Quark – A gamified time tracking application and website  
Assignment Report

CI536 – Integrated Group Project

18th April 2022

Lab Tutor: Karl Cox

Group Members:  
Kim Lam (19823013),  
Karan Vani (19800011),   
Fin Watling (20838503),  
Ben Edmondson (20812940),  
Danil Bronnikov (),  
Jazer Barclay (20837308)

**Table of Contents**

[1. Introduction 1](#_Toc99750303)

[2. Methodology 2](#_Toc99750304)

[3. Product Description 3](#_Toc99750305)

[4. Legal, Ethical and Security Issues 4](#_Toc99750306)

[5. Evaluation of Fitness for Purpose 5](#_Toc99750307)

[6. Critical Review 6](#_Toc99750308)

[7. References 7](#_Toc99750309)

[8. Appendix 8](#_Toc99750310)

[8.1. Appendix 1 8](#_Toc99750311)

[8.2. Appendix 2 8](#_Toc99750312)

[8.3. Appendix 3 8](#_Toc99750313)

Introduction

Summary of project goals, and achievements (One page)

This report aims to…

* Taken from the module outcomes:

1. Function as a computing professional, with understanding of professional, legal, ethical and information security issues in a business and commercial context
2. Research and analyse problem situations; model the requirements of possible solutions before implementation
3. Work effectively in an agile team, plan projects and manage time
4. Effectively manage the development process, adhering to an agile project process
5. Design, specify and implement usable ICT solutions which address the problems and requirements
6. Assess the fitness for purpose of such an ICT solution

Our project is…

* Detail out the project as a whole going into surface level details that a project manager/owner would request
* Problem description – What is the business opportunity or problem to be solved
* Requirements description – What do we need to change in our jobs, lives, real world to achieve the business opportunity or solve the problem
* Specification description – What properties do we expect a computer to have and how should it behave in order to achieve our needs

We accomplished…

* Cover what we got completed in the 12 week timespan (design, development, testing wise) as the minimum viable product

Methodology

Explanation of choice of relevant methodologies such as project   
management, surveys, development tools/environments, testing and UX approach.

To develop this application and website we chose to use the agile development process. This was chosen for multiple reasons. With all developers have a …

|  |  |
| --- | --- |
| Methodology | Why |
| Agile Development process | * More even workload can be assigned by project manager * Higher general knowledge of all fields required for project manager raises the difficult however if all are known, support can be given to all developers when needing help. Also makes distribution of workload easier knowing what is required for each stage of the work process cycle * More manageable meeting schedule compared to scrum * Burndown chart would be redundant with GitHub tracking commits and Trello with design and documentation |
| Research | * Checked for if our app already exists in some capacity * Used other similar apps from the same field to see how they are effective * (room to improve with survey before design and development) |
| Design | * Top down approach going from very plain sketches of overall layout without any detail * Moved to more in depth designs which influences user stories getting from one screen design to another * Further this via wireframes which are used during the development process where development can match the wireframe design * Aimed for 3 main factors: simple to use, intuitive and minimal. * Reason for these factors is to aid in its function: as a passive time tracking tool which records valuable data to consider work ethic and improve with a more healthy workflow |
| Development (General) | * Used GitHub to track all code during the development process. This allows for collaboration of code as well as auto-deployment * Use of Trello which all developers have full access to allows for autonomy in keeping track of an individual’s progress and in adding more specific issues or requirements when issues arise. * Acquired a virtual private server in the cloud with a front facing domain name (quark.rocks) to both test during development and deploy the live app for public use * Choice of a broad tech stack to allow all developers to use their preferred or currently studying area in order to build efficient and clean code. * All languages and/or tech stacks are understood by the project manager allowing close pair collaboration to prevent a specific part of the project becoming undeliverable * A primary focus on documentation including access to designs within Trello and within code as comments to ensure code doesn’t ‘rot’ over time where developers don’t understand what they wrote after time has passed |
| Development – Modular design | * Use of a modular design allows easier extensibility where a monolithic design would box developers into a specific tech stack. * A greater consideration of what is exposed to other parts of the app is useful to ensure security between modules and allows for switching technology to more suitable ones when required. |
| Website | * Built using HTML, CSS, JS with minimal use of external libraries * Done to ensure we control the majority of the code written where changes can be audited compared to external libraries which can update without any checks and may cause instability in the app * Functionality separated from the website via the API which allows for a more locked down access to the database however allows for other developers to integrate with the app using their own custom extensions. * Can be tested using suites such as lighthouse by google and w3c |
| App | * Use of Java for its cross-platform nature and ease of development * The JUnit library allows for test driven development process * Ant build scripts allow for easier CD/CI compared to platforms such as Jenkins and can easily be offloaded to GitHub actions as a self-contained image * Maven project setup makes setup on other developers systems easier since configurations are managed via the package manager |
| Database and API | * Accessed via the API to prevent SQL injection attacks via the website and through sanitising requests * Used PostgreSQL for its built-in function and trigger states where processing can occur within the database server for final checks * Use of entity relationships to design tables which reduces overloaded tables and reduces duplication of content * API can perform checks and handle requests before they access the database saving database processing and optimising for high traffic |
| Testing | * Each area includes their own testing process however all can be tested by the developer first * This could include black and white box testing techniques * Java has JUnit where tests are written for error states, edge cases and expected results * Website is more focused on UX where a user would be required to test it from different perspectives. * User accessibility can be tested using lighthouse and to ensure best practices are used, w3c can be run on the desired pages * The API can be tested directly by the developer by submitting error states, edge cases and valid results * Once stable, move to other students in the group to see what they think of the app and website in both UX and UI * After revision a beta can be released to the public to try and give feedback on * Each further test group would incur a revision change to improve based on the feedback we receive |

Product Description

What has been created including a link to a short video   
demonstration, where relevant, explanation of requirements, design,   
implementation and evaluation of choices and outcomes;

The demo:

Demo the app first

Move onto website to view

Gloss over the API access and how this can be integrated into further user developed apps

Go into the MVP requirements of the application and how we reached these

Move onto the stretch goals we wanted to add by showing the designs

Say how this could be extended with further revisions using our agile development process

Segue into the design process used to get to this point and how that went

Use the wireframes to show how these moved into the final design – the ease we had with accurate designs to build from reduced tiem take to develop by up to half since design and development didn’t have to happen at the same time (context switching)

Mention the implementation we currently have and how it works compared to what we imagined at the beginning from the sketches

Note how this could be further improved upon even before the stretch goals and can be added in the next iteration of agile sprint

Legal, Ethical and Security Issues

A discussion of issues relevant to your project;

User data:

* Email addresses can be a phishing vector that other actors may wish to capitalise on. Made easier to access via an API would save them having to scrape them from the website manually.
* To solve this we made emails private from the interface and from the API. Also the login requires the user’s email to access the content. This makes brute forcing the login for any given user far more difficult with emails being vastly longer and more broad in characters.
* Correlation can also be a problem If a person can be correlated to a particular user then we could have the issue where they may be tracked via their own posts to the server. A malicious actor may decern that a person is away from home (aka at work completing units on the app) and may use this opportunity to do something nefarious while knowing they are away.
* To combat this we do not display nor record any user’s personal details. Their username is all that is displayed on the platform and is not required to login. This curbs shoulder surfing and key logging. Also a notification is given to the user when creating their account so they understand their username can be a reflection of their daily habits to other users so choose wisely who you share with.
* Note that data protection is at play here also!
* Passwords can be guessed. Having a sufficiently difficult password is important however limiting the user to a specific set can sometime hinder rather than help. Not to mention web browsers now recommend complex passwords that are then stored in a keychain.
* We do not limit the user on anything other than character length. This ensures that the user can have a sufficiently secure password while not boxing them into a specific set of combinations
* Another note would be to remove password entry and email a one-time passcode which allows access to a certain device for a certain amount of time. Does away with the password while also keeping the account secure

SQL injection attacks: Sanitise inputs and validate before running

Cheating the game part: Ensure token is valid, validate end time matches current clock time or close enough to prevent submitting bogus times. Also check that none overlap where you could multiply your points by resubmitting the same unit over and over (replay attack)

Evaluation of Fitness for Purpose

Describe the criteria by which your creation could   
be evaluated to evaluate its fitness for purpose.

* Is it intuitive? Can a new user pick it up and use it within a session (between 30 mins and 2 hours)
* Is it engaging? Measure user retention over a period of time
* Does it complete its purpose?
  + Do users gain valuable insights into their work practices?
  + Are users gaining healthier work habits taking regular breaks?
  + Do users improve over prolonged use of the app and from the stats they can view?
* Is it accessible? Can a disabled user access all features provided?
* Would a user want to be seen with it running? Is it trendy? Desirable by the niche?

Critical Review

Review success and areas for improvement, emphasise what has   
been learnt and how this would affect future projects

References

Example references here:

Consumer Rights Directive 2011/83/EU, Available at <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32011L0083> [Accessed 05 Jan 2021]

Department for Business, Innovation & Skill - Misleading and aggressive commercial practices (BIS/13/1114) (2013) Available at <https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/298969/bis-13-1114-misleading-and-aggressive-commercial-practices_a-new-private-right-for-consumers.pdf> [Accessed 06 Jan 2021]

Department for Business, Innovation & Skill - Public Views on Ethical Retail (BIS/14/907) (Updated June 2014) Available at <https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/322624/Public-views-on-ethical-retail.pdf> [Accessed 05 Jan 2021]

European Commission - Consumers: Commission proposes EU-wide rights for shoppers (IP/08/1474) (Updated 8 October 2008) Available at <https://ec.europa.eu/commission/presscorner/detail/en/IP_08_1474> [Accessed 05 Jan 2021]

Harry Brignull (2010) Dark Patterns – What are dark patterns [Online] (Updated 03 Jan 2021) Available at <https://darkpatterns.org/> [Accessed 07 Jan 2021]

Mike Monteiro - A Designer’s Code of Ethics (Updated 10 July 2017) Available at <https://muledesign.com/2017/07/a-designers-code-of-ethics> [Accessed 06 Jan 2021]

Paul Randall – Ethical CRO: The end of dark patterns (Updated) Available at <https://econsultancy.com/ethical-cro-the-end-of-dark-patterns/> [Accessed 07 Jan 2021]

The Consumer Contracts (Information, Cancellation and Additional Charges) Regulations 2013 (Part 4 - Protection from inertia selling and additional charges) Available at <https://www.legislation.gov.uk/uksi/2013/3134/part/4/made> [Accessed 06 Jan 2021]

Appendix

* 1. Appendix 1

Record of team meetings with information on what was discussed at   
meetings, decisions that were made.

* 1. Appendix 2

Source/Asset files. Where appropriate, a link should be provided to an external repository such as GitHub that contains the source code/asset files required to build/create the project. The location/content must be agreed with the   
LAB tutor.

* 1. Appendix 3

Any further resources you think are relevant to the project.