* choice of project and how it fits in with the modules you have studied \*\*\*\*
* background research and the way it has influenced your project \*\*\*
* your methodology and planning (Include your original project plan, together with any later versions or a discussion of any necessary changes to the plan) \*\*\*\*
* an assessment of the progress you made, problems encountered, their solutions and the lessons learned
* aspects of your work you are particularly proud of
* further areas for possible investigations or enhancements.
* Technical grasp
* Understanding of problem area
* Project management
* Report quality
* Evidence of learning
* Research effort.

smaller datebase objects

task completion by the serever

nno specific object validation

QRCode

JQM

Database objects

## Abstract

Very brief summary of the project, goals, products and outcomes (One paragraph)

Contents

[Abstract 1](#_Toc33616181)

[Introduction 3](#_Toc33616182)

[Methodology 4](#_Toc33616183)

[Project management 4](#_Toc33616184)

[Plan Development 5](#_Toc33616185)

[Risk Analysis 10](#_Toc33616186)

[Collection restoration 11](#_Toc33616187)

[Surveys 13](#_Toc33616188)

[Test Driven Development 16](#_Toc33616189)

[Product Description 17](#_Toc33616190)

[Design 18](#_Toc33616191)

[Requirements 26](#_Toc33616192)

[Implementation 26](#_Toc33616193)

[Custom Elements 27](#_Toc33616194)

[Document List 28](#_Toc33616195)

[Critical Review 29](#_Toc33616196)

[References 30](#_Toc33616197)

[Appendices 30](#_Toc33616198)

## Introduction

1. Summary of project goals, and achievements (One page)

The main goal of this project was to design and create a gamified application. Design is informed by research into the gamification and game design in general. I both successfully learned several technologies and built on knowledge I have gained throughout my degree and

When I set out to write the Workaholic application, I had no knowledge of modern JavaScript (JS) syntax. \*\*\*MSDN ES6 is the latest specification of JS has allowed developers to notate prototypes in the form of Classes. This syntax is something I have never used before and have gained exposure and learned a lot through developing this application. Over my course at university I have had modules on object orientated design and implementation. This enabled me to understand how objects work in this context which was a good starting point to allow me to understand how I would go about practically using this type of prototype notation.

Follow OOD

Follow Agile

Learn ES6

Learn custom elements

Learn firebase

Using jqm learned previously

## Methodology

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

### Project management

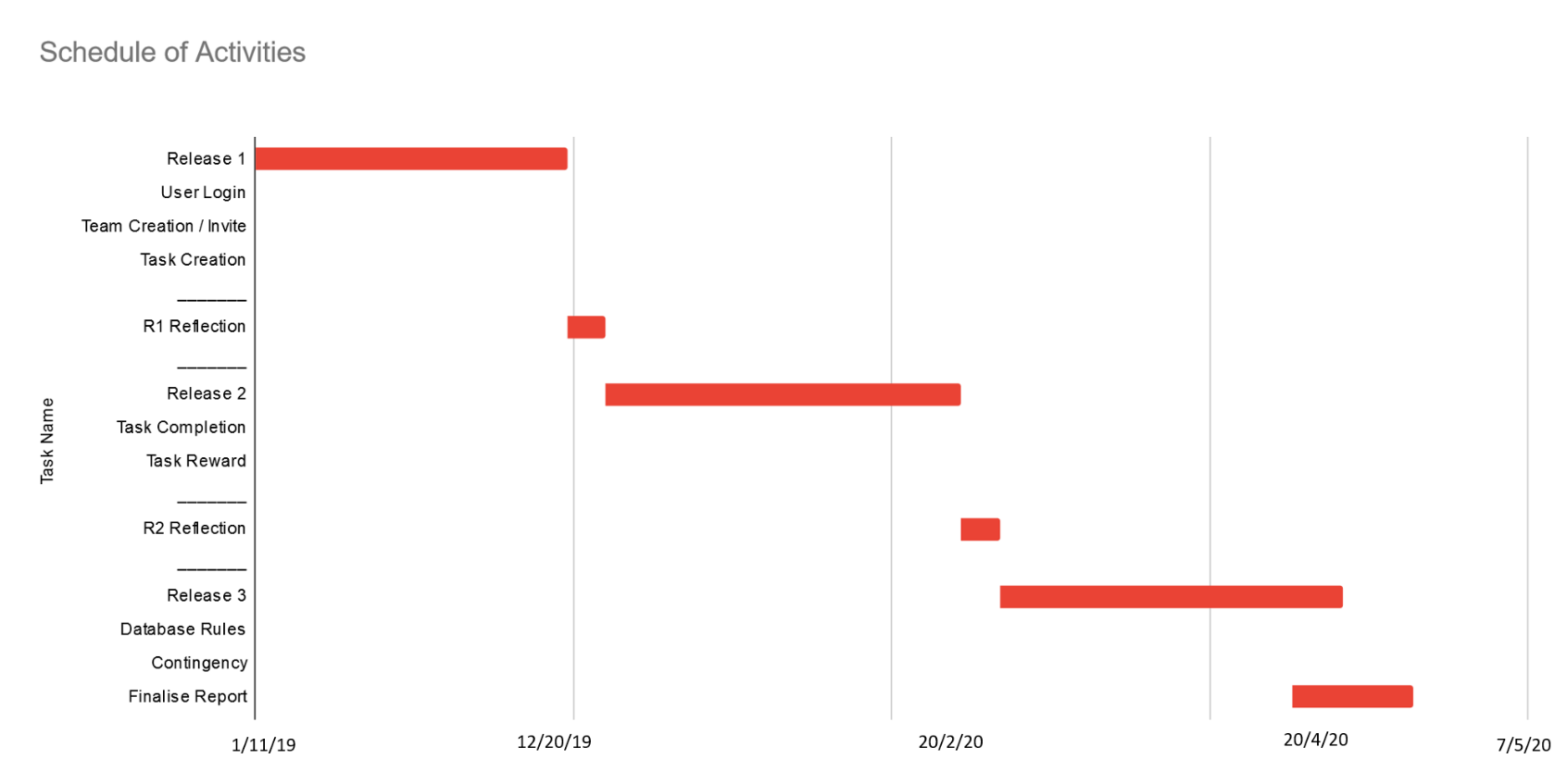
I knew from the beginning that this project will have significant amounts of functional dependency, for example, the user will need to be able to log-on in order to uniquely identify them and the documents they have the privilege to access. For this reason, I considered using a Waterfall approach to development as this would have the advantage of being able to strictly plan each of the individual dependencies. Due to my choice in following OOD waterfall became a much less attractive methodology. Being more compatible with OOD and having better mobility to deal with uncertainty, Agile has become my choice of development methodology.

Agile has much tighter feedback loops, this allows the testing of elements in the project as I develop, enabling change of design and reallocation of time, based on results of tests. This also allows for restructuring of the project if specific functionality takes longer to make than initially planned. Further research revealed Agile is considered to increase efficiency and the successfulness of a project (Serrador and Pinto, 2015)\*\*\*\*.

Developing the application in this way allowed for sprints to be arranged at the beginning of a two-week period. As I progress through the project, I kept a diary of the development trying to remain critical of both the process and the result. This diary was largely added to at the end of any day where I had done any development.

I also have kept running documentation on the duration of time spent on tasks in the sprint. This proved an invaluable way of analysing the process and making sure time was being spent in a constructive way.

### Plan Development

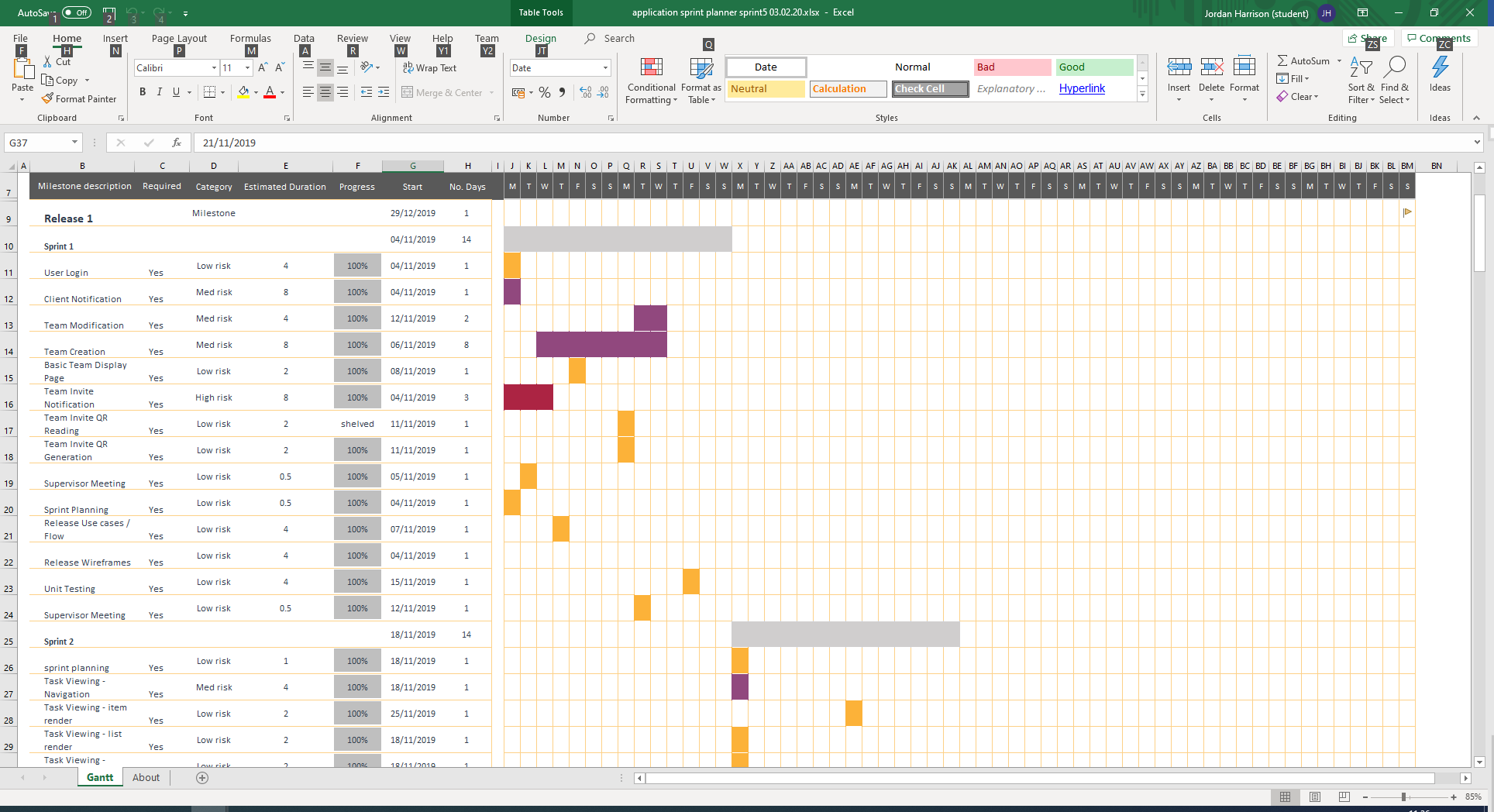
My plans in the initial stages of the project consisted of simply releases and functionality that I wanted to be in each release. This plan evolved continuously throughout the process, it was evident when attempting to work through some of these tasks that even as a release outline this lacked detail. For example, ‘task creation’ involved not only the display and database queries but also a significant amount of research in how to respond to live changes in data. This shortcoming lead to a difficulty in ability to estimate task length that was carried over, into the first development sprint.

List the features for each development

This outline was developed into a more descriptive plan for proper sprint-based development. I further developed the plan into (\*\*fig\*\*) with more tasks still grouped into releases. It was only at the planning stage for each sprint that I decided which tasks should be completed over the two-week period. Each sprint has unallocated contingency to allow me to adapt to development changes or to recover for tasks taking longer than estimated.

The process of allocating tasks to sprints at the beginning of each period at first was farely labourious. Time spent planning and arranging tasks could have been used to develop. That being said it proved to be extremely effective as the process went on. I could take lessons learned from previous sprints and make adaptations to what tasks I was completing on a fornitely basis. This proved invaluble when additional requirements where brought up, either by research or by discussion with my supervisor. Eg GDPR, QR, Scanning, iPhone bug, task requirements, new user bug when testing questionair

This is an example of a sprint plan after the sprint has been completed. The development tasks here where allocated from a list of tasks adapted from the initial plan. Tasks that carried high risk or ones that I had a low confidence in the accuracy of the estimate where prioritised. This gave me capacity to correct any errors made and extend development time of features where required.



Documenting and following through with the plan is the main process that I attribute to the success of this project. Being able to anticipate the workload of sprints allowed me to manage my time properly with both this and other assignments, maintaining a consistent work output. Separation of sprint planning and development became an effective way to prioritise tasks and maintain focus of development of certain features.

In future projects, especially ones of this size I will use a lot of these processes. Saying that this process was not without flaws. I did not write detailed stories about each task/feature, this would improve the process by having more detail about tasks to refer to later and add a success criterion for each individual task. The reason I didn’t follow this specification of agile methodology for this project was the additional time requirement in the planning stage of each sprint all though on reflection this would have helped add focus to development. Following this doctrine would be especially relevant when working in a team to clarify intention and reinforce requirements.

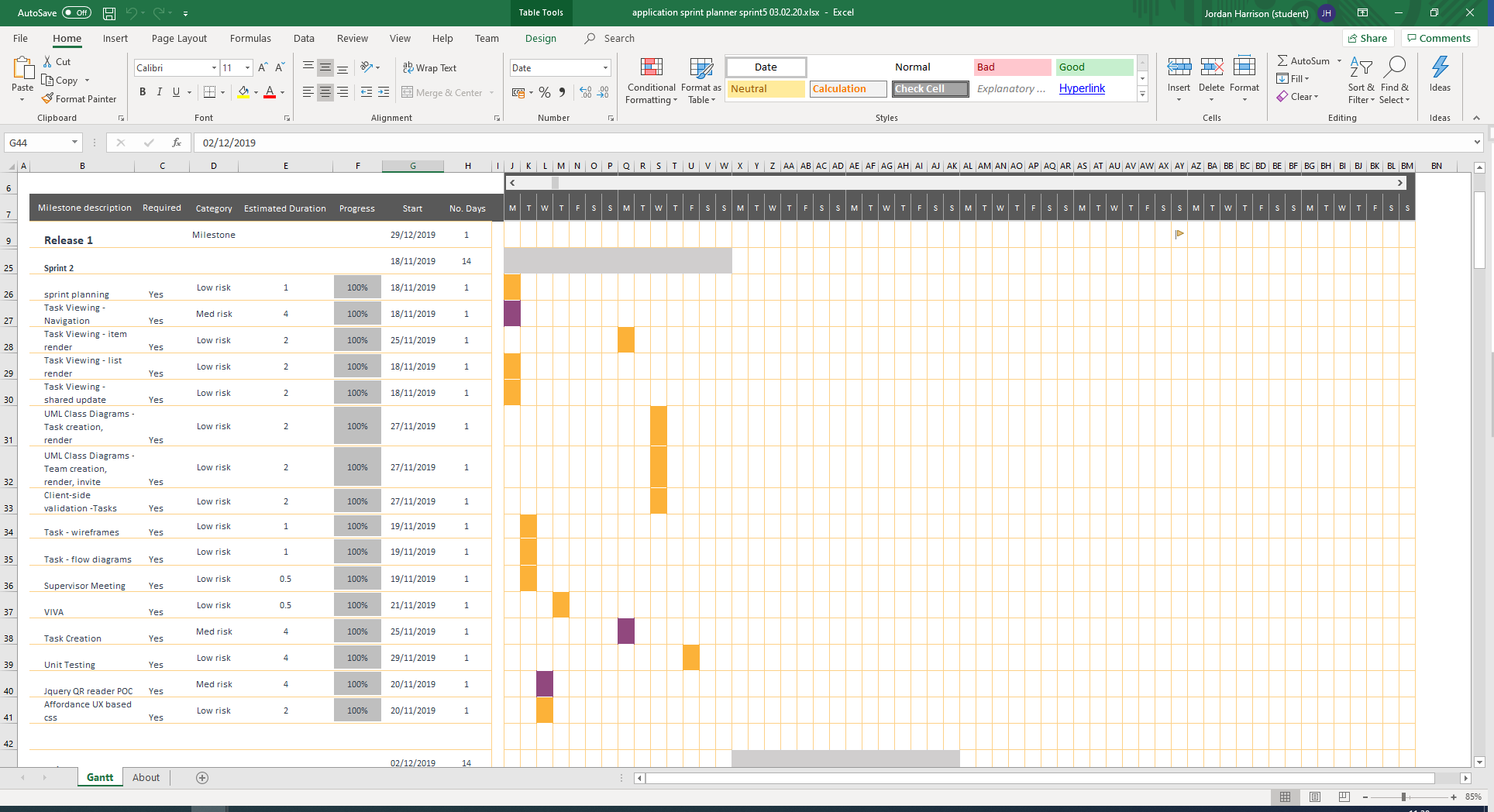
I initially struggled to time tasks accurately the burndown chart completed for this sprint presented a worrying overrun in task length. This is something that fortunately I had the spare time to handle for this sprint as I had no other assignments out and so could solely work on this project. I had to make sure that this didn’t happen again: the contingency allowance for each release did not allow for this level of over-run.

This highlighted a fundamental flaw I was making with my planning process. Two of the tasks during this sprint (Team Modification, Team Creation) over ran substantially. The main error was that these two tasks consisted of both fulfilling the requirements of being able to create and modify teams and architectural systems that enable these processes. As part of these tasks I needed to start creating a generic way of editing and creating document objects in specific collections. I made a significant modification to the way I was planning from this point on, breaking down tasks into more atomic ones.

\*\*\* doesn’t include any writing of the project proposal

In the second sprint I took observations of issues with the process into practice. There are several tasks within this sprint that where initially part of one bigger task. UML documentation of application process was initially one task this was broken down and spread among multiple sprints. Task viewing was split down into its constituent parts, improving accuracy of estimated length.

Ali reinforcing decision to go ahead with sequence diagrams

Week 2 burndown

and after the second sprint I had a substantial over run of time taken beyond my estimates. I had to adapt my planning process by splitting up tasks into smaller, more atomic tasks this proved to be very effective and the more planning I did the more accurate my estimates became.

Working document spreadsheets

Burndown charts

Risk tool add burnout talkabout downtime

Starting early has allowed me to see problems and deal with them,

Week break - burnout

### Risk Analysis

An important part of managing a project, particularly one of this size, is understanding, anticipating and managing risks involved in development. A risk analysis tool can be used to document and analyse risks involved. The following is a table created at the beginning of the project and was adapted following feedback after the viva meeting to include a Likert scale to represent likelihood and impact where 5 is the most likely to occur and highest impact.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Effect** | **Protection** | **Likelihood** | **Impact** |
| Low performance of client-side application on low spec hardware. | Potential UI lag and browser crashes. | Choice of technology needs to have the capacity to control updates. | 2 | 5 |
| Time taken to develop sections of the application maybe more than estimated. | I could run out of time to properly develop and test each component. | Using Agile and OOD to break down production into sections. Enables reallocation of time on individual tasks. I have also allocated a month-long contingency period. | 1 | 3 |
| Reaching the usage cap of a cloud hosted service. | May incur hosting costs to allow application to proceed. | Pay careful attention to database metrics and move testing to a different billing/usage period when appropriate. | 3 | 5 |
| Data-engine unable to fulfil application requirements. | Functional requirements are either impossible or time consuming to produce. Potential lacking in implementation quality. | Extensive research and proof of concepts as part of the design and planning for the initial release. | 1 | 5 |
| Changes to framework/API methods or functionality. | Any changes to the way API methods are called will impact development time and could remove functionality all together. | Choosing only well established and well documented external APIs. Only upgrading client-side packages if required. | 3 | 3 |
| Changes to IDE and development environment applications. | Potential loss of code through file format changes. Applications either don’t work altogether or require relearning. | Use commonly used IDEs and application. Pay careful attention to release notes before updating. | 3 | 2 |
| Database downtime | Loss of ability to test/develop. Downtime at the point of marking/examination. | Any service managed by me will be carefully monitored and include redundancy. Any external services must have a proven track record. Segmentation of project to allow for non-database reliant work. | 4 | 2 |
| Data loss/corruption | Inaccessibility of current dataset. Loss of modification to hierarchy/schema. | Maintain backups of dataset and testing of restoration. Backup of any substantial hierarchy/schema changes. | 1 | 1 |
| External service outage/deprecation | Inaccessibility or complete loss of data. Time impact on configuration of a new service. | Maintain backups of datasets. Only use tried and tested external services. Selected services need to have high availability. Maintain access to website code so hosting can be moved. | 2 | 5 |

A risk documented here that was present during the development of the application was the database downtime / external service outage. During development there was a small amount of downtime on the firebase servers managing the command line interface (CLI) authentication. This had an impact that I did not anticipate, I had assumed that any downtime of Firebase servers would only apply to live hosting services this turned out to have local testing implications. During this downtime I was unable to start a node.js based local hosting service I was using to test development changes. The assumption I made that local changes would be possible made the protections in place less affective. Fortunately, this downtime was only brief, and I was able to identify the issue via communication with the Firebase support team \*\*\*.

A problem that came about due to mismanagement of task length and general workload of this year was burnout. I did not anticipate this risk but was able to provide a way of remedying. This came about because of bad estimate of task length along with not setting aside time create a project proposal this resulted in an excess of workload which has impact on my mental health. I managed to reduce the impact of this by setting aside a week where I didn’t do any project work. This improved the amount of output during the weeks following this period.

Firebase CLI cloud hosting

Firebase CLI cloud hosting

### Surveys

Link affordance with descernability\*\*

Surveys where conducted in order to evaluate the User Experience(UE) of using Workaholic, to evaluate the success of base usability of initial development and investigate opinions on potential additional features. The study was conducted on a very small sample of 5 people. Each of the users was given the same amount information on the things they needed to do, and the application version was maintained throughout.

The survey consisted of atomic tasks that are designed to be understandable from the moment of opening the application. Actions where explained in terms of what the participant needed to do but not how they go about doing it. This was an important in order to give a way to assess the affordance of the UI.

Tasks where timed from the point of giving the instruction to being observed to completing it. Times taken where rounded to the nearest whole second with tasks under a second being rounded to 1. This creates an empirical basis to assess a participant’s understanding of how the application works based on affordance.

|  |  |
| --- | --- |
| questions |  |
|  |  |
|  |  |
|  |  |

The following graph shows (fig\*\*) the time taken for each survey participant on each question.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | S1 | S2 | S3 | S4 | S5 |
| mean | 59.75 | 47.25 | 45.75 | 56.25 | 144.75 |

A surprising observation made Is that every user was able to transition from the team’s page to the tasks page without being told how to do so. I was expecting to see a significant difference in the time taken between completing task 1a and 2a for all participants. This was demonstrated by S5 which I know personally and is particularly unfamiliar with mobile based applications.

This demonstrates that a user can identify how to do each task from by what is being displayed on the screen. Which is a sign that basic functional requirements have been met and that the application affordance is correct.

The next set of results are grouped until similar assessment. In all the following results the users where asked to attempt to complete a task. \*\*\* consider un re-rounding these

This set of results demonstrates that a user can complete tasks completely. This is a very important concern with the uptake of the application. From the prospective of a task doer at this point of using the application any labour required is already understood and completed. This means any time spent completing the task on Workaholic is a potential productivity loss and therefore should be as quick as possible.

\*\*\* additional features, other user notifiication

The following results are the participants were asked to measure their level of understanding on the effect their actions had. The purpose of these survey questions where to evaluate how discernible actions where. There is a fundamental requirement for actions to be discernible both in terms of using an application and for gamification elements to be effective\*\*\*.

Most of the people surveyed where people I knew personally so there are potential biases arising due to the personal relationship. The biases may result in a skew towards a positive result.

A major shortcoming of this survey is the number of participants. This means any conclusions made are tentative. A bigger sample size would allow for any potential statistical anomalies to be picked out. Although several S5 results are well above the average of the survey the lack of results means the results cannot justifiably identified as anomalous.

Further surveys should include more participants but also include alternate process for comparison. This could be either an application that has similar functional capabilities or a more traditional form of personnel management.

Although there is evidence of discernibility measuring the effectiveness of gamification elements was difficult for several reasons: gamification and progress within the application would require long term use, gamification is dependent on using the systemic context \*\* explain elsewhere \*\*.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Feature** | **Order potential features in terms of what you most want to see** | | | | | | |
|  | S1 | S2 | S3 | S4 | S5 |  | Avr |
| i: Tasks contingent on other tasks | 8 | 3 | 5 | 1 | 3 |  | 20 |
| ii: Other user task competition notifications | 3 | 4 | 3 | 2 | 2 |  | 14 |
| iii: Email/Text based team invitation | 7 | 1 | 6 | 7 | 7 |  | 28 |
| iv: Additional login options | 6 | 6 | 2 | 8 | 6 |  | 28 |
| v: Inbuilt calendar | 5 | 7 | 1 | 4 | 8 |  | 25 |
| vi: Team types Restricted edit | 2 | 5 | 4 | 5 | 5 |  | 21 |
| vii: Performance Graphs | 4 | 8 | 8 | 6 | 1 |  | 27 |
| viii: Configurable Skill icons | 1 | 2 | 7 | 3 | 4 |  | 17 |

This table shows how participants responded to ordering features in terms of what they most want to see with 1 being most wanted and 8 being least wanted. With the lowest average ii was shown to be the most desired feature. This was previously considered for removal from the planned features. Opinions given here helped decide between the positives of promoting a sense of community and the negatives of potential competitive elements.

### Test Driven Development

Test Driven Development (TDD) is the process of writing program changes in order meet the conditions of tests. These tests are written before development occurs and have pass and fail conditions that are known results. This is an effective way of mitigating risk from making changes to complex systems.

Research \*\*where\*\* and initial testing showed that firebase rules had poor default error handling client-side. Actions can be prevented but the nature of the permission error is hard to identify. When querying a collection if the query *could* return a record that a user doesn’t have permission to access, it will fail regardless of the actual results of the query. This added a requirement to be able to identify the effects rules had on a user’s ability to access documents and collections.\*\*

The following test where written and performed from a separate page built for purpose. A test account was used and no database changes where made during the process.

|  |  |
| --- | --- |
| Test Query | Expected Result |
| document in the accounts collection that does **not** have the same ID as the authentication token UUID. | Reject |
| document in the accounts collection that has the same ID as the authentication token UUID. | Accept |
| Entire Collection of teams | Reject |
| Collection of teams where UUID is in the members collection | Accept |
| Task of a team where UUID is **not** in the members collection | Reject |
| Task of a team where UUID is in the members-collection | Accept |

After writing these test rules where created using the firebase console. This proved to be a very effective way to develop these. Although writing the initial tests was time consuming this time was saved by speeding up development when writing the rules. Having tests to run at hand rather then having to use the application to test rule changes had large productivity benefits. These tests also gave confidence that the rules I have written had restricted access to any personal data.

It was important to not add rules until I had the capacity to test as any changes to the database structure or document values could affect access. This would not only waste time disabling and rewriting rules but have development issues where error messages are misinterpreted hindering debugging coding errors.

This carried a small amount of risk where access to my database was relatively unrestricted during development, due to the restorative nature of the application this was somewhat mitigated.

## Product Description

What has been created, where relevant, explanation of requirements, design, implementation and evaluation of choices and outcomes

3000

https://developers.google.com/web/progressive-web-apps

I have created a work management application called Workaholic. The purpose of this application is to add intrinsic value to doing tasks in the form of gaining experience points. The intension of experience gained in the application is to correspond to gains made in real life. This allows teams of people to create an ecosystem which motivates people to not only do tasks but to actively engage in their own professional development.

Users earn experience points (EXP) by completing tasks. I initially considered EXP as superfluous to the purpose of the application, after researching I realised this was necessary to add enough complexity to allow user integration in the game system while adding feedback (ZHANG, 2008) and “persuasive elements”(OPRESCU; JONES; KATSIKITIS, 2014).

Initial research showed successful engagement with digital gamified systems may also involve following implementation of motivational affordance (Zhang, 2008). Some of the motivation sources more relevant to a system such as this and principles proposed by Oprescu, Jones and Katsikitis (Oprescu, Jones and Katsikitis, 2014) have highlighted several common properties that have been observed. A significant amount of these draw parallels with the Octalysis framework proposed by Chou (Chou, 2016). \*\*\*\*

The mobile context requires for actions to be quick and require minimal attention \*\*\*. This something that is a potential weakness of Workaholic when creating tasks users took between 34 and 263 seconds to add a task. This maybe problematic when a user needs to add a large volume of tasks. This speed concern maybe slightly alleviated by signs that the speeds improve with use.

When a user is for fulling the role of adding tasks, they are either only adding a single task or their duty has shifted to a more organisational role. I see this adding importance to the capacity to use the application in a different context. PC users are known \*\* to be able to input text much faster this increase in productivity suggests that someone for fulling an organisational role is better suited to a pc context.

Vue.js

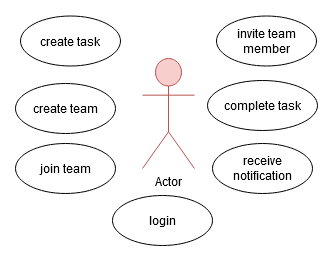
### Design

#### User Stories

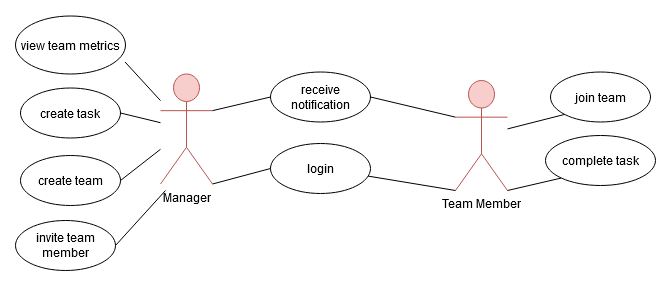
#### Use Case Diagrams

\*\*user stories

Initial plans to decide on functionality requirements made use of simple use case diagrams. My first diagram fig\*\* has only a single type of actor, of which all users where one of. This means that that user will be able to add and edit tasks of any team the user is part of as well as edit the actual team. This makes sense in teams where it’s conducive for any user can fulfil both organisational and task doing roles. This type of actor would function perfectly in the context of story 1 as the team has no single person organising them. But less so in the professional context of 2. This system also has the potential for abuse with team members being able to modify task data, such as experience reward, awarding themselves more for an easier task.



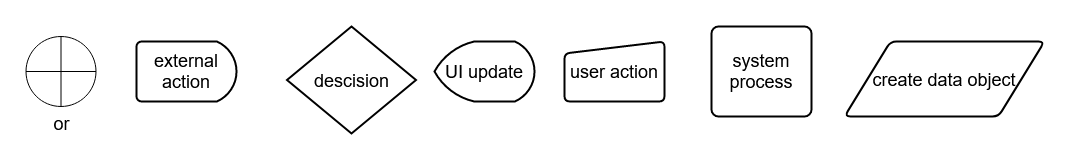
During a meeting with my supervisor a concern was brought up about how widely this work pattern would be accepted. In the case of a more structured team it is often the case that organisational and task doing roles are completed by different people. This led to the development of a second use case.



This diagram was created after the discussion and represents the actions available for a more structured team. The manager is the only user able to create and edit tasks, they are also the only person that has control over the properties of the team. Ideally the application would be able to support both use cases.

#### Flow Charts

As part of initial designs, I use flow charts to enhance my understanding of the processes the user needs to go through and the actions the system my take as part of that process. It is important to have a good idea of how these processes work as this will enhance screen design.



##### Login

The account record as part of the login process (Fig\*\*) was adapted after initial rule testing showed checks could be done against array elements. An empty skill level collection is now created on a new account object to prevent an error when attempting adding values to this.

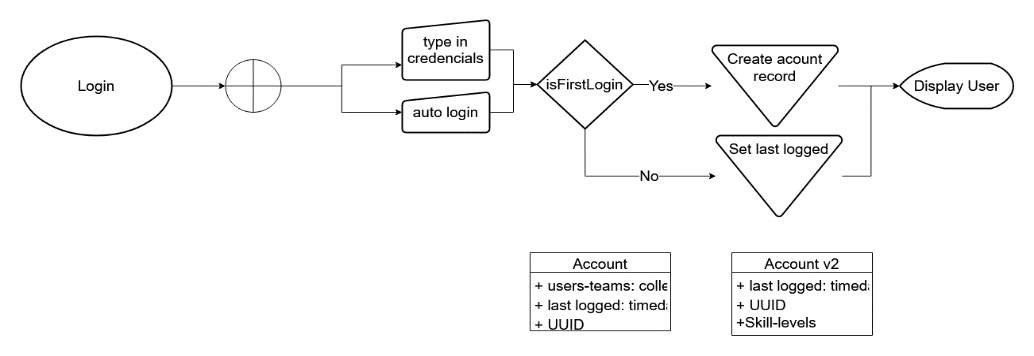
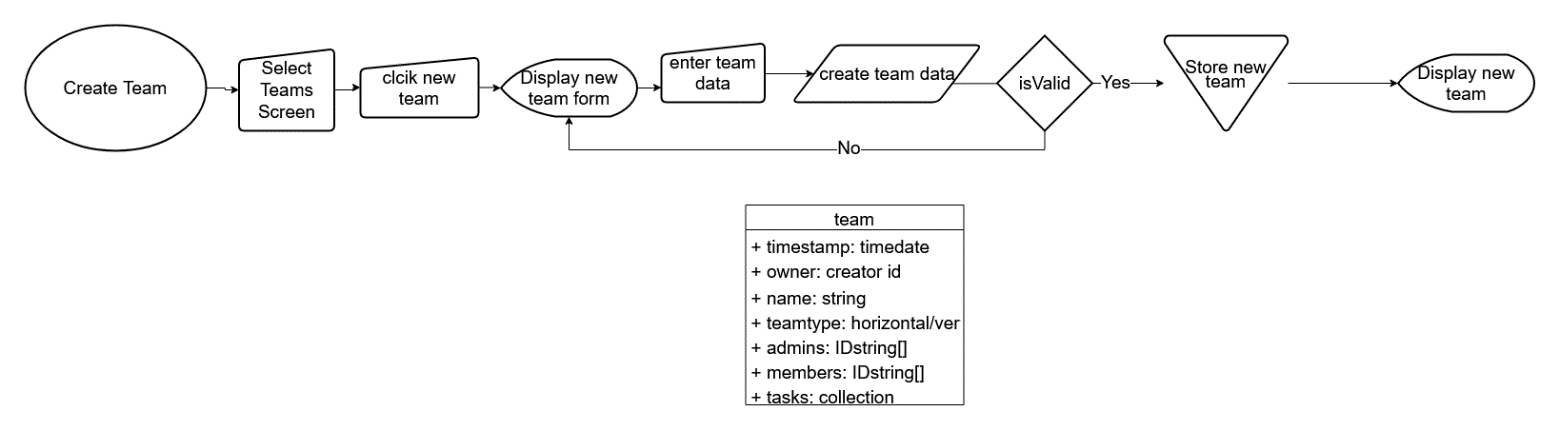
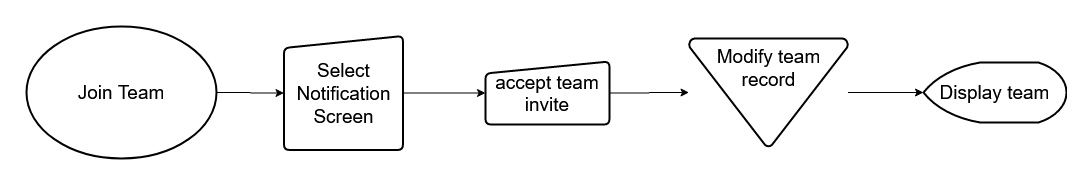
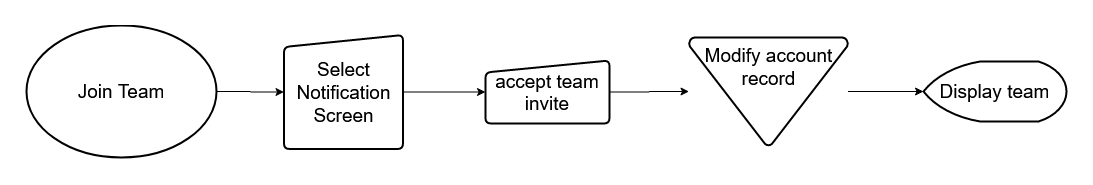


Fig \* : Showing changes made to the account object creation as part of the login process.

##### Create Team

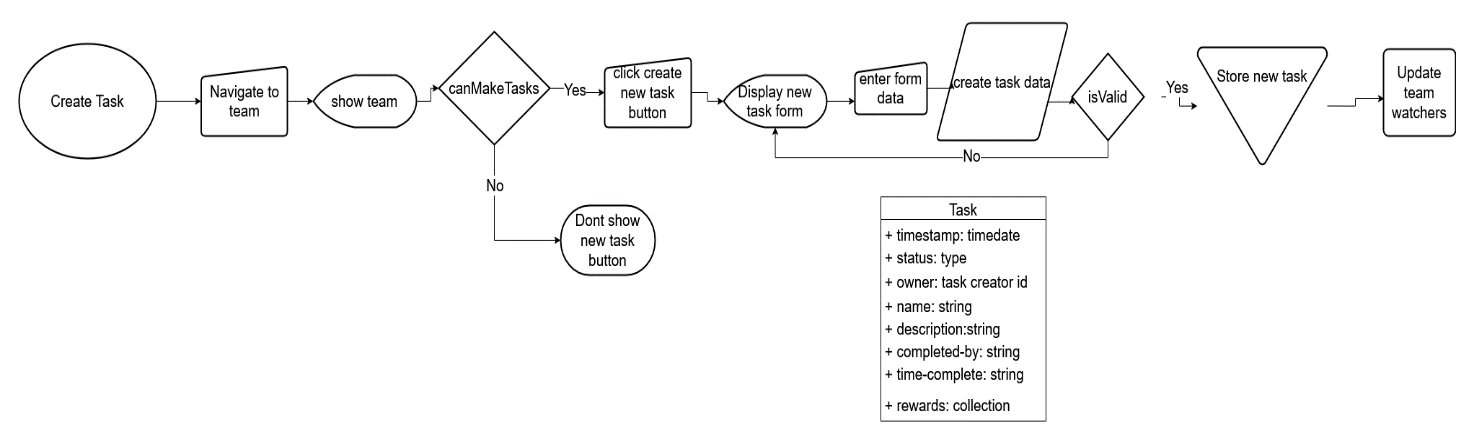


##### Join Team



##### Create Task

In fi\*\* the process highlighted is being enforced by input attributes allow the browser to handle input validation. Improvements to this process could be made by adding object validation after user input and server-side data validation before committing to the database.

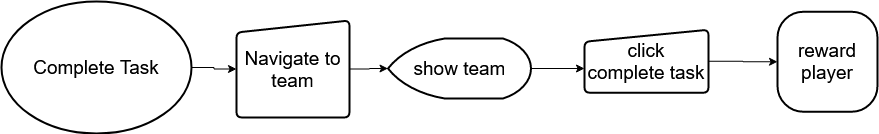
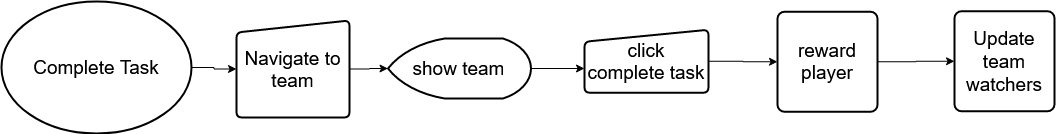


Fig\*\*: User and system actions in the process of creating a task

##### Complete task

Fig\*\* shows changes made to the complete task process with the hope of adding additional sense of community. I was initially sceptical as this change could also be seen to be adding competition elements. Survey results indicated that this feature would be well received, scoring the highest of all listed potential features.

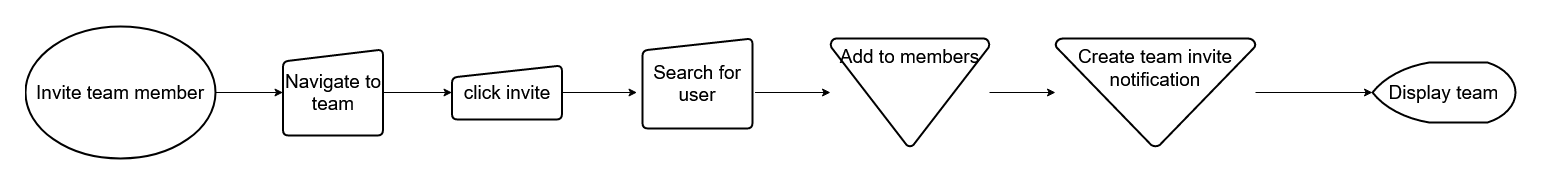
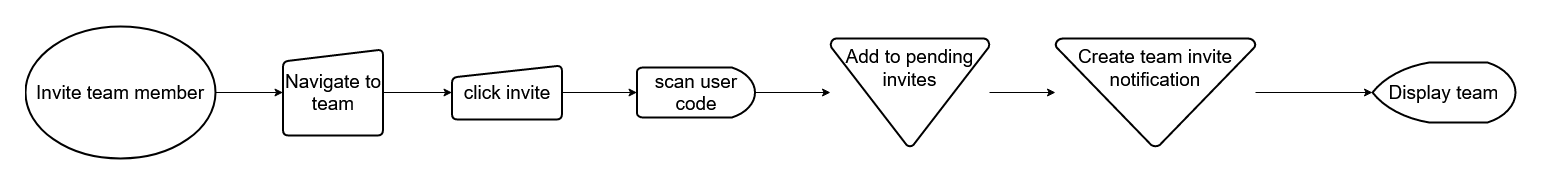
Fig\*\*: Potential user and system actions in the process of completing a task. Version one (top) version two (bottom)



##### Invite Team Member

The process of inviting a team member received the most change this is due to GDPR requirements. Initial designs of this process involved allowing a user to search for other users. There are several concerns with this process the first being GDPR related. A discussion with my project supervisor raised the concern that in order to search for other users there needs to be a collection of names which has concerns with storage and information accessibility. In order to enable searching of user data particularly where there are duplicate names additional data is then needed to be both stored and accessible in order to differentiate between.

In order to circumvent these issues, the invite process uses document codes to create invites, which a user then accepts in order to join a team. Document codes are passed over via the use of QR code scanning. This system allows users to access team documents without unnecessarily exposing team or user data to the UI. Adding a member to pending invites gives them access to the team.



Fig\*\*: Showing changes made to the invite process to improve GDPR compliance

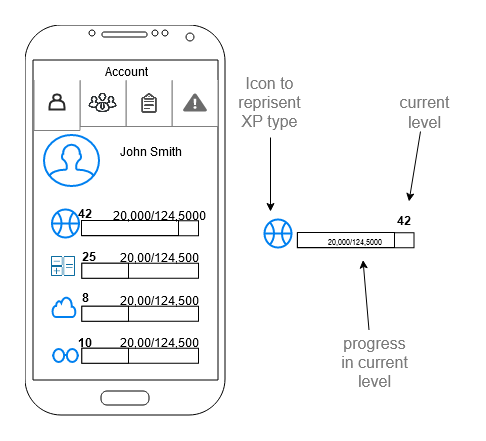
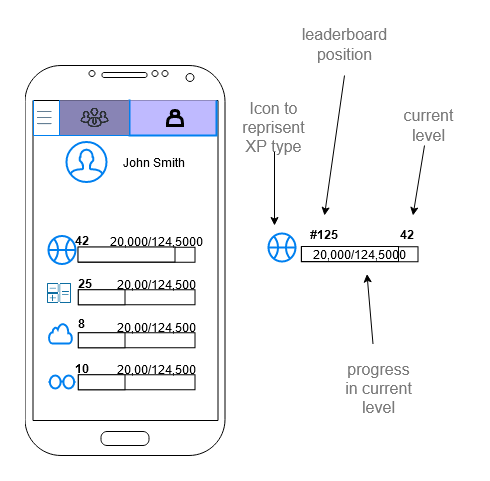
#### Wireframes

Following the creation of flow diagrams where wireframes, these where informed by the flow diagrams. Several significant changes were made as a result of further research. Additional pages have been added between the designs. Firstly, the notification page was created to facilitate community via feedback of team actions.\*\*requirements\*\* A separate task page was added to allow users to quickly access the tasks list they were previously viewing.

##### User Screens

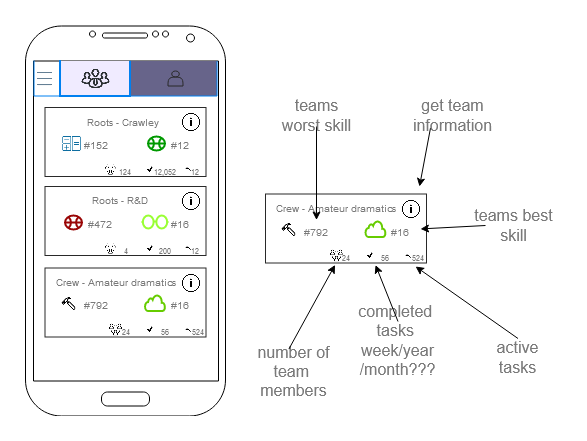
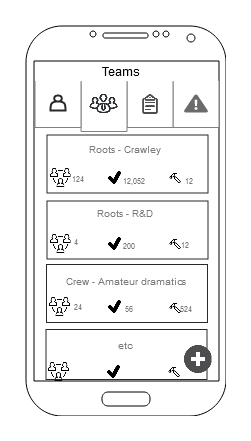
My preliminary research highlighted potential issues with leader boards; this was initially a component of the design. Further research showed that comparison with peers can cause anxiety in individuals (ZHANG, 2008), although this could have the desired motivational effect I consider it the wrong route to take in progression of gamification in general. It could be considered a “black-hat” motivational factor and may backfire reducing motivation in the long run (CHOU, 2016). It is because of this that the design does not feature any form of leader boards. This design change can be seen in design changes between versions of the user screen (fig\*\*)

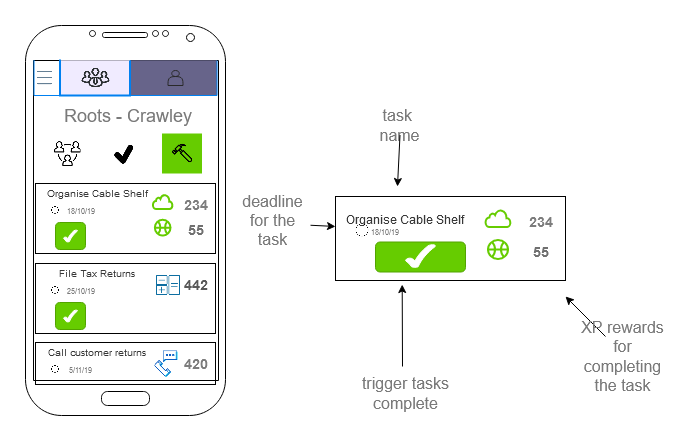
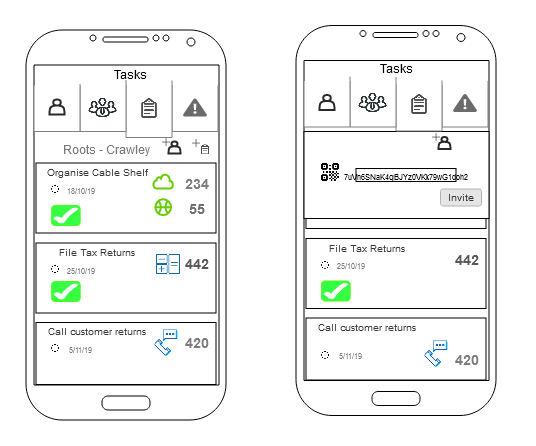
Fig \*\*: The user page version 1 (left) and version 2 (right). Additional pages added to the navigation bar.



##### Teams Screen

Competative





### Requirements

One of the major requirements of the application is a persistent data source that all users have access to either a SQL or noSQL(document) database. This data needed to be secure, only being accessible to members of a specific team. Appropriate measures needed to be taken to conform to 2016 GDPR regulations.

Infrastructure and hosting requirements of applications evolving persistent data are normally more complex. Webservers need secure high availability access to database servers. This is often done through virtualisation where both servers are situated on the same machine and often involve clustered networks where one or multiple web servers have access to one or multiple database servers.

Due to my limited access to hardware my options where: create a Linux web and database server on a raspberry Pi hosted via my home network, create a virtualised setup using a provider such as AWS or Brighton domains or using an external all in one service such as Firebase. Hosting via my home network has both security and availability concerns on top issues with reliability and capability of available hardware. Brighton domains are capable of running both types of services to gather, process and display data these services can also be integrated with Firebase the fundamental issue with this is one of workflow were node servers must be restarted to view any changes. The hosting provided by Firebase operates on a cloud network giving access to replicated hosting configurations in multiple locations improving availability whilst handling all communication between web and database servers without any setup.

During my placement and whilst at university I have had exposure both databases technology that meet the storage requirements and had the opportunity to developed multiple APIs in a variety of languages to access this data. A significantly capable data-engine would take a lot of time to write and as a result would make up either the whole or at least a significant part of development.

### Implementation

#### Firebase

My initial investigations turned up several alternatives to writing a data engine. \*\*\*\*

Some of the features that attracted me to firebase client notification of data changes, the ability to secure documents based on the values contained there, cloud hosting features, f

OAuth OKTa

Significant operational cost to function based process – opted to create clientside processes to interact with data

In order to keep data accessible only to team members some form of authentication process is required. This will uniquely identify the user and thus the tasks and team data they have the security privileges to view.

The login process of Workaholic utilities Googles authentication service in order to login to the application. This utilises an OAuth2 based \*\* token system to validation transactions with a web server and therefore access to the database. This process could use a self-written token authority or by utilising Firebases login services. The primary advantage of using google authentication is in the user experience, if a user has logged into another google based service there will be an authentication token on the device already. This improves the user experience by removing the necessity for a user to create separate accounts requiring them to remember additional login credentials.

##### Collection restoration

An effective form of risk mitigation that I did not anticipate or discover during my initial research was the ability for firebase to restore collections. When writing database queries a query that adds a document to a collection that doesn’t already exist the data-engine will create that collection before adding the document. This has the advantage that database structure can be restore by users performing application actions. This has series positive operational implications: databases can be moved, created and deleted with minimal setup and without having to prescribe a schema.

In a traditional database (SQL) schema is written before any data is committed to tables. This has the advantage of being able to strictly enforce format and validate data input resulting from queries. Firebase is described as schema-less this means any data can be put in any object and documents within the same collection can be entirely different.

ES6

Custom-elements

UML \*\*\*\* user roles task doer() task creator() both team types link to survey result seperateing group 1 task does group 2 task creator

A number of design changes came about due to conversations with my supervisor. Firstly the capacity to cater for organisations that have specific members of the team allocating tasks and secondly potential issue with GDPR prevented the ability for users to search by name. This made invites by QR reading more important and changed the plan as such.

This upgrade was from a conversation with my project supervisor

\*\*UI issues identified before the survey\*\*

### Custom Elements

Custom elements are used extensively throughout this project. These allow developers to define their own HTML elements. Method used to register elements to the DOM is enabled by four specifications ‘The Custom Elements specification’, ’The HTML Template specification’, ‘The ES Module specification’ ISSo\*\*\*. Those particularly relevant to this project are template and custom element specifications.

//https://www.webcomponents.org/specs#the-custom-elements-specification

<https://github.com/w3c/webcomponents/issues/509>

<https://www.chromestatus.com/feature/4670146924773376>

<https://github.com/w3c/webcomponents/issues/509#issuecomment-230700060>

https://en.wikipedia.org/wiki/Liskov\_substitution\_principle

Elements can be used in two main ways: autonomous elements can be created via extension of the base HTMLElement class and any native implementation of this class can be extended to create a customized built in element. Both methods are widely supported by most modern browsers an exception to this is the lack of support of customized built in elements in Safari. Support of this can be done through implementation of polyfills like \*\*v1 custom elements\*\*.

Customizing built-in custom elements is a very powerful way of modifying and extending functionality. Initially the edit button was being created via extension of the native button element. This allowed additional functionality whilst maintaining normal button semantics for styling purposes. There was an issue with this way of defining elements that primary research revealed.

Several participants of the Workaholic usability survey where using iPhones to use the application. This meant they were utilising a version of Safari to view the webpage. This quickly showed an issue where users where unable to trigger display of the edit form for any documents.

Further research into the browser support of customized elements showed a reason for this could be the lack of support of customizing built in elements. After confirming this bug was also present on mac versions of safari this element was changed to an autonomous element instead of opting to polyfill this functionality. Safari is opposed to the standardisation of customized base elements due to the potential reduction in type interoperability, of more complex implementations of base elements which would violate the Liskov substitution principle \*\*,. This justifies the losses in semantic clarity from switching to an autonomous custom element.

VUE.JS

Displaying HTML in this way relies heavily on JS functionality of the browser displaying the page. Any errors that happen during this process could result in changes to the DOM that cannot be rendered

### Document List

An architectural was made relatively early decision regarding how the documents are displayed to users. I quickly realised that there will be functionality similarity with how any collection of documents are displayed. In most cases the application needs to respond to any modifications to query results including the data in the documents themselves.

This functionality is relatively easy to implement using firebase as the API exposes changes to query results. This enabled the creation of some generic class to support display of these changes. The ‘Static Query List’ (static list) simply uses the result of ‘getQueryReference’ to create and append to the DOM a set of cards. The creation of HTML elements used to display individual document data is done through ‘createCardDOMElement’. The extension of this class ‘ActiveQueryListElement’ (active list) modifies how query references are handled adding methods that respond to ‘added’, ‘removed’ and ‘update’ changes to the query result. Finally, active list is extended into ‘ChangeableActiveQueryList’ (changeable list) to enable changes to 'collection-target' to modify the collection target of the list after it has been added to the DOM.

Fully derived versions of active list are used to display the team and notification lists. Team and notification data are situated at the top-level collections. Therefore, the query that is used to access this data doesn’t change but the contents of the query and values of documents within does. The requirement of these lists inspired the creation of the active list base class where there is a significant similar functionality that can be compartmentalised to a less derived class. These serve as examples of how active lists can be used, derived class can be used to access and display any collection of documents. Displaying any updates to individual documents through modification of HTML attributes.

The tasks page presented a different requirement for data access. This page is used to display documents that are stored in different collections. Task data is stored in a collection underneath the team it belongs to. When a user views team task data this list changes performing a different query against the task collection depending on the team viewed. This inspired the creation of the changeable list class, of which the tasks page is the only fully derived version. It was important to have a generic way to handle this functionality, despite there only being one example of its use, as changes to document structure may bring additional requirements for this.

## Critical Review

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

No rule based validation

A major short-coming of this project is the lack of server-side data validation. User input is somewhat guarded using form-based input validation. But after the query is made there is no server-side guard to prevent invalid data. This, under normal use, should not be an issue because of the way queries work. After objects are created to query the database an error in this process will lead to no database changes rather than erroneous data. There are two main ways this could cause issues: a user entering data in non-normal form elements cannot be validated in the same way and malicious use of the application can produce objects that contain data that isn’t intended. The former has been compensated for in the generation map data an example of this can be seen in personalised skills of a team. When this input forms an object to commit to the database the uniqueness of map keys prevents duplicate data input.

\*\*how

No rule based task edit restriction

In order to allow for additional use cases of the application a team owner can restrict editing of team data. The is done through modification of UI controls to prevent user accessing team edit forms. This has a fundamental issue with how the restriction is made, a user could modify HTML elements to give them access to team edit forms. This process could also be restricted by database rules, correctly written rules \*\*\* would be able strictly enforce restrictions to a user’s ability to edit team data that they are not an owner of.

\*\* how

Throughout the development process I have shown good discipline following development and reflection processes. Creating a list of development tasks required to fulfil functionality requirements has enhanced focus and created success criteria for each individual task. This is something that I attribute to the successfulness of development. This is something I have never done before and have learned lessons in task description and allocation that is going to be invaluable to any future project.

Firebase has been a pivotal component of this project. Use of a data engine has allowed me to focus on the functionality of the application and the enablement of gamification. Learning to use this API was essential for development. Firebase rules are an effective way of validation and securing data learning to use these was essential to improve the security of the application. Appling listeners to documents/collections as a way of monitoring and displaying document changes is something that, I learned during developing this and I have already brought into other projects.

Custom elements became an effective way to display document data. This is something that was a struggle to start using as it required me to both learn ES6 object notation and the specific methods associated with HTML element-based prototypes. A major lesson I learned about custom elements is the level of support between browsers. An example of this difference is seen in safaris interpretation of extension of default elements, this by default is not supported by safari because \*\*\*\*.

Regular meetings with my supervisor and reflection about my process have aloud for continuous evaluation and adjustment/refinement. I have treated my supervisor and myself as stakeholders in the project as replacement for actual customers.

\*\*\* external node server to calculate experience gains

3000

## References

1. A full accurate list of references to all sources of information that you have used including the source of any non-original material such as code and media assets. You must also reference any tutorials or other sources of information that informed your project.

## Appendices

Chou, Y.-K. (2016) *Actionable gamification: beyond points, badges, and leaderboards.* Fremont, CA: Octalysis Media.

Oprescu, F., Jones, C. and Katsikitis, M. (2014) 'I PLAY AT WORK-ten principles for transforming work processes through gamification', *Frontiers in psychology,* 5, pp. 14-14.

Serrador, P. and Pinto, J. K. (2015) 'Does Agile work? — A quantitative analysis of agile project success', *International Journal of Project Management,* 33(5), pp. 1040-1051.

Zhang, P. (2008) 'Motivational affordances: reasons for ICT design and use', *Communications of the ACM,* 51(11), pp. 145.