] which, when panning, will [pin-point the center coordinates displayed on the map] therefore, [making it easy to view, locate and pan to and from features and locations etc.]”	2	UI Map OS	2	Include basic HTML and CSS which displays reticle
FUNCTIONAL – COULD-HAVE				
“I wish to [have my data stored] in the system so I can [access it down the line] to [use it in various site features]”	1	Back-End	3	Implement a back-end database system to store and interact with user data
“With regards to [data access], I wish to be [assured of appropriate security measures] implemented by the system controllers which will give me [peace of mind] regarding data protection	2	Back-End	3	Implement appropriate authentication and data protection measures in the system”
“I wish to be able to [use and reuse attributes related to my account] in order to [tailor results to my own abilities and equipment], for example; allowing me to [gain a more accurate understanding of what I’m able to do] with less effort”	3	UI Back-End	3	Allow already-implemented features such as user ability and equipment services to access and use stored user data, as opposed to using basic inputs
“I wish to be able to [choose ‘priority attributes’] relevant to data stored upon my previous routes which returns [more relevant route recommendations] to historic ones, relevant to what I have and haven’t done based on my preference; allowing me to [more dynamically choose routes]”	4	UI Back-end	3	Implement appropriate measures
FUNCTIONAL – WON’T HAVE				
“I wish to use a GPS device such as a watch or cellular device to [record GPX data] which can be used to [generate route information] and [display useful route records] which I can interpret and benefit/learn from”	1	Back-End	4	(Will not) implement functionality to accept GPX uploads and transform relevant components into human readable information
NON-FUNCTIONAL				

“I wish to be able to [use the site on any browser] therefore meaning I can [view all content] regardless of what browser platform I use, meaning I’ll be able to [consistently interact with the functionality]”	1	UI Browser Support	5	Ensure all languages, packages, libraries, etc., are using the most versatile syntax and have aggregate browser support
“I wish to be able to [use the site on any device platform] therefore meaning I can [view all content] regardless of what device I use, meaning I’ll be able to [consistently interact with the functionality]”	2	UI	5	Implement the correct CSS protocol to allow the site to dynamically scale to various device sizes; using appropriate adjustment and hiding/showing of content
“I wish to [view additional] literature based upon [information regarding topics I wish to explore]; [enhancing my knowledge of the components of the sport and my decision making]”	3	UI Research	5	Implement appropriate bodies of text relevant to areas upon which they may be if use to the user
“I wish to [view elements of graphic design] which [show relevance to the areas in which they appear] therefore, stimulating, engaging and immersing me further in the sport]”	4	UI	5	Use CSS to implement the correct elements and principals of graphic design, primarily color schemes and shape/geometry, which are relevant to the outdoor context Use graphic design tools (such as Serif PagePlus and GIMP) to create and implement relevant features such as logos, images and renders, which are relevant to the outdoor context
“I wish to be able to optimally interact with the site when it comes to navigation and viewing experience, regardless of any impairment or things of the sort”	5	UI Research	5	Implement appropriate design protocol to make this experience as easy as possible

*: ‘Must-Have’ = 1; ‘Should-Have’ = 2; ‘Could-Have’ = 3; Won’t Have = 4; Non-Functional = 5

Various Sources (2022)

FIGURE A2:

(Draper, Paudyal, 1999)

APPENDIX 2: FOUNDATIONAL MATERIAL

AUTHOR’S NOTE

The following content is manufactured by myself in aid of basic understanding of the background to contexts, data and methods present within this study. It is presented as teaching material, in a format I would output if in such position.

FIGURE B1: LINGUISTICS OF ‘L^AT_EX’

L^AT_EX (or LaTeX, even latex (Donald E. Knuth’s more recent installment of T_EX)) is usually pronounced /la:tɛk/ (‘lah’) or /leɪtɛk/ (‘lei’/‘lay’) in English (that is, not with the /ks/ pronunciation English speakers normally associate with X, but with a /k/). The characters T, E, X in the name come from capital Greek letters tau, epsilon, and chi, as the name of T_EX derives from the Greek: $\tau\epsilon\chi\nu\eta$ (skill, art, technique, precision); for this reason, Donald E. Knuth promotes a pronunciation of /tɛk/ (tekh) (that is, with a voiceless velar fricative as in Modern Greek, similar to the last sound of the German word “Bach”, the Spanish “j” sound, or as “ch” in a Scottish ‘loch’).

FIGURE B2: DON. KNUTH’S COMPUTER MODERN UNICODE (CMU) FONT FAMILY

Serif	Sans Serif	Monospaced
CMU Serif Roman	CMU Sans Serif	CMU Concrete
CMU Serif Bold	CMU Sans Serif Bold	
<i>CMU Serif Italic</i>		<i>CMU Concrete Italic</i>
<i>CMU Serif Oblique</i>	<i>CMU Sans Serif Oblique</i>	<i>CMU Concrete Oblique</i>
CMU SERIF SMALL CAPS		CMU CONCRETE SMALL CAPS
<i>In the presence of traditionalists, a suitable alternative to Donald E. Knuth’s Computer Modern Unicode font family may be considered: Andale Mono.</i>		

FIGURE B3: WHY MY PRE-TITLE’S RIGHT AND YOU’RE WRONG

I have received numerous comments which anyone would regard naïve and under-educated regarding my pre-title of this study: *AG436 Dissertation Coursework Assignment*. The argument originates in the ‘Coursework Assignment’ portion. People argue that a dissertation ‘is not’/‘does not have’ an assignment. Not only is this poor characteristic recognition, it is semantically wrong. *AG436: Dissertation* is a class

just like any other. However under this class, there are no lectures, no tutorials and therefore no exams as there is no [taught] content. Do not confuse this with the class ‘having no content’ though. AG436’s content is apparent through literature of the student’s choice. Therefore, it is possible for a ‘coursework assignment’ to be based on this. Hence, any further comments are null.

BIBLIOGRAPHY

- [1] Nielsen, J. (1994). *Heuristic Evaluation*. Usability Inspection Methods, John Wiley & Sons
- [2] Abraham, A., Ikenberry, D. L. (1994). *The Individual Investor and the Weekend Effect*. Journal of Financial and Quantitative Analysis. Volume 29. Issue 2. Pages 263-277.
- [3] Investopedia. (2020). *What's the Most Expensive Stock of All Time?* Available At: <https://www.investopedia.com/ask/answers/081314/whats-most-expensive-stock-all-time.asp>. (Accessed: 08/10/2020).