Task Keeper

Ben Teft (N0865531)
Ethan Wilde (N0864419)
Hannah Ashna (N0865554)
Simon Fincham (N0864929)
Owen Rennie (N0864494)

Group: 14B

Abstract:

For this project, the team decided to conduct research into and investigate whether gamification can reduce procrastination. From the research conducted, the team found that there were a variety of ways to reduce procrastination but the most prominent is through the theory of risk and reward. The team opted for a solution whereby the user was given the responsibility of taking care of their virtual pet. The user can complete tasks to earn currency which can be traded for toys, food or water depending on what the user's pet requires. The team imagined that this gamified component, among other features, would entice the user into logging in to the app daily. The team aimed to use sources derived from credited sources across the globe to ensure the scientific theory behind the team's idea was legitimate.

The project the team has been working on for the past few months is called "Task Keeper". It is a gamified task-tracking mobile application that enables the user to create to-do lists. The process of how this component works can be found in the heading: Flowchart 2: To-do List Functionality. The team believes that Task Keeper acts as a simple yet effective alternative to mainstream competing products already on the market, such as Habitica (Habitica, 2021), Habit Hunter (Active User Co., LTD, n.d.) and Epic Win (Rexbox, n.d.) that collectively, have over 1.1 million global downloads on the Google Play Store.

Aside from the team's use of a detailed test plan, to test the viability of the product in a real-world context, the team's System Manager provided an Android Application Package (APK) version of the app to some university peers. Following this distribution of the APK, the team received feedback via a survey (Google Forms, 2021)(Appendix H).

The team concluded from the survey's feedback and outcomes of the testing phase, that the Task Keeper application fulfils all its objectives as it encompasses the project's aim of being a gamified task tracker used to combat procrastination. Moreover, the team also found that the application is suitable for distribution, predominantly due to its "ease of use and user-friendly UI".

Table of Contents:

Table of Figures:

Section 1- Introduction:

Procrastination is a battle most are familiar with. A survey conducted by (MacNaught, 2019) showed that '84% of us are affected to some degree by procrastination'. Procrastination is also a point that is frequently brought up in conversations discussing the impact the COVID-19 pandemic has had on productivity. For instance, the Q3 report from the (Bank of England, 2020) estimated that "COVID-19 would reduce TFP (total factor productivity) in the private sector by up to 5% in Q4 of 2020". Undoubtedly, the COVID-19 pandemic has had a massive effect on every component of normal, day-to-day human life and has forced many people to spend prolonged periods online. According to (CIPD, 2020) "maintaining a healthy work-life balance has become increasingly challenging, with 3 in 10 finding it hard to fulfil commitments outside of work due to time spent on their job".

Although procrastination is such a widespread issue, no blanket solution has been proven to work for everybody. However, collectively, the team believes that gamification could be one solution to this issue. A study published by the International Journal of Educational Technology in Higher Education using a gamified application showed that "app users on average achieved marks 7.03% higher compared to students who chose not to use the app" (Pechenkina, 2017).

Why gamification? A study (Kuo-chen Li, 2018) found that certain elements of games can be used to combat procrastination factors. Some key examples being task aversiveness and the use of rewards and punishment. The study notes that the use of "goals in the game" can shift "unpleasant, boring or uninteresting" tasks to being "more interesting and motivating goals". Hence, subtly pushing a user towards the pursuit of successful task completion. Moreover, the study notes that the use of "virtual rewards and role formation" results in a "significant improvement of performance" in an individual. Thus, the use of gamification in a to-do list centric app will help improve productivity by making the process of task completion more engaging. Furthermore, (Bhanji, 2013) argue that "out of a vast space of possible actions, the prospect of a reward helps us select those actions that will lead to the most and best rewards and motivates us to carry out those actions".

Also, upon surveying some peers at university (Appendix A) the team found that 80% of them believed a gamified version of a task-tracking application would help them work through procrastination. Through the survey, the team also highlighted potential game elements that could be implemented into the application and after an analysis of participant feedback, the team found that features such as points/rank systems and character customisation were highly requested due to their ability to create an engaging and competitive game. Thus, linking back to psychological research that emphasises the importance of motivating goals and rewards to increase performance.

Aim:

The team's aim for this project is to design and develop a gamified task tracking mobile application that helps combat a user's task-completion-related procrastination by making it a more engaging and interactive process.

Objectives:

1. Research into gamification
2. Design and develop a new application
3. Test and improve that application
4. Release an alpha-stage of that application
5. Design and develop a website

Requirement Capturing Method:

All requirements listed below have been captured through research into existing solutions and a survey designed by the team. Details regarding these existing solutions are available in section 2 of this document.

Functional Requirements:

Must-Have Requirements:

- Users must be able to create, modify and delete a to-do list
- Users must be able to have multiple to-do lists simultaneously
- Users must be able to earn in-game money when they mark a task as complete
- Users must be able to spend their earned money to purchase pet care items for the game via the in-app store
- Users must be able to modify their pet's happiness, hunger and thirst levels using pet care items purchased from the in-app store
- Users must receive in-app alerts when they run out of in-game pet care resources
- The application must automatically reduce the pet's hunger, thirst and happiness levels daily
- Users must receive a daily notification to remind them of their to-do lists
- All user and pet data must be stored in an offline database
- The application must have an accompanying website outlining what the application is and to showcase more information about the project team

Should-Have Requirements:

- Users should lose in-game money when they uncheck a previously completed task
- Users should be able to register and create an account
- Users should sign in using their account credentials to get access to the application
- The application should display changes to the pet's hunger, thirst and happiness levels in real-time

Could-Have Requirements:

- Users could be able to share their in-game pet via their social media
- The application could have a web version
- The application could have a leader board to compare user stats against
- The application could host all user data on a cloud-based database instead of an offline database

Non-Functional Requirements:

Operational:

- The application must be able to run on Android devices
- Website must be designed to be responsive
- The application could be able to run on iOS devices

Performance:

- The offline database must be instantly updated whenever any changes occur

Accessibility:

- The application must have a friendly User Interface (UI)
- The application could have light and dark-mode options

Security:

- All user account credentials must be stored in remote cloud storage to reduce risks of account security compromises

Cultural:

- The application could have multiple language options

Section 2 - Survey of Existing Solutions:

Habitica:

An existing application that is comparable to Task Keeper is an online task management game called Habitica (Habitica, 2021). Habitica is a similar application to Task Keeper, allowing users to gamify tasks to encourage their completion. While the applications are similar, Habitica adopts a role-playing game (RPG) style, while Task Keeper focuses on a caretaker style game, in which the user takes care of a virtual pet. The primary advantages and disadvantages of the Habitica application are tabulated below -

Advantages	Disadvantages
Allows for the gamification of various tasks e.g., fitness and work.	The game is in RPG style. This style of game with varying buffs, stats and other data can be somewhat overwhelming to a player who is not familiar with RPG's.
Multiplayer parties allow for accountability, companionship, and support.	Some of the game's customization content is locked behind a paywall.
An extensive system of levelling up, with boss fights to keep users entertained. Also has various classes e.g., warrior and mage allow for lots of content and replay-ability.	The character customization is limited, especially without the paid content.

Task Keeper differs from Habitica in several ways, namely because it will be focused on taking care of a pet as opposed to undertaking quests and defeating bosses. The team aims to have several features Habitica is lacking, including users being able to share their in-game character via social media as well as having a leader board to allow players to compare progress. The team hopes these features will act as a further incentive for users to complete their tasks and will add an air of competitiveness to the game.

Gamifying a task must be done in such a way that simplifies the task the user has set, the "have to" task e.g. washing the dishes, but the "want to" task, e.g. playing the game, should be more complex and the user should feel satisfaction out of the completion of this task. The application should provide clear and interesting challenges to the user as well as providing clear and structured goals with feedback. (Gamification: State of the Art, 2012)

To gauge customer interest, requirements and demand for a gamified task manager application, the team conducted an online survey. Within this survey (Appendix A), individuals were asked various questions including whether they currently used a gamified task manager, if they would like to and what features they would like to see in such an application.

Habit Hunter:

Another similar application to Task Keeper is the game Habit Hunter (Active User Co., LTD, n.d.). Habit Hunter is effectively a gamified task manager, much like the team's application and Habitica (Habitica, 2021). Similarly, just like Habitica, it differs in several ways. Habit Hunter follows a similar formula to Habitica in that it is fundamentally a task manager, with RPG aspects to it. It is available on both the Google Play Store and Apple's App Store but lacks a web app version of the game.

The application allows for the creation of a character to represent the user who will level up and become more powerful as the user completes their tasks. Rewards such as gold, skills and armour are gained upon completions of tasks. The user can

Figure 1: Screenshot of Habit Hunter's In-App Screen

then use their character and items to progress upon a campaign map defeating monsters along the way and unlocking new content. Habit Hunter, much like Habitica, differs from the team's app in that it is in the style of a role-playing game while Task Keeper is a caretaker style game. It does however have additional features Habitica is lacking such as being able to share activities in the game on social media and a winder range of activities present better in a map visible to the player, as pictured below.

EpicWin:

The Final gamified task manager application that will be discussed is Epic (Rexbox, n.d.). EpicWin is available on both the Google Play Store and Apple's App Store yet also lacks a web app version of the game. This task manager also uses an RPG style and is similar to Habit Hunter in this respect. It has additional similarities including a campaign map and the ability to share results on social media, both of which Habitica lacks.

The app is beholden to the same fundamental concept as Task Keeper – that being to incentivise users to complete tasks in a gamified and fun environment. The app however lacks both customization and content, being quite repetitive and the character customization is very limited, especially relative to Habitica, but also Habit Hunter. The game, while free on the Google Play Store is locked behind a paywall on IOS and Apple-based devices.

To conclude, many of the existing applications are comparable due to their focus on an RPG style of gamified task manager, as opposed to the team's idea of a caretaker system. They also differ in features with some contained advanced features such as multiplayer and paywalls to access content, or in the case of EpicWin, to access the game on Apple devices. Web apps are another feature lacking from both Habit Hunter and EpicWin but are available from Habitica.

Demand for Product:

There is a clear desire for software that can reduce procrastination and increase productivity. Collectively, Habitica (Habitica, 2021), Habit Hunter (Active User Co., LTD, n.d.) and Epic Win (Rexbox, n.d.) have amassed over "1,100,000,000" global downloads on the Google Play Store. This is an indication that there is demand for this type of application. Additionally, the responses received from the questions asked in the survey found in Appendix A further emphasise this point, because the majority of participants agreed that they were looking for something that could improve their productivity.

Section 3 – New Ideas:

This project is focused on creating Task Keeper - a gamified productivity/to-do list tracking mobile application. The premise of this app is to encourage users to complete a set of tasks. Following the completion of a set task, the user will be rewarded with money that will be spent taking care of their pet. The user can spend this money on either food, water or toys that correspond to their pet's level of hunger, thirst, and happiness, respectively. Though it is important to bear in mind that should the user decide to untick a task once it was marked as complete, they will lose the money they were rewarded with (or a negative amount if the money was already spent). The pet's levels of hunger, thirst and happiness will diminish over time, so Task Keeper should promote replay-ability, making it a useful tool for anyone trying to get work done consistently.

The team decided to market this app towards university students (predominantly) based on the ease at which their views and opinions could be accessed through official communication channels (such as university emails containing surveys or in-person interviews with flatmates) where questions regarding the app could be asked. As Task Keeper follows an agile-like methodology, the team knows that client feedback is important at every stage of this project and therefore will be using these channels when appropriate, to progress with the project.

Due to how simple Task Keeper is, the team felt it was redundant to have a conventional homepage often found in modern applications. Instead, the team decided that a side-bar menu would be a suitable alternative that would still enable the user to navigate through each component of the app with ease.

An application such as (Habitica, 2021) hosts a lot of functionality, and can sometimes seem busy to users trying to utilise the app's main purpose. Therefore, as Task Keeper aids to reduce procrastination, the team has decided to limit the total amount of unnecessary noise on the screen by focusing on each page's main purpose. For instance, limiting the task page to its intended functionality rather than including additional, useless extras. The team believes that this should reduce the time wasted among users when using the app.

Section 4 – Design and Development:

Development Phase Gantt Chart:

To ensure that the team was able to deliver the final submission on time, the team designed a Gantt Chart (Appendix C) to cover all the necessary tasks, milestones and deliverables involved in the development process. The team adopted an agile approach for the project methodology, each sprint involves the completion of certain tasks followed by both a milestone and deliverable which summarises the expected outcome of each sprint stage.

Risk Analysis and Mitigation Plan:

The team also created a table (Appendix D) that showcases the probability of occurrence and potential impact severity of potential risks that may occur during the duration of this project. Each risk is accompanied by a mitigation plan to prevent it from impacting the project success.

Project Management:

Team Roles:

During the initial process of determining team roles, members of the team expressed that role assignment should be decided based on an individual's suitability for the role. Upon researching studies surrounding teamwork and roles, the team was introduced to Belbin's Team Roles study (BELBIN Associates, 2021). The study's research showed that the "most successful teams" comprised of individuals that had a "diverse mix of behaviours" These behaviours were put into "nine clusters" by Dr Meredith Belbin and his team. Due to the test requiring a fee, the team opted to instead review the details of the different roles available on the official website and discuss what each member would categorise themselves as.

Ethan resonated most with the 'co-ordinator' and 'completer finisher roles, hence, the team felt that he would be the perfect fit to lead the team as our Project Manager. Hannah identified with the 'specialist' and 'implementer' roles as she has more of a technical background through her work with the university's Developer's Society (NTU DevSoc, n.d.). Thus, the team felt she would be best suited for the Systems Manager role in addition to the Deputy Project Manager role. Upon further discussion, Ben felt that he related to the role of 'resource investigator' due to his ability to conduct extensive research. Moreover, the team also felt that Ben shared similar behaviours to the 'team worker' role. Thus, Ben was assigned the primary role of Head of Market Research and a secondary role as a software developer to support Hannah in the application's development. Owen resonated with the roles of 'plant' and 'monitor evaluator' due to his experience with web development and ability to review major decisions from a logical perspective. Therefore, the team felt that he would be best suited for the primary role of Head of Testing and the secondary role of System Debugger as his practices would allow for him to approach all the application's testing and debugging processes without bias and would review them objectively in his evaluations.

Finally, due to Simon not being present (Appendix F – Meeting Attendance Records) in the team's initial discussion and finalisation of roles, the team did not have the opportunity to discuss what roles Simon had related to within Belbin's study. In later meetings that Simon attended, the team was able to identify that he identified with the 'monitor evaluator' role's traits, thus, he was then assigned as a System Tester to support Owen and as Head of User Acceptance to ensure the team's final product was up to standard.

Communication:

Due to the project timeline occurring amidst a global pandemic, the team had to ensure that constant online communication was maintained as meeting in person was unviable and not possible. The team's first approach was establishing a channel of communication. Upon discussion of the potential options which included, Slack (Slack Technologies, Inc., 2021) and Microsoft Teams (Microsoft, 2021), the team settled on Discord (Discord, 2021). This is due to its popularity amongst members in the group, in addition to being a platform that members frequently used for other projects and purposes. This meant that it was much easier to contact each other when it was necessary. The team ruled out both Slack and Microsoft Teams due to the platforms either being not as well known by team members compared to Discord or because they were unviable for the type of communication that was due to occur.

Figure 2 depicts how the team's deputy project manager, Hannah, had structured the communications environment. The separate channels allowed for the team to keep track of different sections of work far easier and enabled ease of use when it came to looking back for information.

Team members would use the discord server as a means of keeping each other up to date on task progress. The 'pending tasks' channel was especially useful for this. Members would post their check lists (Figure 3) of pending work and tick items off upon completion. This ensured all members were aware of any changes or progress occurring as the project progressed.
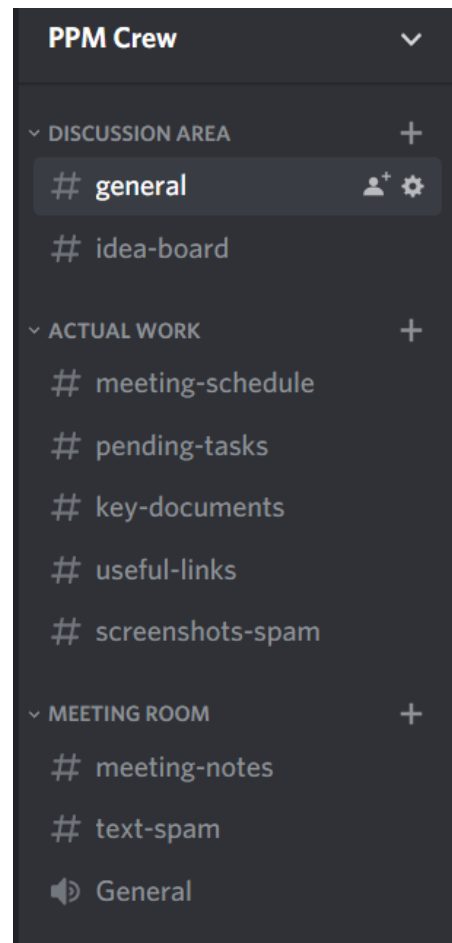
Figure 2: Screenshot of the Discord server's structure
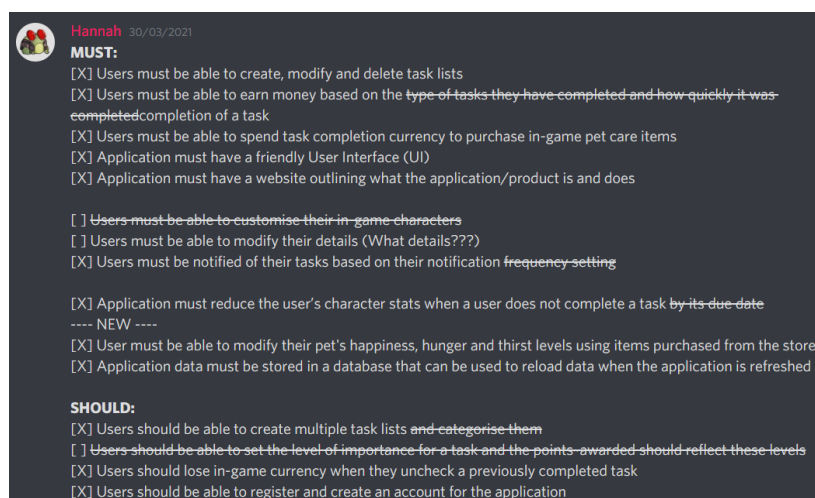
Figure 3: Screenshot of checklist message

Finally, this discord server is where the team would host the twice/thrice a week group calls in addition to the daily text-based check-ins. This constant flow of communication in addition to keeping members in the loop of project progress, also allowed members to voice out any struggles they faced or concerns they may have had.

Task Assignment:

During the team's meetings held via Discord, Ethan and Hannah would post a message in the 'meeting notes channel post-meeting with a summary of all the tasks assigned to each member during the session. Task assignment is based on the project's requirements and Gantt chart. The team would often first discuss all progress made between the previous and current meetings before moving onto what tasks need to be done next to propel the project's development further.

The team would then discuss which team member's skill set would be best suited for the task and then assign it to that member. All tasks are expected to be completed before the next meeting unless explicitly stated otherwise. This was done to ensure that a team member would not be given a particularly difficult task as the team also had to consider other incoming module project deadlines that may potentially clash with task assignment for each member. Included below are some screenshots of post-meeting task assignment summaries.

Figure 4: Examples of Task Assignment

Workflow:

A key tool in every programmer's toolbelt is the ability to use Git and GitHub. One group of researchers (Yan Hu, 2016) noted that it is an "essential part" of "social coding" especially for "decentralized teamwork" as it makes the development process "more productive". Due to the team having to work remotely for the entire duration of the project, we found that GitHub would be the best platform to collaborate on the application and website's development and visually track project progress through the Git commit history. GitHub provided the team with the opportunity to work together on the project simultaneously while being miles apart from each other.

GitHub repositories also act as a cloud-based backup of the project's source code. This ensures that if any one member experiences loss or corruption of data on their side, the source code will remain intact via GitHub.

Version control software also allows users to revert changes to the repository if necessary. This feature was especially useful during the development phase wherein the team decided that the older implementation of a solution was far better than the newest implementation.

Highlighted below in Figure 5 is the file structure of our project's repository. It is neatly organised using separate folders for ease of use in searching for a particular file.

Subsequently, the team also decided on the use of meaningful commit names as this allows for other members to scroll through the commit history and quickly make note of any major changes that have been made by another member (Figure 6).
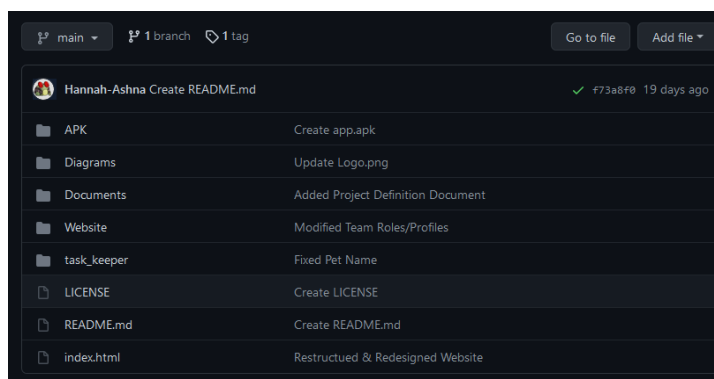

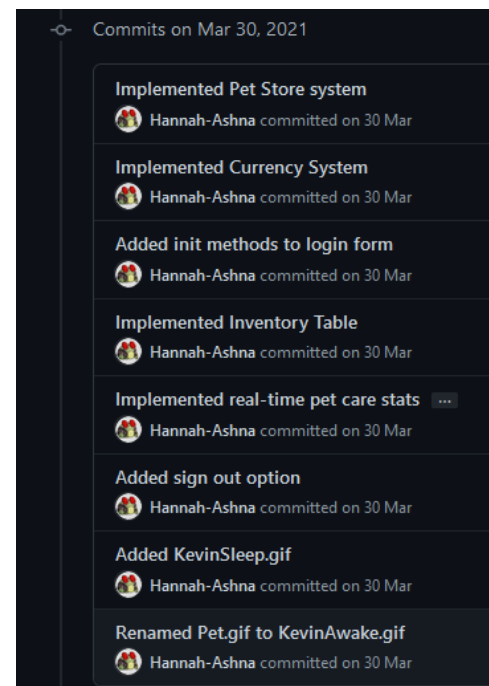
Figure 5: Screenshot of the Git Repository

Figure 6: Screenshot of the Git Commit History

Entity Relationship Diagram (ERD):

To better visualise the relationships between the offline databases' entities, the team designed an Entity Relationship Diagram (ERD). Highlighted in Figure 16 (Appendix B) are the four tables used to store data offline and locally, within the app.

The 'list' table stores a unique ID for each list created by the user, the title of the list and its description (both provided by the user). Each list will have a one-to-many relationship with the 'to-do' database table. As one list can have 0 to many to-do items for a user to mark as complete, however, each to-do item can only belong to one list (the one it was assigned to upon creation).

Moving on, the user 'Inventory' table is used to store all user data. This includes a unique ID value that could be used in further development of the app when comparing user stats against other users. In addition to this, it also stores a user's funds and purchased item amounts. The user's money is updated with the number of 'to-do' items that have been marked as true for the isDone boolean. The inventory table also stores a LoginDate parameter which is used to determine how long ago the user had last used the application. The purpose of this parameter is discussed in the Background Task Execution sub-heading of the code design section of this document.

Finally, the 'pet' table contains all the data related to the pet's statistics. This table also includes a unique ID value so that the app can be expanded upon in the future to include multiple pet options for users instead of just one. The pet's hunger, thirst and happiness values are updated with how much of a user's purchased items are used to boost the pet's stats. These are all handled using internal functions within the Database handler class.

To-Do List Database Class Diagram:

A key aspect of this application's functionality lies within its ability to store, manipulate, and remove a user's to-do list and its task items. Hence, the team decided that an offline database connected to the other aspects of the application would be the best approach to ensure that the data could be accessed for use elsewhere within the app if required.

Figure 17 (Appendix B) focuses on the use of the DatabaseTool class concerning the creation, management and maintenance of the offline database used for the user's task list data.

It interacts with the TaskList class to store all the necessary details for any one instance of a TaskList when a user creates one via the TaskPage class. Similarly, it also uses the ToDo class to store the specific data associated with any one instance of a checklist task item.

All of this data is then displayed through the TaskPage class and all the different TaskLists and their pages are linked and displayed in the Task Class build screen. Hence, many TaskPages are contained within the one Tasks class.

Other Diagrams:

To ensure the team's development phase would go smoothly, several other diagrams were created to aid in the visualisation process of the application's framework. These diagrams and their descriptions can be found in Appendix B.

Code Design:

This section aims to provide an overview of the application's (app) significant development phases. Highlighted below are some key implementation decisions that were made and a few challenges that arose during development and the solutions that the team designed to resolve them.

Background Task Execution:

As per one of the requirements, the application needed to reduce the pet's statistics daily. The team's initial plan was to implement a background task that would run even when the app was closed. However, the team soon ran into issues surrounding threading and modifying the app's user permissions. Given that the team is new to app development and that mastering this specific aspect of android app development would take too long, the team elected to implement an alternative solution that would be equally as effective and not cause users to worry about what they may perceive as suspicious background execution of code.
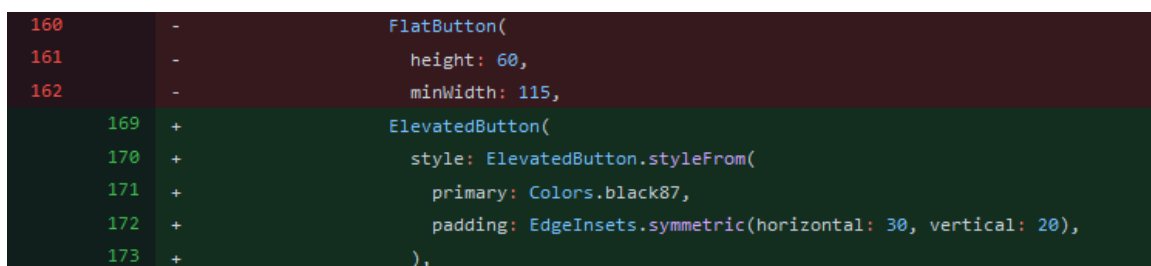
The solution involves storing the date of when a user loads up the app within the offline database. Whenever the user then opens the Pet page within the app, a check is first made to see how many days it has been since they last used the app. If it has been more than 1, the pet's statistics are updated accordingly, and the most recent date is stored in the database to be used next time.

Hence, this approach allows the app to make the necessary changes to the pet's statistics to fulfil the project's 'must have' requirement.

Deprecated Functions:

Midway through development, Flutter had released an update (Flutter Dev, 2021) that resulted in certain types of user interface (UI) buttons becoming deprecated. This meant that the code still retained some level of functionality but meant that it could cause issues in the future. To ensure that the app's code would be up to date upon the final submission the team endeavoured to rework the buttons to use an alternative button type.

The buttons now use the Elevated Button type (highlighted in green) instead of the Flat Button type (highlighted in red) as depicted via the team's Git commit log in Figure 7.

```
160    -                  FlatButton(
161    -                      height: 60,
162    -                      minWidth: 115,
       169  +              ElevatedButton(
       170  +                  style: ElevatedButton.styleFrom(
       171  +                    primary: Colors.black87,
       172  +                    padding: EdgeInsets.symmetric(horizontal: 30, vertical: 20),
       173  +                  ),
```

Figure 7: Screenshot of Git Commit Log

Dynamic Pet Statistics:

For the display of Kevin the Chicken's Pet statistics (happiness, hunger and thirst), the team wanted to have it dynamically reflected within the app. This meant that whenever a user fed Kevin, for example, the hunger bar would update immediately with simple animation. To do this, the team used the Flutter Charting (charts_flutter, 2021) external library.

To simulate the look of a traditional game character's statistics status bars, the team used a horizontal bar chart.

```
List<PetDataModel> data = [
PetDataModel(
    title: "Hunger",
    value: 0,
    barColor: charts.ColorUtil.fromDartColor(Colors.red),
),
PetDataModel(
    title: "Thirst",
    value: 0,
    barColor: charts.ColorUtil.fromDartColor(Colors.blue),
),
PetDataModel(
    title: "Happiness",
    value: 0,
    barColor:
charts.ColorUtil.fromDartColor(Colors.green),
),
```

Figure 8: Source Code of Model Data Format

During this development process, an issue arose. the team quickly realised that the bar chart always needed to be initialised first using the model data format (Figure 8) when accessed. This meant that when the user opened the Pet page on the app, they would see Kevin's stats as being all 0 and it would only update with the correct stats stored within the database once the user clicked any of the buttons pictured in the image below.
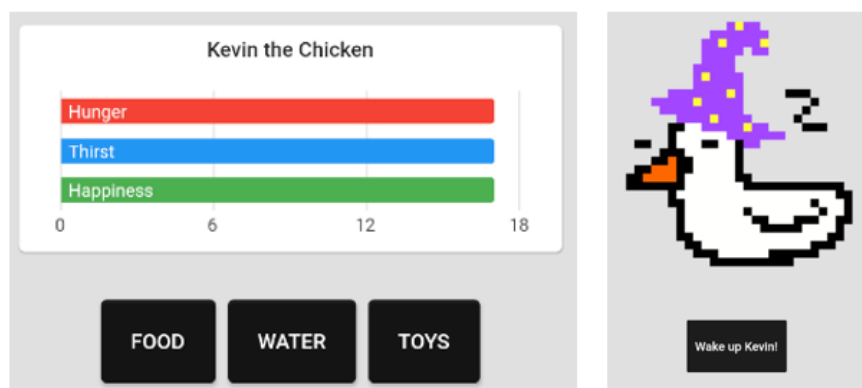

Figure 9: The Pet's stats display (Left) and the Pet wake up screen (Right)

As a work around, the team implemented the 'wake Kevin up' button (Figure 9). The primary purpose of the button was to trigger the update function for the bar chart before the user got to view the chart. An additional bonus of the use of this button is an added user and pet interaction scenario.

Section 5 - Evaluation of Product:

In accordance with the project plan, the team allocated sufficient time to review and test all of the application's implemented features to ensure that the project had met its objectives and requirements within the allocated time frame.

Highlighted below are the key objectives and requirements of the project. The testing and survey results referenced in this section can be found in the Appendices.

Objectives:

- Research into gamification

The intention behind this objective for the project was to encourage the team to conduct in-depth, extensive research into the science behind gamification and its onion impact within the world of to-do list applications aimed at tackling procrastination. Through the survey into existing solutions (Section 2), the team successfully honed its understanding of gamification as a tool that can be used to improve our lives. Thus, aiding us in our implementation of the concept into the project's outcome, the Task Keeper Application.

- Design and develop a new application

Mobile applications development is a vital skill to have within the industry (FRG Technology Consulting, n.d.). However, it is also a new concept for all members of the team, hence, as a personal growth challenge for the team to further build upon its member's skillset as developers, the team decided to set the design and development as an objective for the project. The team was successful in the attainment of this objective as supported by the successful outcomes in the testing stages (Appendix G) and through the completion of the objective mentioned below. The team also found that its understanding of mobile applications development had significantly improved throughout the project and all members of the team are now confident with the use of popular technologies and languages such as Android Studio (Android Developers, n.d.) and Dart (Dart.dev, n.d.).

- Release an alpha-stage of that application

In conjunction with the objective above, to measure the success of the application, the team aimed to release an APK version of the application so that it could be used for real-world testing amongst a small external testing team. Upon completion of the implementation of all the 'must' and 'should' have requirements, the team created and shared the APK version with the external team. This external team comprised of members of the Developer's Society at the university (NTU DevSoc, n.d.) as the team believed the society's members would be able to provide an objective evaluation of the product. A survey was released to collect feedback. The team received outstandingly positive feedback regarding the application and the features that it provides (Appendix H) through this survey's (Google Forms, 2021) results.

- Test and improve that application

As developers, the team recognises that an application constantly needs to be reviewed, tested and modified following changes occurring within the industry or within the project's plan itself. There were not many significant updates that occurred during the duration of this project that would result in a major revision of the application's source code. However, as outlined in the Code Design section above, the team did need to modify parts of the source code as certain functions were deprecated because of a Flutter update that occurred mid-development. Through the team's rigorous approach to testing immediately after a new

feature was developed, it helped the application avoid any major bugs from being uncovered in a much later stage in the project. This decision to test frequently and improve upon the application is in line with the team's risk analysis and mitigation plan (Appendix D).

- Design and develop a website

Since the team consists of students preparing to break into the tech industry, a detailed portfolio to showcase one's work is a crucial factor in how an individual can present themselves to employers. Hence, the team decided to use the website for this project to showcase the team's capabilities in a simple and accessible way. Despite the project being available via the GitHub repository, the website (Task Keeper Team, 2021) allows non-tech-savvy individuals to engage with the project and learn more about it without the technical barrier of Git-related knowledge.

Primary Requirements:

This section includes five requirements that the team viewed as being incredibly important to the success of the project. Outlined below is a detailed discussion of these features.

- The application should run on an Android device.

With Flutter and Android Studio, the team has been able to produce a functional APK version that is capable of being run on an external Android device outside of the simulated environment within Android Studio itself. Subsequently, this requirement has helped the team carry out 'real-world' tests (Appendix G – Test ID 8) for the application as we can run it on our personal devices (Lifewire, n.d.). Hence, giving the team a more detailed insight into how the application would work if it were to be released via the Android PlayStore.

- The application should let the user create, modify, and delete a to-do list.

This is one of the team's 'must-have' requirements for this project. This is due to the team believing that without the ability to create, modify and delete a to-do list, the application would not fulfil its aim. Through the testing process (Appendix G – Test ID 1, 2 and 3) the team was able to determine that the implementation of these features was successful.

- The application should provide a user with in-game currency upon a task's completion.

The purpose of this feature was to provide the users with an incentive to complete their tasks. Thus, it is also considered one of the team's 'must-have' requirements. To create a gamified environment within the application, the team needed to create a reward system that encouraged players to carry on with certain behaviour; in this case, being productive and completing their to-do lists to earn in-game currency to take care of their pet. The application does this successfully by immediately updating the offline database to reflect the increase or decrease in a user's funds (Appendix G – Test ID 5).

- The application should enable the user to purchase in-game items using the earned currency.

In conjunction with the requirement above, this too is considered a must-have requirement for the application as it relates to the gamification aspect. Being able to purchase food, water, and toys for a pet, allows the user to feel responsibility in the management of their pet. Thus, encouraging them to complete their to-do lists to earn in-game currency so that they may continue to purchase the in-app items. As highlighted through one of the tests (Appendix G – Test ID 6), the team was able to successfully integrate this feature within the application.

Other Requirements:

Aside from these key requirements that have been covered above, the team was also able to integrate all the other 'must' and 'should' have features into the application. Most of these features focus on further developing the functionality and the gamification of the application. This includes features such as the ability to have multiple to-do lists (Appendix G – Test ID 4) as users may want to separate their tasks based on specific categories; or features such as in-app alerts or daily notifications to keep users engaged with the application (Appendix G – Test ID 8).

The application also automatically handles changes to the pet's hunger, thirst, and happiness values whenever the user accesses the app to ensure that the values are decreased daily. Therefore, encouraging users to complete tasks for in-game currency as they must purchase pet-care items to increase those values again. This is another feature the team uses to ensure user engagement with the app is high (Appendix G – Test ID 7).

When reviewing the need for an offline versus a cloud-based database the team recognised that given the time-frame limits of the project, the offline database was the preferable route due to it being far easier to set up, manage and work within the chosen development environment. Moreover, users retain a sense of privacy with their data as the team does not have access to a user's to-do lists.

Finally, the application requires users to register an account before obtaining access to the application. The credentials are managed using Google's Firebase authentication console (Appendix G – Test ID 9). Currently, the purpose of the authentication process is to be able to provide a basic level of security for the application in addition to allowing the team to obtain rough estimates of the application's user base size. Ideally, future use of the account feature would be to combine it with a cloud-based database so that users can transfer their app data onto another application with ease by simply logging in with their registered credentials.

Section 6 - Discussion and Conclusion:

Overall, the team believes that Task Keeper is a user friendly, simple, and effective app that achieves its goal of reducing procrastination using gamification. This section will be dedicated to discussing important decisions made, with some reference to the issues that arose as the project progressed.

Initially, the team had a hard time receiving a reply from Simon Fincham after numerous attempts to reach him. The team continued with the project and redistributed Simon's component of the work between the other members of the team. This issue persisted until the first project submission date (18th of January 2021). Following this date, the team continued messaging Simon. Eventually, Simon responded and started joining the team's meetings. For the breakdown of meetings and attendance of all members, see Appendix F.

The team agreed on dates, but assignments got in the way (clashed)

Simon, didn't share many ideas and thus we didn't know what to put him in the Belbin test, we just assigned him to work randomly but didn't work out because the work we did assign was late, of poor quality and had to be redone. He didn't do anything.

During development, the team made various decisions with regards to the app (Task Keeper). The team first needed to determine which software and coding language would be used to develop the app. Although difficult, the team opted for Android Studio (Android Developers, n.d.) and Flutter's Dart (Dart.dev, n.d.) due to the multitude of functionalities both tools provide. Moreover, Dart has a similar syntax to HTML, hence, the team agreed that it would be the quickest to adapt to and learn. This decision was primarily influenced by most of the team's members being experienced with varying degrees of web development.

To go alongside the app itself, the team agreed that a website should be developed to record the team's accomplishments as the project progressed. To make development easier, the team opted to use Bootstrap (Bootstrap Team, 2011), which is one of the most popular cascading style sheet (CSS) frameworks that grants a large amount of customizability. Bootstrap also provides built-in responsiveness, enabling the website to be viewed on all manner of devices.

Although Task Keeper provides the intended functionality initially planned, it would require the implementation of additional accessibility-focused features such as language translation and colour-blind mode (among others) to make it accessible to anyone that may need it, per the Equality Act (HM Government, 2010) if published on the Apple or Google Play Store.

In terms of database deployment, the team opted for Firebase. Firebase enables the team to store the user's data offline, making it easier to comply with the Data Protection Act (HM Government, 2018) by making the database virtually unattackable, resulting in minimal issues concerning the privacy of user data. On one hand, this can be massively beneficial, however, it is very difficult to migrate data through Firebase due to there being a lack of Standard Query Language (SQL) features as it utilises the JavaScript Object Notation (JSON) format. Because of this, publishing Task Keeper on the Apple Store may require a reasonable amount of refactoring to enable cross-compatibility with both Android and IOS devices.

Both Hannah Ashna and Ethan Wilde (deputy project manager and project manager respectively) did not believe it was entirely necessary to utilise external platforms, such as (ClickUp, 2021) to manage this project. The team communicated daily on the private Discord (Discord, 2021) server Hannah set up at the start of the project. Furthermore, the team sent various images and screenshots of work that was underway throughout the project.

However, If the project were to be conducted over a longer period and required more efficient resource management, the team would use ClickUp (or something similar) due to the plethora of features and quality of life tools this type of software tends to provide.

Due to all the team's work being conducted online, it was paramount that backups were made and updated consistently to ensure that work was not lost, even though the platform the team were using for collaboration (Google Docs & GitHub) are very reliable and automatically saves work at regular intervals.

In conclusion, the team is very happy with the project's outcome as most members performed their assigned roles to the best of their ability. Aside from some of the issues listed in this section, the project was very well managed. The team met consistently (as shown in Appendix F) and most members contributed when assigned with work to complete. It could be argued that the team's discord server played a big role in the success of this project. It was used to both communicate ideas and share work, which played a big part in enabling the team to complete work on time and meet deadlines from both the team's Gantt chart and the University's submission dates.

Section 7 - Professional, Social, Ethical and Legal issues:

Introduction:

When planning, developing and implementing a system for purpose, several issues need to be considered. The issues discussed in this section will be regarding those of Professional, Social, Ethical and Legal concerns to the team. The team has collated a list of issues that must be followed to ensure integrity and confidentiality throughout this project. To achieve this, the team has agreed to develop and implement a gamified task tracker mobile application. By working out the potential risks, the team can aim to reduce the chance of these risks occurring as much as possible, especially the ones that have been highlighted to link with the project. When the professional issues were first brought to light within the team, it was evident that the British Computer Society's (British Computing Society, 2020) Code of Conduct and Code of Good Practice had to be evaluated. These Codes provide a set of guidelines that professionals within the computing industry must adhere to and so as a team it was agreed they were important. The team also agreed that it was important to consider the Software Engineering Code of Ethics, published by the Institute of Electrical and Electronic Engineers Computer Society (IEEE-CS) (ACM Ethics, 1997).

Professional Issues:

The first aspect of this project that was identified was that the consumers would expect the team to keep any data collected by the team from them, to be kept secure. The BCS Code of Conduct states that the team should "have due regard for public health, privacy, security and wellbeing of others and the environment" (British Computing Society, 2020). This means that the users of the product should be comfortable with disclosing their personal information and details when signing up to and using the gamified task tracker mobile application, provided the data is secured via measures that meet the legal standards outlined in the General Data Protection Regulation (GDPR) (Intersoft Consulting, 2018) and Data Protection Act (HM Government, 2018).

When developing the project, the team had to ensure that the project did not breach the Equality Act (HM Government, 2010). This act protects people, whether in the workplace or wider society, from all varieties of discrimination. This has been considered for all aspects of the project including the team as well as the application. Concerning team equality, the project manager ensured that each member's voices were heard, and ideas were considered for every aspect of the project including the planning, development, documentation, and evaluation.

When the team brainstormed to create the idea for this project, the initial target audience was students due to how critical feedback was to this project and how easy it was for the team to reach different teams of students from different courses. This meant that the results from the survey the team conducted would be more applicable to a larger sample size. Therefore, it gave the team a realistic representation of student opinions. However, since procrastination affects so many people, the team understood that it could be universally utilized by people of all ages, within reason. This was done to coordinate the project with the guidelines set by the code of conduct within the Software Engineering Code of Ethics (ACM Ethics, 1997). In particular, Principle 1, "Software engineers shall act consistently with the public interest".

Social Issues:

When considering the social issues linked with this project, the team needed to focus on the gamification aspect, as that is what this project is solely based on. With this being the case, the team had to ensure that the gamified part of the application was not addictive and does not outweigh the social side of the user's life as that would instead make the application become the cause of the original problem the application aims to solve, procrastination. An example of this issue is the user seeing the application as more important than their general health and social life. This would be a breach of the British Computing Society's public health section of the code of conduct, therefore not in the best interest of the public. To comply with these conditions, the team has tested the elements thoroughly, developing and implementing them in a way that will not interfere with the user's immersion and/or engagement with the application without the application becoming addictive.

Ethical issues:

When considering the ethical issues associated with this project, it was agreed that the team had to consider all potential problems and ensure all were acted upon appropriately. One of the issues that were identified, was that the user may not want their information on incomplete tasks or missed deadlines to be stored and/or disclosed with other third-party organizations or other users. Within the British Computer Society Code of Conduct (British Computing Society, 2020), it states that entities must, "have due regard for the legitimate rights of Third Parties", meaning that the third party, in this case, the end-users, must be allowed to have a say on what happens with this data. To tackle this issue, the application informs the user of what data is being collected, and how it will be processed, accessed, and stored.

Legal issues:

Regarding legal issues, the team agreed to investigate and analyse all the issues faced in terms of legalities when creating an application within this section.

The main legal issue was concerning the collection of data required by the application. For instance, to provide leader board functionality or personal user profiles (if implemented) the application must collect data from its users. Some examples of the type of data collected may include but are not limited to usernames, task names, and email addresses. As a result of this, it is paramount that members of the team do not breach the (Data Protection Act, 2018). This Act declares how data must be legally used, making sure that it is not misused or abused. A way the team has achieved this is by providing the users' information on what data is being collected, as well as enabling the user to give consent to the data being collected from the application.

As the application is relatively simple, the legal issues that need to be considered are minimal. However, the team has ensured that all professional, social, ethical, and legal responsibilities have been met with adequate measures.

References:

ACM Ethics, 1997. *Software Engineering Code | ACM.* [Online]
Available at: https://ethics.acm.org/code-of-ethics/software-engineering-code/
[Accessed 3 March 2021].

Active User Co., LTD, n.d. *Habit Hunter.* [Online]
Available at: https://habithunter.activeuser.co/
[Accessed 19 April 2021].

Android Developers, n.d. *Android Studio.* [Online]
Available at: https://developer.android.com/studio
[Accessed 18 April 2021].

Bank of England, 2020. *The impact of Covid-19 on productivity. Staff Working Paper No. 900.* [Online]
Available at: https://www.bankofengland.co.uk/-/media/boe/files/working-paper/2020/the-impact-of-covid-19-on-productivity.pdf?la=en&hash=E08581632820D88A45D2FCFAFF54C6D9FB04E27C
[Accessed 21 February 2021].

BELBIN Associates, 2021. *Belbin Team Roles | Belbin.* [Online]
Available at: https://www.belbin.com/about/belbin-team-roles
[Accessed 4 May 2021].

Bhanji, J. a. D. M., 2013. *The social brain and reward: social information processing in the human striatum,* Newark: The National Center for Biotechnology.

Bootstrap Team, 2011. *Bootstrap.* [Online]
Available at: https://getbootstrap.com/
[Accessed 20 April 2021].

British Computing Society, 2020. *BCS Code of Conduct.* [Online]
Available at: www.bcs.org/membership/become-a-member/bcs-code-of-conduct/
[Accessed 6 January 2021].

charts_flutter, 2021. *charts_flutter | Flutter Package.* [Online]
Available at: https://pub.dev/packages/charts_flutter
[Accessed 14 April 2021].

CIPD, 2020. *Impact of COVID-19 on working lives.* [Online]
Available at: https://www.cipd.co.uk/knowledge/work/trends/goodwork/covid-impact
[Accessed 7 February 2021].

ClickUp, 2021. *ClickUp.* [Online]
Available at: https://clickup.com/
[Accessed 22 April 2021].

Dart.dev, n.d. *Dart Programming Language.* [Online]
Available at: https://dart.dev/
[Accessed 18 April 2021].

Discord, 2021. *Discord.* [Online]
Available at: https://discord.com/
[Accessed 22 April 2021].

Flutter Dev, 2021. *New Buttons and Button Themes.* [Online]
Available at: https://flutter.dev/docs/release/breaking-changes/buttons
[Accessed 14 April 2021].

FRG Technology Consulting, n.d. *Increased demand for mobile app development skills.* [Online]
Available at: https://www.frgconsulting.com/blog/demand-mobile-app-development-skills/
[Accessed 3 May 2021].

Google Forms, 2021. *APK Feedback Form.* [Online]
Available at: https://forms.gle/9NjSrSc6LZDemC6a6
[Accessed 03 May 2021].

Google Forms, 2021. *Procrastination & To-Do Lists.* [Online]
Available at: https://forms.gle/8ypnghKL9gB7Hn2k9/
[Accessed 5 March 2021].

Habitica, 2021. *Habitica - Gamify Your Life.* [Online]
Available at: https://habitica.com
[Accessed 24 January 2021].

HM Government, 2010. *Equality Act 2010.* [Online]
Available at: https://www.legislation.gov.uk/ukpga/2010/15/contents
[Accessed 3 March 2021].

HM Government, 2018. *Data Protection Act 2018.* [Online]
Available at: https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted
[Accessed 3 March 2021].

Intersoft Consulting, 2018. *General Data Protection Regulation (GDPR).* [Online]
Available at: https://gdpr-info.eu
[Accessed 3 March 2021].

Kuo-Chen Li, S.-Y. C. C.-W. W., 2018. *Can a Game-based Productivity Tool Improve Procrastination?,*
s.l.: Scienpress Ltd.

Lifewire, n.d. *APK File (What it is and how to open one).* [Online]
Available at: https://www.lifewire.com/apk-file-4152929
[Accessed 18 April 2021].

MacNaught, S., 2019. *Procrastination Statistics UK 2020 (And How to Stop Procrastinating).* [Online]
Available at: https://www.microbizmag.co.uk/procrastination-
statistics/#:~:text=Procrastination%20Statistics%20%E2%80%93%20How%20Many%20of%20us%20
Procrastinate%3F&text=Just%2015.6%25%20of%20people%20claim,14.4%25%20admit%20to%20pr
ocrastinating%20rarely
[Accessed 31 January 2021].

Microsoft, 2021. *Microsoft Teams.* [Online]
Available at: https://www.microsoft.com/en-us/microsoft-teams/group-chat-software
[Accessed 6 May 2021].

NTU DevSoc, n.d. *The Developer's Society.* [Online]
Available at: https://devsoc.co.uk
[Accessed 22 April 2021].

Pechenkina, E. L. D. O. G. E. D. a. H. D., 2017. Using a gamified mobile app to increase student engagement, retention and academic achievement. *Int J Educ Technol High Educ 14.*

Rexbox, n.d. *Epic Win - Level up Your Life.* [Online]
Available at: http://www.rexbox.co.uk/epicwin/
[Accessed 23 April 2021].

Slack Technologies, Inc., 2021. *Where Work Happens | Slack.* [Online]
Available at: https://slack.com/intl/en-gb/
[Accessed 6 May 2021].

Task Keeper Team, 2021. *Task Keeper.* [Online]
Available at: https://hannah-ashna.github.io/Task-Keeper/
[Accessed 6 May 2021].

Weber, M. W. B. S. F. P. M. K. B. A. N. a. G. F., 2012. *Gamification: State of the Art Definition and Utilization.,* Louisiana: Institute of Media Informatics.

Yan Hu, J. Z. X. B. S. Y. a. Z. Y., 2016. *Influence analysis of Github repositories,* China: SpringerPlus.

Appendices:

Appendix A - Survey

Link to Survey: https://forms.gle/8ypnghKL9gB7Hn2k9

In the questionnaire, the team asked a variety of questions to gain insight into the market the team would be developing the application in and to gain an understanding of who the target audience would be and what they would expect from the team's application. The questionnaire included 14 questions over three pages, I shall be listing and expanding upon the questions below to explain their purpose and the decisions that were made as a result.

The first question the team asked was the age of the participant, this allowed the team to gauge the average age of the target audience for the team's application. The next question asked the participant to rate their level of procrastination on tasks on a scale of 1-5. This allowed the team to understand the user's need for a task manager application such as Task Keeper. The team then asked if they already use a similar task manager application and several details about it including the applications name, the user's favourite features, missing features and features they dislike. This information gave the team effective insight into what similar task manager applications possess in terms of features and how favourable consumers find them, enabling the team to take this feedback onboard in the development of Task Keeper to produce a unique and interesting end product.

The next questions the team asked was whether the participant believed a gamified task manager would help combat procrastination and if they would use such an application. This allowed the team to gain a consensus as to how effective consumers believe a gamified task manager may be and if they would be willing to use one. Additionally, the participants were asked to explain why they would or would not use a gamified task manager and what they find engaging in a game. This gave the team further insight as to why users do or do not use gamified task managers and their reasons for it, further allowing the team to differentiate the application from others using this feedback. The next question asked the participant as to what they believed, from a predetermined list, would be the most important gamification feature for the team's application. This allowed the applicant to give the team their feedback on which features should be prioritised in the team's application and allowed the choice for "other" options for ideas the team had not yet considered. The second to last question asked the applicant if they believe both an app and a web app would increase their likelihood of using the application. From this, the team could determine whether the application is more suited to be exclusively on the android app store, or also available online to a wider audience. Finally, the last question asked was whether the applicant had features they would like to see implemented within the team's application. This gave the team additional ideas for the gamified application and a better understanding of consumer preferences.

Procrastination & To-Do lists

Hi there! We're a group of students looking to get some statistical data to support a project we're looking to begin working on.

Please do answer this survey as truthfully as possible as it'll be used to help guide us towards the implementation of certain features.

*Required

What age bracket are you in? *

◯ 18 or below

◯ 19 - 21

◯ 22 or older

How would you rate your level of procrastination on any given task? *

	1	2	3	4	5	
Extremely Low	◯	◯	◯	◯	◯	Extremely High

Do you use any task tracking (to-do list) applications? *

◯ Yes

◯ No

Figure 10: Survey Questions Part 1

Figure 11: Survey Questions Part 2

Gamified To-Do lists

We're hoping to gamify the process of completing tasks by combining a to-do list with a game like system to encourage a user to complete their tasks.

Do you think making to-do lists fun would help combat procrastination to some degree? *

○ Yes

○ No

Would you use a gamified to-do list application? *

○ Yes

○ No

Please explain why you said yes or no

Your answer

What engages you in a game? *

Your answer

Please select the feature you believe to be the most important for this idea? *

○ Customisable character

○ Points/Ranking system

○ An interesting plotline

○ Shareable in-app stats (via socials)

○ Unlockable features

○ Other: _____

Would having both an app and a web version increase the likelihood of you using it? *

○ Yes

○ No

Any other features you would like to see implemented? *

Your answer

Back   Submit

Figure 12: Survey Questions Part 3

Appendix B - Diagrams:

Flowcharts:

Before starting development, the team took the necessary steps to plan and produce suitable charts and diagrams to represent how the project would be undertaken from a development point of view. The following three flowcharts represent how three of the main pages within Task Keeper will work.

Login and Account Creation Functionality:
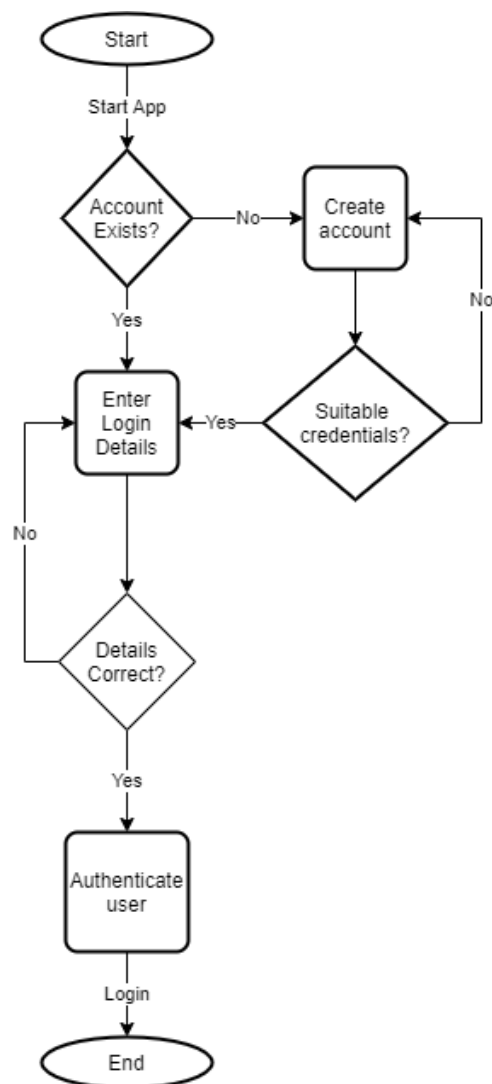
Figure 13: Login and Account Creation Flowchart

This flowchart represents a step-by-step process of how the user will interact with the create account/login page on Task Keeper. The diamond shapes portray decisions and choices that the user must choose. For instance, they need to choose between logging in with existing credentials or creating a new account with new credentials.

To-Do List Functionality:
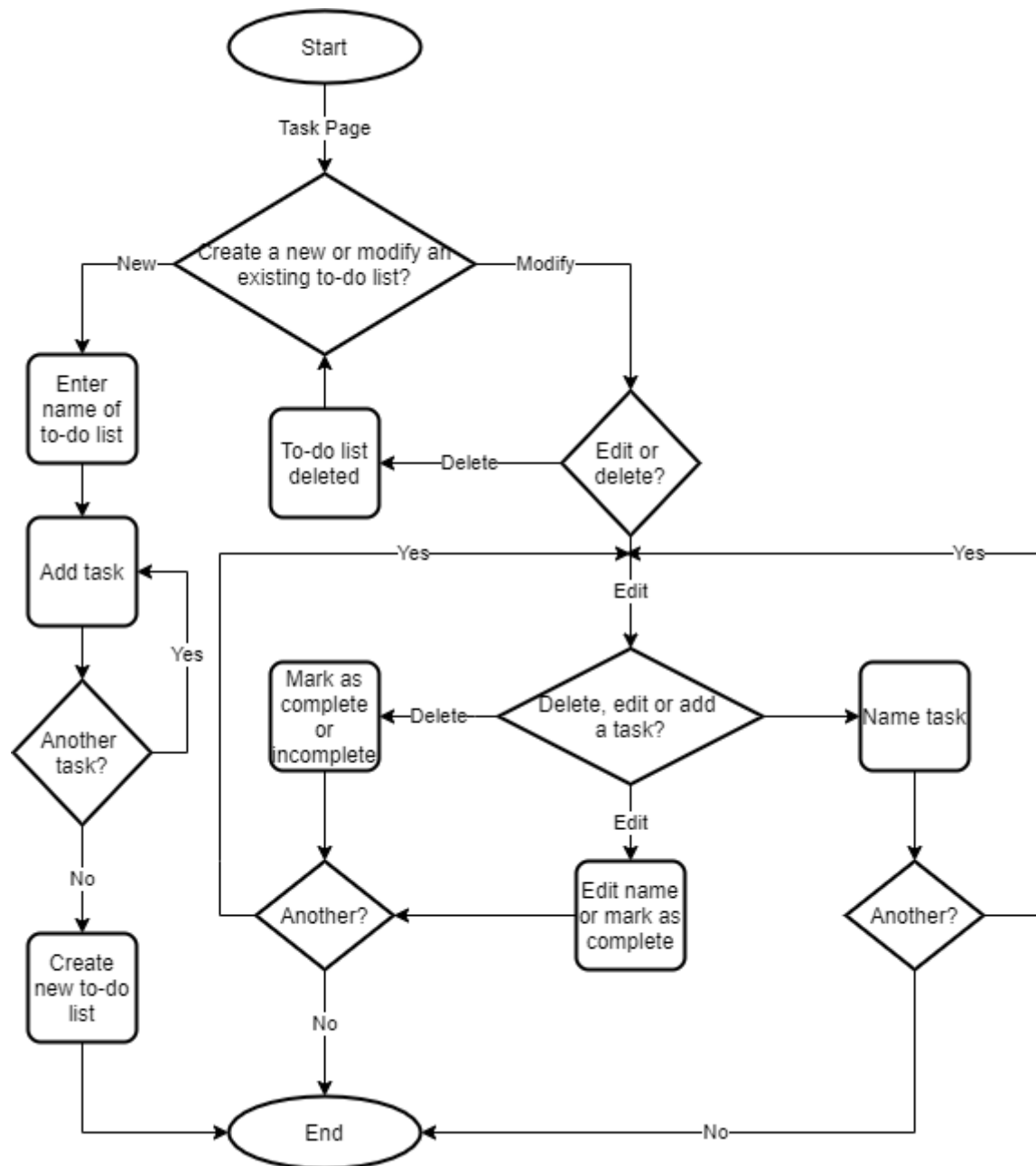
Figure 14: To-Do List Flowchart

In a similar way to flowchart one, flowchart two represents the step-by-step process that will be followed by the user when they interact with the task page. This process is more complicated than the first, but this is required as each decision is necessary to the app to enable it to provide its intended functionality. For instance, offering the ability to delete, create and edit tasks is crucial to providing the to-do list system in Task Keeper.

Figure 15: Pet Maintenance Flowchart

The pet maintenance functionality is the component responsible for providing the gamified system to Task Keeper. Otherwise, the app would be a standalone to-do list tracker not able to achieve the team's goal of reducing procrastination. For instance, there would not be a fun element prompting the user to launch, use, and return to the app if they were only able to create to-do lists.

Entity Relationship Diagram (ERD):
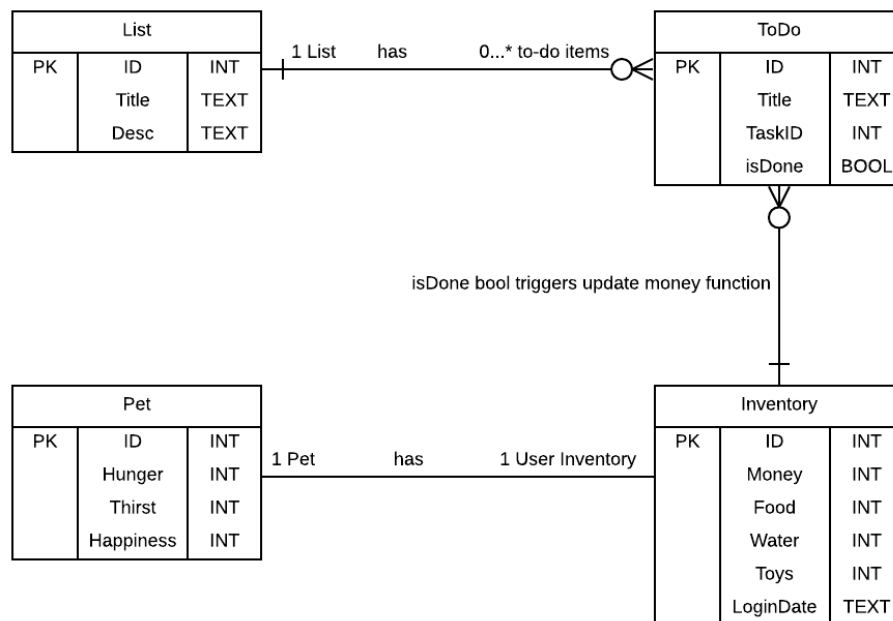
Figure 16: Entity Relationship Diagram

To-Do List Class Diagram:
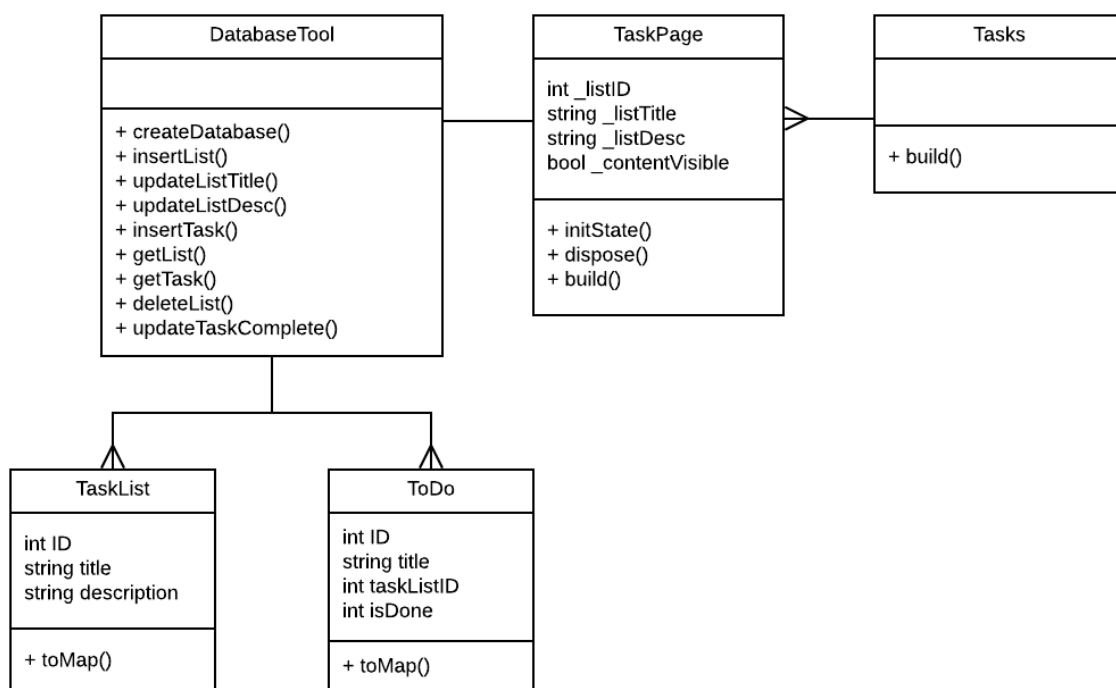
Figure 17: Class Diagram for App's Database

Use-Case Diagrams:

Login Scenario:
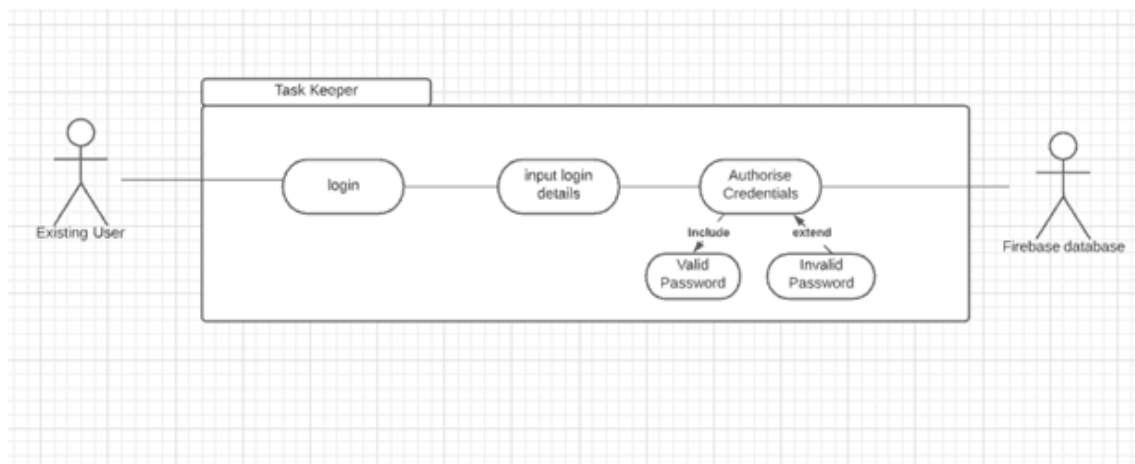
Figure 18: Login Use-Case

This Use Case Diagram describes the process, the user undertakes when accessing Task Keeper for the first time. The two actors (existing user and firebase database) initiate the system and are significant throughout the login process. When starting up the application the user has to log in via inputting their login details. The firebase database would then authorise the credentials of the user's details and check if their password is valid or not.

Pet-Maintenance:
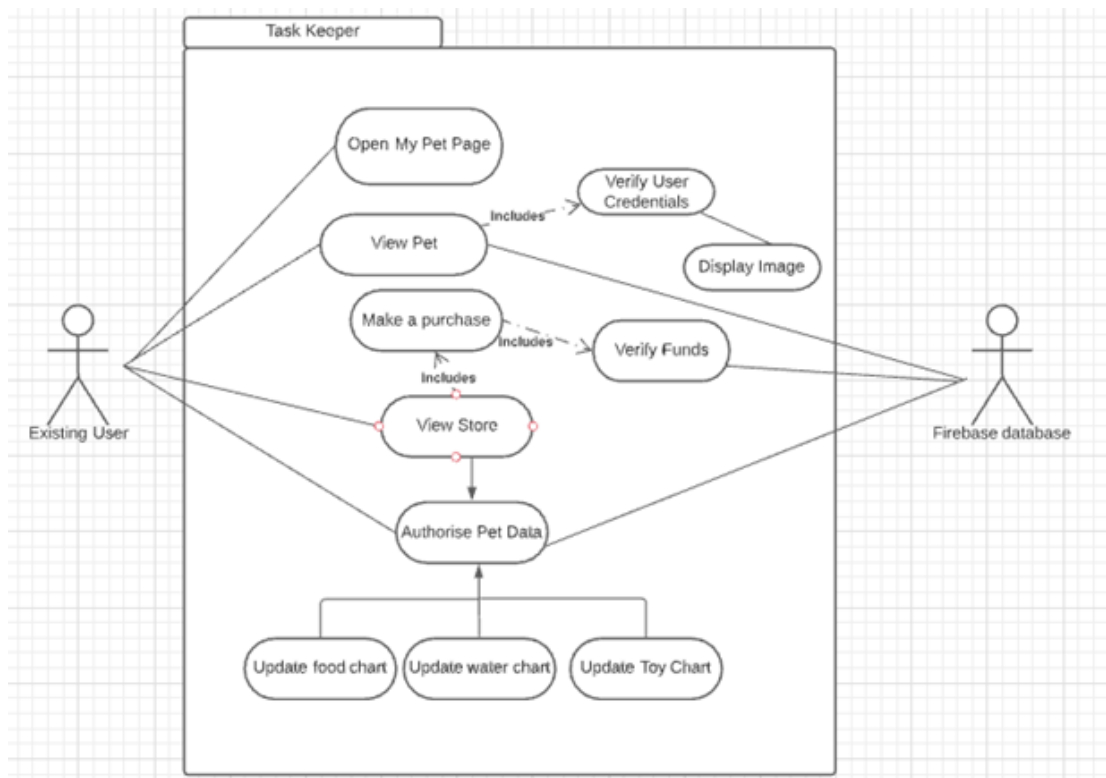
Figure 19: Pet Maintenance Use-Case

This case diagram summarizes the process the user initiates when using the pet maintenance page. This is the gamified section of the application. The two actors' existing user and firebase database operate the system. The user can open the my pet page, view the pet and balance. The firebase database would then authorize the data of the user's pet and the chart would be updated. This includes the food, water and toy chart of the pet. This case diagram captures the core functionalities of the Task Keeper system.

Creating a To-Do List:
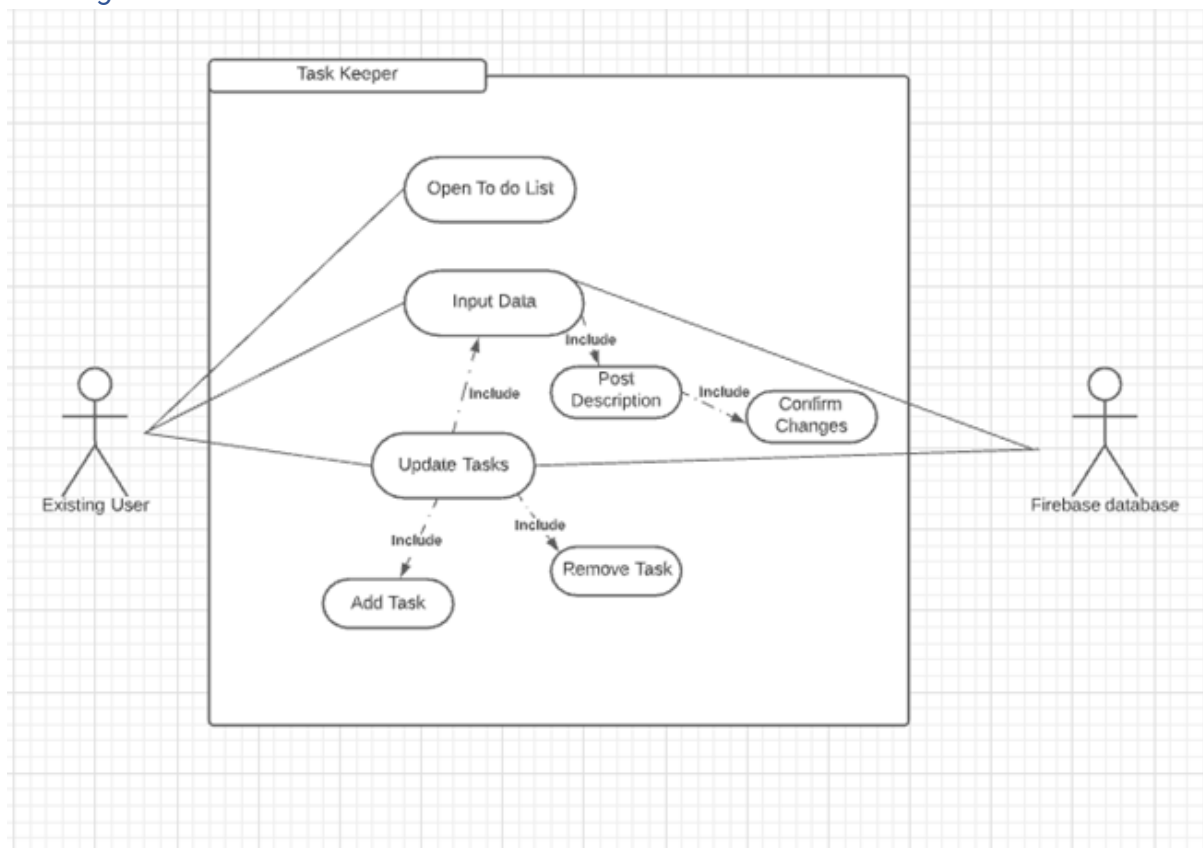
Figure 20: To-Do List Use-Case

This Use Case Diagram represents the action, the user completes when using the to-do list. The two actors, the existing user and the firebase database operate the system throughout. The existing user can open the to-do list, add, and remove tasks on the application. Finally, the user inputs the data and then it would verify the user's credentials with the database.

Appendix C - Gantt Chart:

| Tasks	January				February				March				April				May	
	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2
The application must have an accompanying website outlining what the application is and to showcase more information about the project team	■	■	■	■														
Website must be designed to be responsive		■	■	■														
Milestone 1: Completion of all the website's requirements				■														
Deliverable 1: A functional release of the website using GitHub pages as a host				■														
The application must have a friendly User Interface (UI)					■	■												
Users must be able to create, modify and delete a to-do list					■	■												
Users must be able to have multiple to-do lists simultaneously						■												
Users must be able to earn in-game money when they mark a task as complete						■	■											
Users must be able to spend their earned money to purchase pet care items for the game via the in-app store							■	■										
Users must receive a daily notification to remind them of their to-do lists							■	■										
Milestone 2: Completion of some of the app's '**must**-have' requirements								■										
Deliverable 2: A pre-alpha stage of implementation of the project idea through a functional application that can run via an emulator								■										
Users must be able to modify their pet's happiness, hunger and thirst levels using pet care items purchased from the in-app store							■	■										
Users must receive in-app alerts when they run out of in-game pet care resources									■	■								
The application must automatically reduce the pet's hunger, thirst, and happiness levels daily									■	■								
All user and pet data must be stored in an offline database										■	■							
The offline database must be instantly updated whenever any changes occur											■	■						

Task																									
The application must be able to run on Android devices											█	█												
Testing 1: Begin creation of Test Plan										█	█	█												
Milestone 3: Completion of all the app's '**must**-have' requirements												█												
Deliverable 3: An alpha stage implementation of the project idea through a functional application that can run via both an emulator and an actual android device (using APK)												█												
Users should lose in-game money when they uncheck a previously completed task											█	█												
Users should be able to register and create an account												█	█											
Users should sign in using their account credentials to get access to the application												█	█	█										
All user account credentials must be stored in remote cloud storage to reduce risks of account security compromises													█	█										
The application should display changes to the pet's hunger, thirst and happiness levels in real-time														█	█									
Testing 2: Finalise Test Plan and begin testing phase											█	█	█											
Milestone 4: Completion of some of the app's '**should**-have' requirements														█										
Deliverable 4: A beta stage implementation of the idea through a functional application that can run via both an emulator and an actual android device (using APK)														█										
Review could have requirements and decide on any additional feature implementations														█	█	█								
Finalise project report															█	█								
Complete final testing phase														█	█	█								
Milestone 5: Completion of all the project's requirements															█									
Deliverable 4: A stable release version of the application and website accompanied by a detailed project report															█									

Appendix D - Risk Analysis and Mitigation Plan:

Risk Number	Description of Risk	Probability	Impact	Mitigation Plan
1	The project's requirements and scope are unclear or unrealistic	2	5	All requirements will be reviewed by team members and thoroughly discussed before being finalised. The team will also actively seek out feedback from experts (e.g., lecturers and industry professionals) when possible to ensure that this project's scope is attainable.
2	Insufficient knowledge and background research on mobile applications	3	4	The team will carry out an intensive research process before beginning the development phase to ensure that all members are well versed in the technical area.
3	Team member falls ill due to ongoing pandemic or is otherwise unable to support the team due to extenuating circumstances	3	4	The team will review the situation and decide on who is most capable and available at that moment to be able to temporarily take over from the team's missing member.
4	Security breach due to user accounts being compromised	2	5	Ensure that all user account credentials are stored in a secure remote cloud database.
5	Data is lost due to a technical failure	2	4	The team will make regular commits to the GitHub repository and will make use of other cloud storage options to ensure backups are created and maintained.
6	Project tasks go over the allotted time	2	3	All project tasks will be given a buffer for overrun time throughout the project. The team will also aim to complete the work a week before the actual deadline.
7	A major bug is discovered during the testing stage	2	2	The team's chosen methodology is Agile development; thus, this ensures the team carries out frequent tests during development which prevents large scale bugs from appearing in the late stages.

8	Team member struggles to engage with the team or is not actively communicating with the rest of the team	3	4	The team will have frequent check-ins to ensure how all team members are handling their workload and if anyone requires assistance with managing their tasks. All members are encouraged to seek help from the rest of the team.
9	Member experiences issues with handling the workload	3	4	The team will meet and review the assigned task to break it down amongst other members to help support the struggling member.

Probability	Impact	Description
5	5	If this risk were to happen it will **seriously** impact the project's progress
4	4	If this risk were to happen it will **significantly** impact the project's progress
3	3	If this risk were to happen it will **moderately** impact the project's progress
2	2	If this risk were to happen it will have a **minor** impact on the project's progress
1	1	If this risk were to happen it will have **little to no** impact on the project's progress

Appendix E - UI and Writeup

For this project, the team aims to have an intuitive and friendly user interface. This will allow users not familiar with the system to use it with ease. To achieve this goal, the team will be following several principles, listed below. These principles will be expanded upon below.

- Clear and readable interface
- Tasks should be streamlined and completed in as few steps as possible - following the "3 click rule"
- The UI should remain consistent throughout the program
- Help should be available if the user needs assistance with using the program

Firstly, the interface should be clear and easy to read and understand. There should be no conflicting colours that make text hard to read, nor any complex and long titles. This will mean the user will not be startled by poor design choices and will clearly understand the UI with no prior training or additional help. Tasks should be streamlined, and the number of tasks and buttons should be minimised to not confuse the user with excessive choices, with each page having a specific focus. The three-click rule should also apply here, meaning users should find the information they want within three-click or taps, otherwise the UI may be too complex. Furthermore, the UI should have continuity throughout the application, with the same font, text size and colours being used throughout. This gives the user a sense of familiarity and consistency with the UI and program.

Finally, the users of the program should have help available to them if they need it. This would likely be in the form of a menu option and would help the users who do not understand the UI.
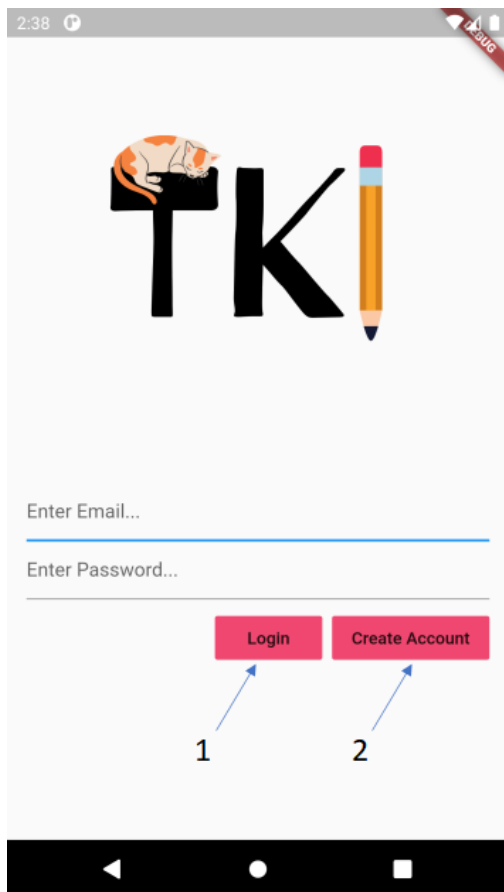
Figure 21: UI Sample 1
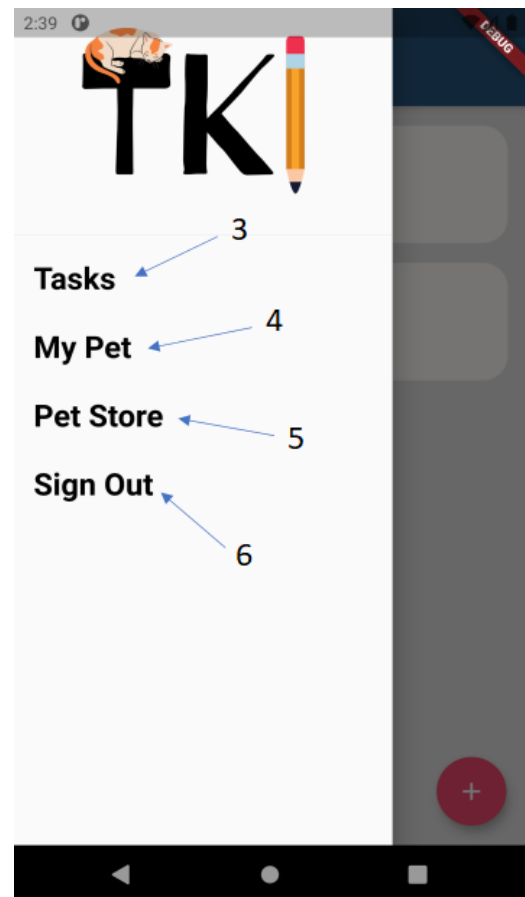
Figure 22: UI Sample 2
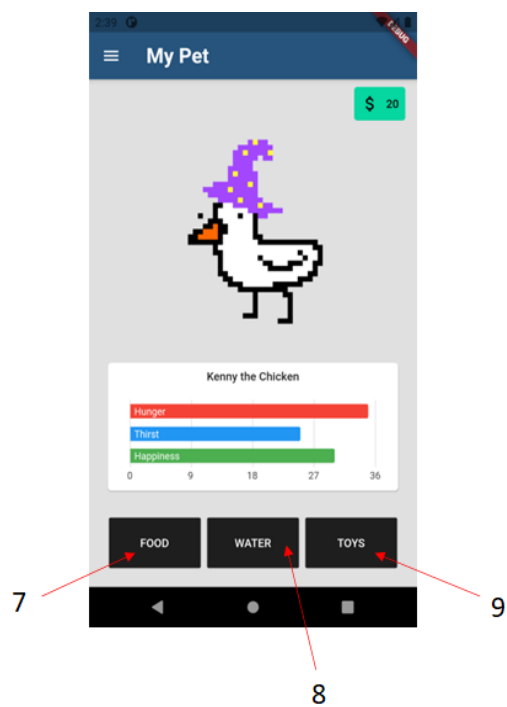
Figure 23: UI Sample 3

Figure 24: UI Sample 4

1.
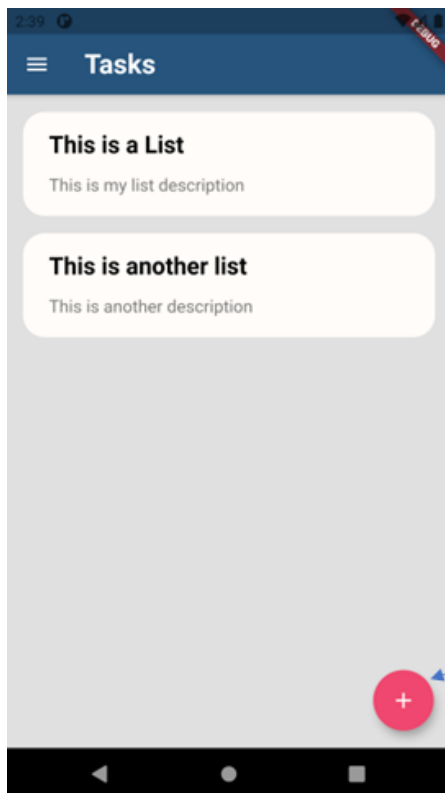
Figure 25: UI Sample 5
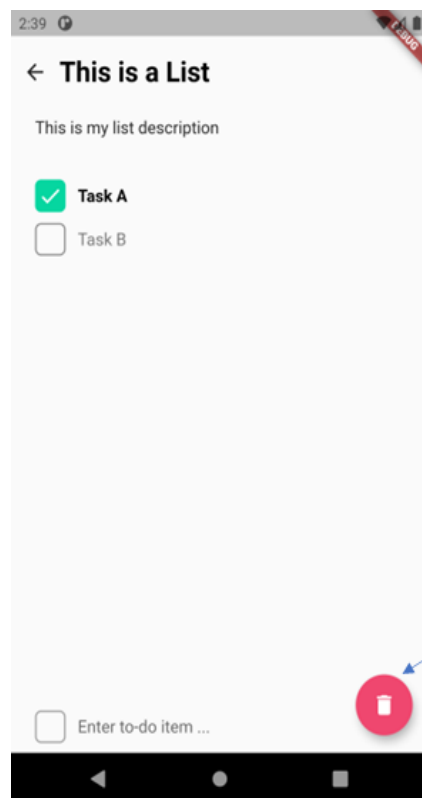
Figure 26: UI Sample 6

Number	Explanation
1	The 'Create an account' button allows the user to create an account after entering their details.
2	The 'login' button allows the user to access their account after their details have been entered.
3	When the 'Home' button is clicked, it then navigates the user to the home page.
4	When the 'Tasks' button is clicked, it navigates the user to the Task page.
5	When the 'My Pet' button is clicked, it navigates the user to that page
6	When the 'Pet Store' button is clicked, it navigates the user to that page
7	The 'Food' button modifies the chart when the user clicks the button.
8	The 'Water' button modifies the chart when the user clicks the button.
9	The 'Toys' button modifies the chart when the user clicks the button.
10	The 'Wake Up' button, wakes up the pet and consequently, a statistics bar chart appears.
11	This button when clicked triggers an action where a task can be added.
12	This button when clicked triggers an action where a task can be removed.

Appendix F - Meeting Breakdown and Notes

Number	Attendance					Date
	Ben	Ethan	Hannah	Owen	Simon	
1	A	A	A	A	UA	27/11/2020
2	A	A	A	A	UA	30/11/2020
3	A	A	A	A	UA	03/12/2020
4	A	A	A	A	UA	07/12/2020
5	A	A	A	A	UA	15/12/2020
6	A	A	A	A	UA	22/12/2020
7	A	A	A	A	UA	04/01/2021
8	A	A	A	A	A	12/01/2021
9	A	A	A	A	A	26/01/2021
10	A	A	A	A	A	02/02/2021
11	A	A	A	A	A	09/02/2021
12	A	A	A	A	UA	16/02/2021
13	A	A	A	A	A	19/02/2021
14	A	A	A	A	A	23/02/2021
15	A	A	A	A	A	26/02/2021
16	A	A	A	A	UA	02/03/2021
17	A	A	A	A	A	05/03/2021
18	A	A	A	A	A	09/03/2021
19	A	A	A	UA	A	12/03/2021
20	A	A	A	A	A	19/03/2021
21	UA	A	A	A	A	23/03/2021
22	Yes	A	A	A	A	26/03/2021
23	Yes	A	A	A	A	30/03/2021
24	Yes	A	A	A	A	02/04/2021
25	UA	A	A	AA	A	07/04/2021
26	A	A	A	AA	A	09/04/2021
27	A	A	A	AA	UA	12/04/2021
28	A	A	A	A	A	14/04/2021
29	A	A	A	UA	A	16/04/2021
30	A	A	A	A	A	19/04/2021
31	A	A	AA	A	UA	21/04/2021
32	A	A	A	A	A	23/04/2021

AA = Authorised Absence
UA = Unauthorised Absence
A = Attended

Figure 27: Meeting Attendance Log
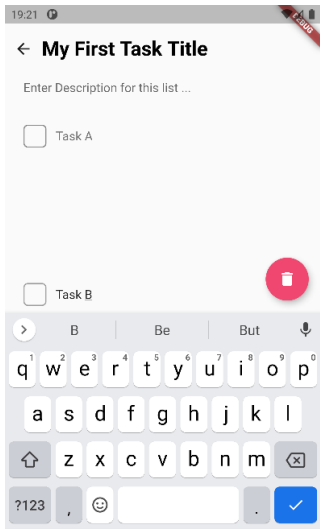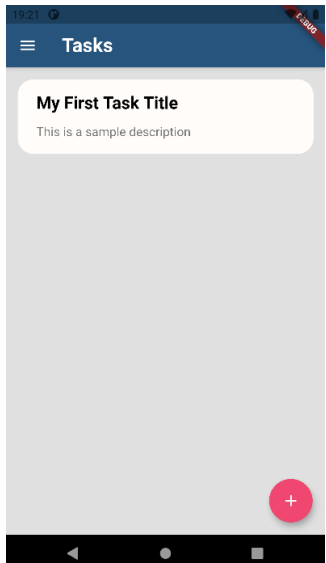
1	The initial meeting where the team spoke about methods of communication, any initial ideas and which degree each member was studying.
2	Meeting to get to know who was in the group, discussed any potential ideas any of the members may have had.
3	In this meeting, the team decided on the project idea that was going to be pursued and came up with ways of reaching the potential clients, i.e., surveys.
4	Distributed tasks, worked on creating a meeting schedule i.e., Monday or Tuesday at 4 PM.
5	In this meeting, the team decided on the deputy and overall project manager, project methodology, checked everyone was okay with their workload and went through the questions members of the team had.
6	Checked through work that had been completed so far, attempted to reach Simon and talked about ways the team should tackle the lack of communication from him, i.e. a 3 point system.
7	A quick check on what work had been completed so far tackled some questions that were asked about the Project Definition Document.
8	Formatted the final project definition document that was due to be submitted, the team were finally able to get through to Simon.
9	Distributed first 3 pieces for the report, intro, survey of current products and new ideas to Ethan, Ben and Simon. The project manager also allocated the task of designing and developing a web page to Hannah and Owen as they said they were happy to start it now.
10	In this meeting, the team just wanted to make sure everyone knew what they were doing. It was a general check on whether the work was progressing - and it was, very smoothly.
11	Had a look through the work the team had done so far. Looked into Dart as the project's language of choice for developing Task Keeper.
12	Discussed the possibility of more than 1 meeting a week to increase workflow, discussed the quality of work, and ensured that everything was progressing in line with expectations. Discussed the need to keep Simon in line to ensure that the work gets completed on time and to a good standard of quality.
13	Uploaded the first 3 sections of the report (intro - Ethan, survey - Ben and new ideas -Simon) to the shared Google Docs, assigned new work (UI writeup - Ben and Simon, legal issues - Ethan, Page linking structure diagram for website - Owen and
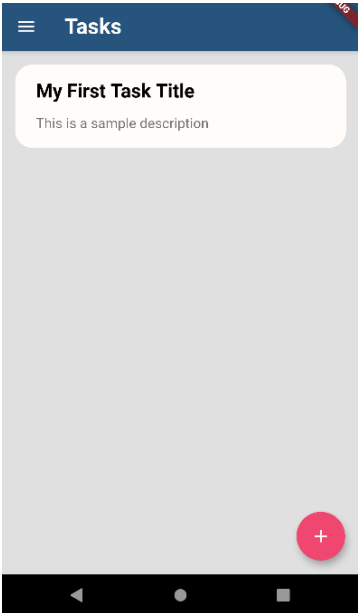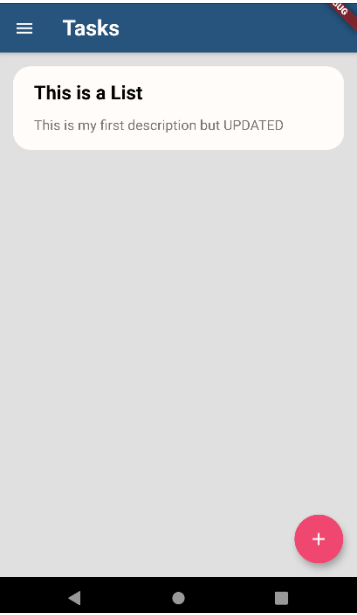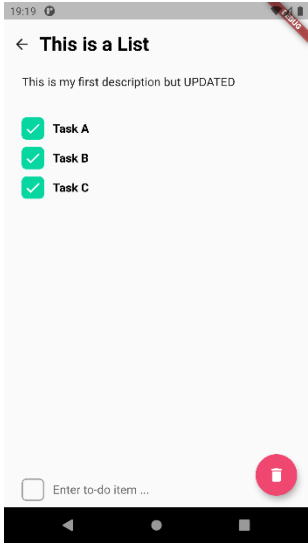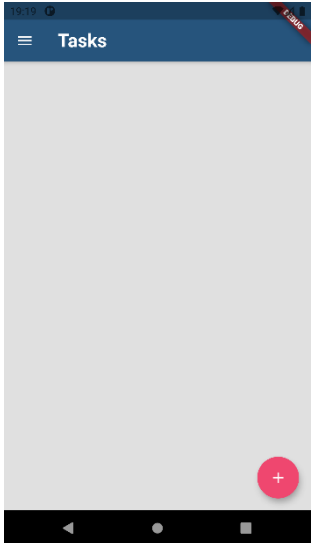
	development - Hannah).
14	A general check-up on the work that was set last week, Hannah showed a demo of what she had completed for the development so far.
15	Went through what has been completed in development and had a general check-in on work. Everyone seemed content with the work that had been set.
16	Discussed what the team were going to do for the upcoming interim review. Also discussed whether all milestones were going to be completed for Friday. Hannah went through what she has done so far in terms of development for the application.
17	Prepared the documentation, application, and web page for the interim review with James.
18	Had a quick check on progress to see how the project was progressing.
19	Checked on progress, went through what the team expected for Tuesday's meeting and had a general chat about what aims the team can realistically achieve by the end of the project.
20	Prepared contents and formatted work for the team's meeting with James.
21	Planned out what the team were going to do for the remaining sections (4,5,6).
22	Observed progress made on the app, distributed work (Owen doing testing, Simon, Ben and Ethan doing documentation and Hannah is continuing with the development. The team agreed to do more meetings over Easter (Mon-Wed - Fri) to ensure work was completed before the final meeting with James.
23	In this meeting, the team made sure everyone knew what they were doing. It was a general check on whether (and how well) the work was progressing.
24	Catch up, set work expected over next week, prepared for meetings over easter.
25	Hannah showed how the app has progressed since the team's last meeting, she added diminishing hunger, hydration and happiness into the app, so the longer the user is off the app, the more the food diminishes over time.
26	Went through everyone's progress with the work so far, talked about planning for the final presentation.
27	Talked through work completed for this week's tasks, ensured everybody knew what
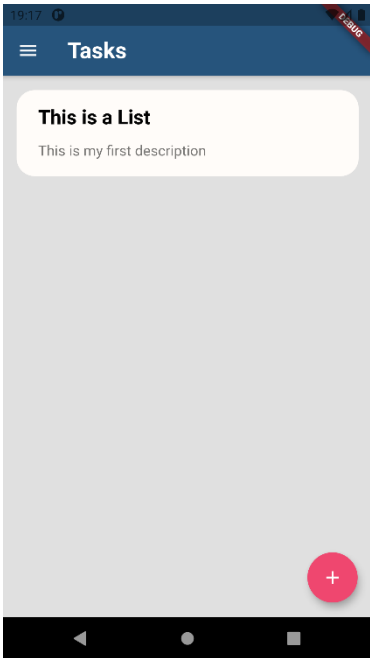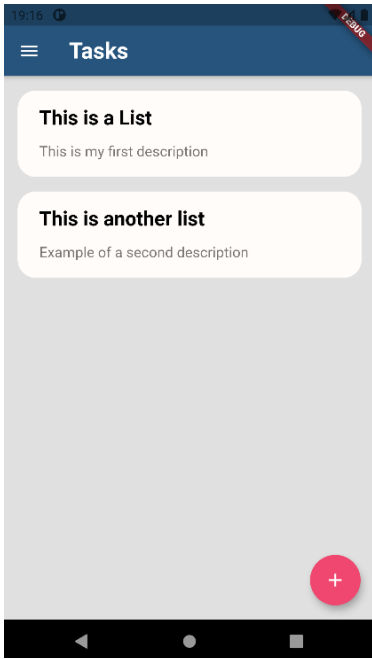
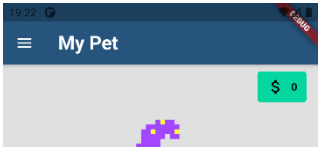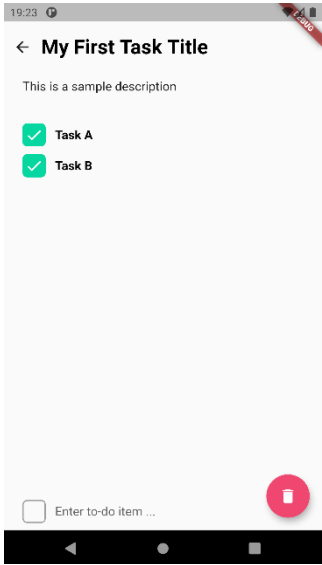|	
	they were doing and what needed to be done for Wednesday's meeting.
28	The final meetings of this month were used to format the final report document, paste in the remaining work and start planning out the video recording of the final demonstration. The team also considered starting to understand who will speak about what section when the team demonstrates it to the other members of the tutor group.
29	
30	
31	
32	

Appendix G - Test Report:
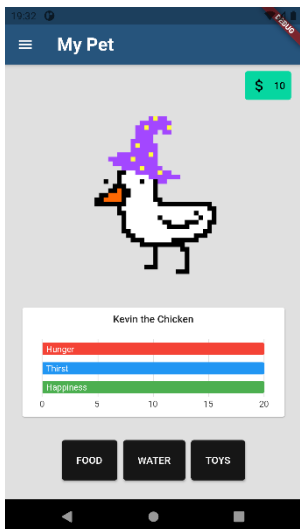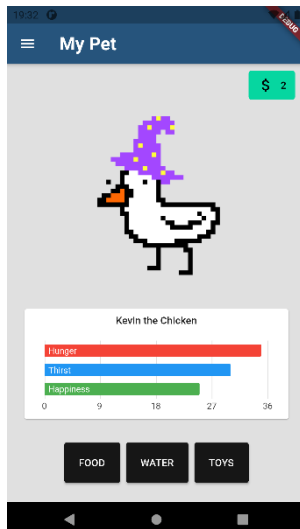
ID	01	Description	Users must be able to create a to-do list with a title, description, and task items when they click on the plus-icon button within the app in the Tasks tab.
Test Type	Qualitative	Success Criteria	User can create their to-do list successfully which includes the title, description, and task items.
List of equipment / requirements	Android Studio to execute the source code Emulator of an Android Device (Nexus 6 API 30)		
Setup Instructions	Start up the emulator and then run the application's source code. Once the application is running, log-in and then navigate to the Tasks tab. Then click on the button in the bottom-right corner with the plus icon on it. Proceed to enter values for the title, description, and task items. Once done, navigate out of this window by clicking the left facing arrow in the top-left of the screen.		
Failure Correction Procedure	Review the code to identify the source of the issue. Begin the debugging phase to resolve the problem.		
Engineer(s) / Technicians	Owen		
Individual Results		*Figure 28: To-Do List Creation Screen (Left)* *Figure 29: Tasks Tab with new To-Do List (Right)*	
Test Date	21/05/21	Result	Passed

ID	02	Description	Users must be able to modify a to-do list by changing its title, description, and adding new task items when they click on the to-do list within the app in the Tasks tab.
Test Type	Qualitative	**Success Criteria**	Users can modify a to-do list by changing its title, description, and adding new task items when they click on the to-do list within the app in the Tasks tab.
List of equipment / requirements	Android Studio to execute the source code Emulator of an Android Device (Nexus 6 API 30)		
Setup Instructions	Start up the emulator and then run the application's source code. Once the application is running, log in and then navigate to the Tasks tab. Then click on the chosen to-do list. Update the title and description text and add two more task items to the to-do list. Once done, navigate out of this window by clicking the left-facing arrow in the top-left of the screen and note the updated title and description displayed within the Tasks tab.		
Failure Correction Procedure	Review the code to identify the source of the issue. Begin the debugging phase to resolve the problem.		
Engineer(s) / Technicians	Owen		
Individual Results	*Figure 30: To-Do List Before Update (Left)* *Figure 31: To-Do List After Update (Right)*		
Test Date	22/05/21	**Result**	Passed
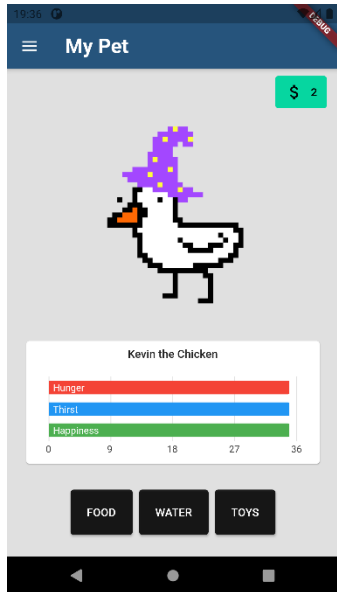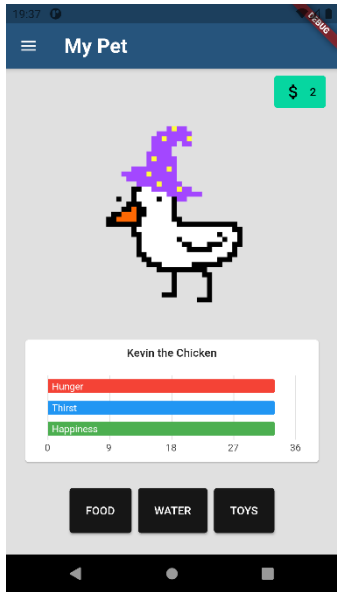
ID	03	**Description**	Users must be able to delete a to-do list from their Tasks tab.
Test Type	Qualitative	**Success Criteria**	User can delete a to-do list from their Tasks tab.
List of equipment / requirements	Android Studio to execute the source code Emulator of an Android Device (Nexus 6 API 30)		
Setup Instructions	Start up the emulator and then run the application's source code. Once the application is running, log in and then navigate to the Tasks tab. Then click on the chosen to-do list. Select the trash-can icon in the bottom-left corner. Once done, the app will automatically navigate out of this window. The Tasks tab will be updated to reflect the changes made.		
Failure Correction Procedure	Review the code to identify the source of the issue. Begin the debugging phase to resolve the problem.		
Engineer(s) / Technicians	Simon		
Individual Results	*Figure 32: To-Do List Page with Delete Button in bottom corner (Left)* *Figure 33: Updated Tasks Tab with To-Do List Deleted (Right)*		
Test Date	24/05/21	**Result**	Passed
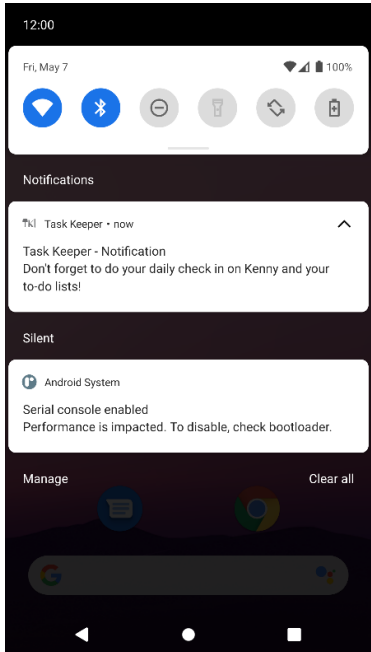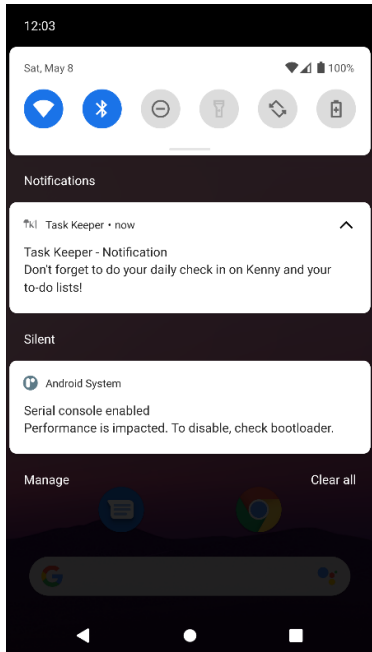
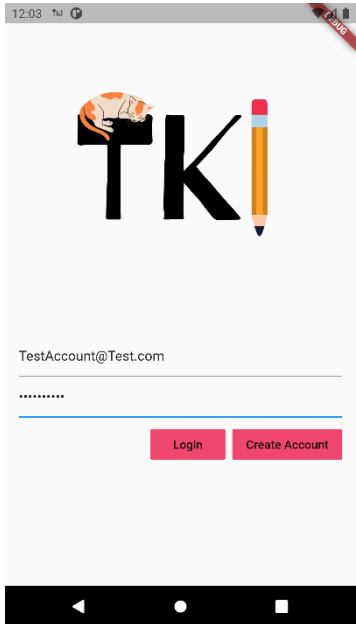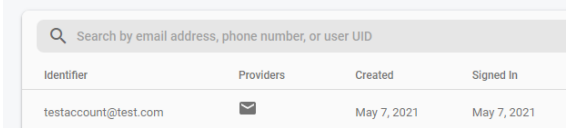| ID	04	Description	Users must be able to make multiple to-do lists and have them displayed within their Tasks tab.
Test Type	Qualitative	Success Criteria	Users can make multiple to-do lists and have them displayed within their Tasks tab.
List of equipment / requirements	Android Studio to execute the source code Emulator of an Android Device (Nexus 6 API 30)		
Setup Instructions	Start up the emulator and then run the application's source code. Once the application is running, log in and then navigate to the Tasks tab. Then click on the button in the bottom-right corner with the plus icon on it. Proceed to enter values for the title, description, and task items. Once done, navigate out of this window by clicking the left-facing arrow in the top-left of the screen. Repeat this process until two or more to-do lists have been created. These separate lists should be displayed within the Tasks tab.		
Failure Correction Procedure	Review the code to identify the source of the issue. Begin the debugging phase to resolve the problem.		
Engineer(s) / Technicians	Owen		
Individual Results	*Figure 34: Tasks Tab with one To-Do List (Left)* *Figure 35: Tasks Tab with multiple To-Do Lists (Right)*		
Test Date	24/05/21	Result	Passed

ID	05	Description	Users must receive in-game currency when they mark a task as complete
Test Type	Qualitative	**Success Criteria**	Users receive in-game currency when they mark a task as complete
List of equipment / requirements	Android Studio to execute the source code Emulator of an Android Device (Nexus 6 API 30)		
Setup Instructions	Start up the emulator and then run the application's source code. Once the application is running, log in and then navigate to the My Pet Tab first. Make note of the user's money value in the top-right corner of the screen. Proceed to the Tasks tab. Then click on the chosen to-do list. Mark two task items as complete. Once done, navigate out of this window by clicking the left-facing arrow in the top-left of the screen. Navigate to the My Pet tab and check the user's money value again to note the change.		
Failure Correction Procedure	Review the code to identify the source of the issue. Begin the debugging phase to resolve the problem.		
Engineer(s) / Technicians	Owen		
Individual Results	 *Figure 36: User's Funds Before Task Completion (Left)* *Figure 37: User's Funds After Task Completion (Left)* *Figure 38: Tasks being marked as complete (Right)* 		
Test Date	25/04/21	**Result**	Passed

ID	06	**Description**	Users must be able to increase their pet's happiness, hunger, and thirst level by purchasing pet care items via the in-app store.
Test Type	Qualitative	**Success Criteria**	Users can increase their pet's happiness, hunger, and thirst level by purchasing pet care items via the in-app store.
List of equipment / requirements	Android Studio to execute the source code Emulator of an Android Device (Nexus 6 API 30)		
Setup Instructions	Start up the emulator and then run the application's source code. Once the application is running, log in and then navigate to the My Pet Tab first. Make note of the pet's hunger, thirst, and happiness values. Proceed to the Pet Store tab. Then click on the food, water, and toys buttons to purchase pet care items. Once done, navigate out of this window by clicking the hamburger menu in the top-left of the screen. Navigate to the My Pet tab and check the user's money value again to note the change.		
Failure Correction Procedure	Review the code to identify the source of the issue. Begin the debugging phase to resolve the problem.		
Engineer(s) / Technicians	Ben		
Individual Results	*Figure 39: Pet Stat Values Before Increase (Left)* *Figure 40: Pet Stat Values After Increase (Right)*		
Test Date	25/04/21	**Result**	Passed

ID	07	Description	Pet's happiness, hunger, and thirst levels must decrease daily.
Test Type	Qualitative	**Success Criteria**	Pet's happiness, hunger, and thirst levels are decreased daily.
List of equipment / requirements	Android Studio to execute the source code Emulator of an Android Device (Nexus 6 API 30)		
Setup Instructions	Start up the emulator and then run the application's source code. Once the application is running, log in and then navigate to the My Pet Tab first. Make note of the pet's hunger, thirst, and happiness values before closing the app. Proceed to change the device's date/time to the following day. Re-open the app and note the changes made to these values.		
Failure Correction Procedure	Review the code to identify the source of the issue. Begin the debugging phase to resolve the problem.		
Engineer(s) / Technicians	Simon		
Individual Results	*Figure 41: Pet Stat Before Daily Decrease (Left)* *Figure 42: Pet Stat After Daily Decrease (Right)*		
Test Date	28/04/21	**Result**	Passed
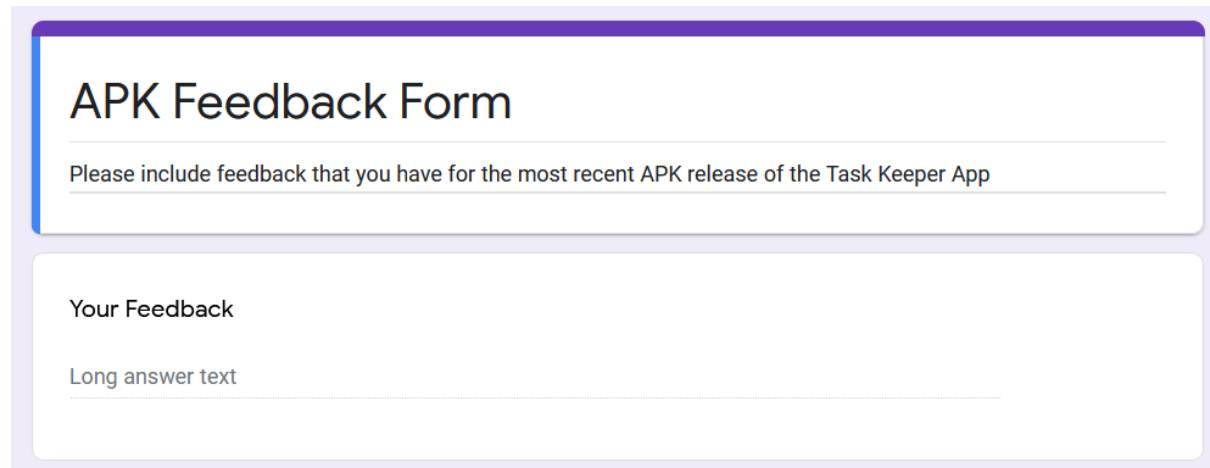
ID	08	Description	Users must be notified daily to check the application and complete their pending tasks.
Test Type	Qualitative	**Success Criteria**	Users are notified daily to check the application and complete their pending tasks.
List of equipment / requirements	An android device APK Version of the Application		
Setup Instructions	Install the APK version onto an android mobile device. Upon completion of installation, open and run the application. Log in to an account and check back in daily at 12 pm for a notification from the database.		
Failure Correction Procedure	Review the code to identify the source of the issue. Begin the debugging phase to resolve the problem. In the case the issue is not because of the code, review the notification settings within the Firebase console instead.		
Engineer(s) / Technicians	Hannah		
Individual Results	*Figure 43: Notification from 7th May - 12pm (Left)* *Figure 44: Notification from 8th May - 12pm (Right)*		
Test Date	08/05/21	**Result**	Passed

ID	09	Description	User must be able to create an account using the sign-up feature within the application. Their credentials must be stored within the Firebase console
Test Type	Qualitative	**Success Criteria**	User can create an account using the sign-up feature within the application. Their credentials are stored within the Firebase console
List of equipment / requirements	Android Studio to execute the source code Emulator of an Android Device (Nexus 6 API 30)		
Setup Instructions	Start up the application and proceed to input account credentials into the set text boxes. Click on the create account button and the application should automatically log the user into the application. Review the firebase console's credentials database to see the new account added to the list.		
Failure Correction Procedure	Review the code to identify the source of the issue. Begin the debugging phase to resolve the problem. In the case the issue is not because of the code, review the credentials storage settings within the Firebase console instead.		
Engineer(s) / Technicians	Hannah		
Individual Results	*Figure 45: Application's Credentials Creation/Login Screen (Left)* *Figure 46: Firebase Credentials Console (Above)*		
Test Date	07/05/21	**Result**	Passed

Appendix H - Feedback Survey:

The form was used to obtain feedback from the external testing team comprised of Developer's Society members (NTU DevSoc, n.d.).

Link: https://forms.gle/9NjSrSc6LZDemC6a6

Figure 47: Feedback Form Template

Figure 48: Example of Survey Responses