



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 made the decision to research 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. With this in mind, the team opted for a solution whereby the user was given the responsibility of taking care of their own 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 on a daily basis. 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, 2021), (Habit Hunter, 2021) and (EpicWin, 2021) that collectively, have over 1.1 million global downloads on the (Google Play Store, 2021).

To test viability, in regards to Task Keeper's distributability, Hannah provided an Android Application Package (APK) version of the app to members of NTU's DevSoc (The Developer's Society, n.d.). Following this testing, Hannah received the feedback via DevSoc's dedicated Discord server. The general consensus was that Task Keeper was suitable for distribution, predominantly due to its ease of use and user-friendly UI. - will be refactored when Hannah distributes a survey to go alongside the APK

Table of Contents

| | |
|---|----|
| Abstract: | 2 |
| Table of Figures: | 5 |
| Section 1- Introduction: | 6 |
| Aim: | 7 |
| Objectives: | 7 |
| Requirement Capturing Method: | 7 |
| Functional Requirements: | 7 |
| Non-Functional Requirements: | 8 |
| Section 2 - Survey of Existing Solutions: | 9 |
| Habitica: | 9 |
| Habit Hunter: | 10 |
| EpicWin: | 11 |
| Demand for Product: | 11 |
| Section 3 – New Ideas: | 12 |
| Section 4 – Design and Development: | 13 |
| Development Phase Gantt Chart: | 13 |
| Risk Analysis and Mitigation Plan: | 14 |
| Project Management: | 16 |
| Team Roles: | 16 |
| Communication: | 17 |
| Task Assignment: | 18 |
| Workflow: | 19 |
| Documentation: | 20 |
| Flowchart Development Plan: | 21 |
| Login and Account Creation Functionality: | 21 |
| To-Do List Functionality: | 22 |
| Pet Maintenance Functionality: | 23 |
| Entity Relationship Diagram (ERD): | 24 |
| To-Do List Database Class Diagram: | 24 |
| Code Design: | 26 |
| Background Task Execution: | 26 |
| Deprecated Functions: | 26 |

| | |
|--|----|
| Dynamic Pet Statistics:..... | 27 |
| Section 5 - Evaluation of Product: | 29 |
| Objectives: | 29 |
| Primary Requirements: | 30 |
| Other Requirements:..... | 31 |
| Section 6 - Discussion and Conclusion:..... | 32 |
| Section 7 - Professional, Social, Ethical and Legal issues:..... | 34 |
| Introduction: | 34 |
| Professional Issues: | 34 |
| Social Issues: | 35 |
| Ethical issues: | 35 |
| Legal issues:..... | 35 |
| Appendices:..... | 36 |
| Appendix A - Survey | 36 |
| Appendix B - UI and Writeup..... | 40 |
| Appendix C - Meeting Breakdown and Notes | 44 |
| Appendix D - Risk Analysis and Mitigation Plan Weighting Table:..... | 48 |
| Appendix E – Test Report:..... | 49 |

Table of Figures:

| | |
|---|----|
| Figure 1: Screenshot of Habit Hunter..... | 10 |
| Figure 2: Screenshot of the Discord server's structure..... | 17 |
| Figure 3: Screenshot of checklist message..... | 17 |
| Figure 4: Screenshot of Task Assignment Example 1 | 18 |
| Figure 5: Screenshot of Task Assignment Example 2 | 18 |
| Figure 6: Screenshot of Task Assignment Example 3 | 18 |
| Figure 7: Screenshot of Task Assignment Example 4 | 18 |
| Figure 8: Screenshot of the Git Repository | 19 |
| Figure 9: Screenshot of the Git Commit History..... | 19 |
| Figure 10: Login and Account Creation Flowchart | 21 |
| Figure 11: To-Do List Flowchart..... | 22 |
| Figure 12: Pet Maintenance Flowchart | 23 |
| Figure 13: Entity Relationship Diagram | 24 |
| Figure 14: To-Do List Database Class Diagram..... | 25 |
| Figure 15: Screenshot of Git Commit Log..... | 26 |
| Figure 16: Code Block of Model Data Format | 27 |
| Figure 17: Screenshot of Pet Wake-up Screen | 28 |
| Figure 18: Screenshot of Pet Data..... | 28 |
| Figure 19: Survey Questions Part 1 | 37 |
| Figure 20: Survey Questions Part 2 | 38 |
| Figure 21: Survey Questions Part 3 | 39 |
| Figure 22: UI Sample 1 | 41 |
| Figure 23: UI Sample 2 | 41 |
| Figure 24: UI Sample 3 | 41 |
| Figure 25: UI Sample 4 | 41 |
| Figure 26: UI Sample 5 | 42 |
| Figure 27: UI Sample 6 | 42 |
| Figure 28: Meeting Attendance Log | 44 |

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 of time 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 et al., 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 upon productivity by making the process of task completion more engaging. Furthermore, (Bhanji and Delgado, 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 (refer to [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
- Application must automatically reduce the pet's hunger, thirst and happiness levels on a daily basis
- 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
- 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
- Application should display changes to 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
- Application could have a web version
- Application could have a leader board to compare user stats against
- Application could host all user data on a cloud-based database instead of an offline database

Non-Functional Requirements:**Operational:**

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

Performance:

- Offline database must be instantly updated whenever any changes occur

Accessibility:

- Application must have a friendly User Interface (UI)
- 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:

- Application could have multiple language options

Section 2 - Survey of Existing Solutions:

Habitica:

An existing application which is comparable to Task Keeper is an online task management game called Habitica (Habitica, 2021), as mentioned in the previous project definition document. Habitica is a similar application to Task Keeper, allowing users to gamify tasks in order 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. | 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. |
| Extensive system of levelling up, with boss fights to keep users entertained. Also has various classes eg warrior and mage allow for lots of content and replayability. | The character customization is limited, especially without the paid content. |

Task Keeper differs from Habitica in several ways, namely due to the fact that 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 leaderboard 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 eg washing the dishes, but the "want to" task, eg 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 challengers to the user as well as providing clear and structured goals with feedback. (Gamification: State of the Art, 2012)

In order to gauge customer interest, requirements and demand for a gamified task manager application, the team conducted an online survey. Within this survey, 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. This survey can be found in [Appendix A](#).

Habit Hunter:

Another similar application to Task Keeper is the game Habit Hunter (Habit Hunter, 2020). Habit Hunter is effectively a gamified task manager, much like the team's own application and Habitica, but 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 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 wider range of activities present better in a map visible to the player, as pictured below.



Figure 1: Screenshot of Habit Hunter

EpicWin:

The Final gamified task manager application that will be discussed is Epic Win (EpicWin, 2021). 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 is also of an RPG style and 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 to 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, 2021), (Habit Hunter, 2021) and (EpicWin, 2021) 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's 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 replayability, making it a useful tool for anyone trying to get work done on a consistent basis.

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 which 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 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 to cover all the necessary tasks, milestones and deliverables involved in the development process. Since the team adopted an agile approach for the project methodology, each sprint involves the completion of certain tasks which is then followed by both a milestone and deliverable which summarises the expected outcome of each sprint stage.

Risk Analysis and Mitigation Plan:

This table 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. A table containing a description of each probability and impact value is available within the appendix.

| Risk Number | Description of Risk | Probability | Impact | Mitigation Plan |
|-------------|---|-------------|--------|--|
| 1 | Project's requirements and scope are unclear or unrealistic | 2 | 5 | All requirements will be reviewed by team members and thoroughly discussed prior to 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 prior to beginning the development phase to ensure that all members are well versed with 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 allotted time | 2 | 3 | All project tasks will be given a buffer for overrun time throughout the duration of the project. The team will also aim to complete the work a week before the actual deadline. |

| | | | | |
|---|--|---|---|---|
| 7 | 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 work load and if anyone requires assistance with managing their own 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 in order to break it down amongst other members to help support the struggling member. |

Project Management:

Team Roles:

<https://www.belbin.com/about/belbin-team-roles>

Communication:

Due to the project timeline occurring amidst a global pandemic, the team had to ensure that constant online communication was maintained as we were not all able to meet in-person. The team's first approach was establishing a channel of communication. Upon discussion of our potential options which included Slack (cite) and Microsoft Teams (cite), the team settled on Discord. This is due to its popularity amongst our members in addition to being a platform that all our members frequently used for other projects or purposes. This meant that it would be much easier to contact another member if necessary and that one is more likely to get a quicker response. 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 the usage of the platform was not as high compared to Discord.

As showcased in Figure 2 it depicts how the team's leader, Hannah, had structured our 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 being made.

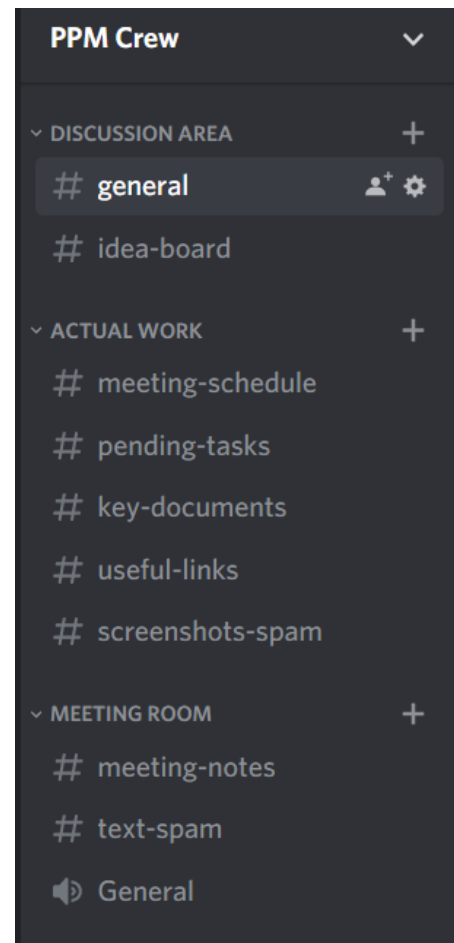


Figure 2: Screenshot of the Discord server's structure

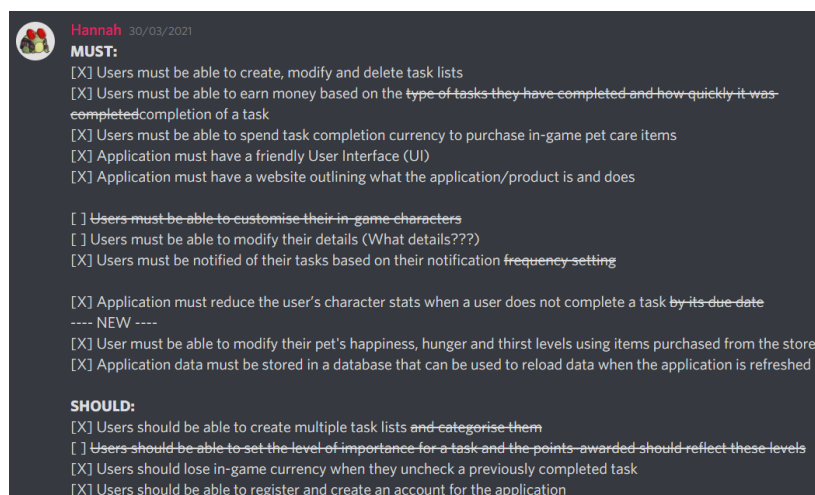


Figure 3: Screenshot of checklist message

Finally, this discord server is where the team would host our twice/thrice a week group calls in addition to our 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 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 meeting before moving onto what tasks need to be done next in order 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 particular member. All tasks are expected to be completed prior to 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 individual member. Included below are some screenshots of post-

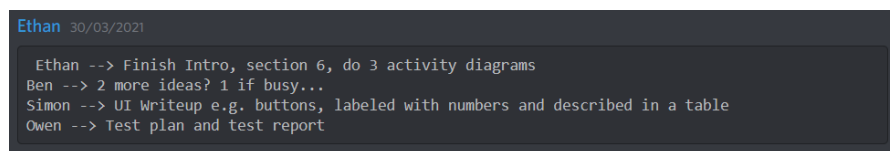
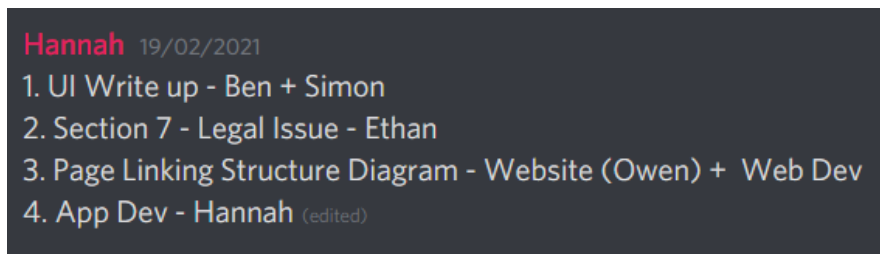


Figure 4: Screenshot of Task Assignment Example 1



meeting task assignment summaries.

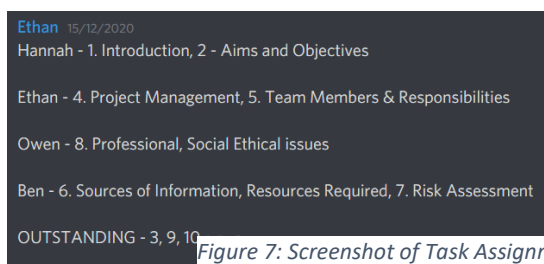


Figure 6: Screenshot of Task Assignment Example 3

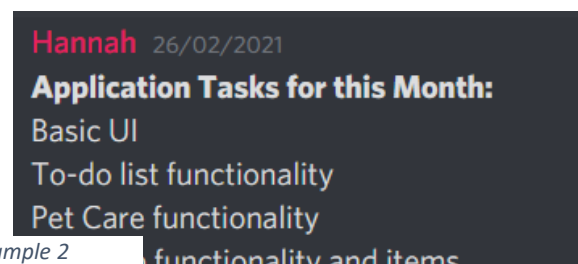


Figure 7: Screenshot of Task Assignment Example 4

Workflow:

A key tool in every programmer's toolbelt is the ability to use Git and GitHub. One group of researchers (citation) noted that it is an "essential part" of "social coding" especially for "decentralized team work" 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 still 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 8 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 9).

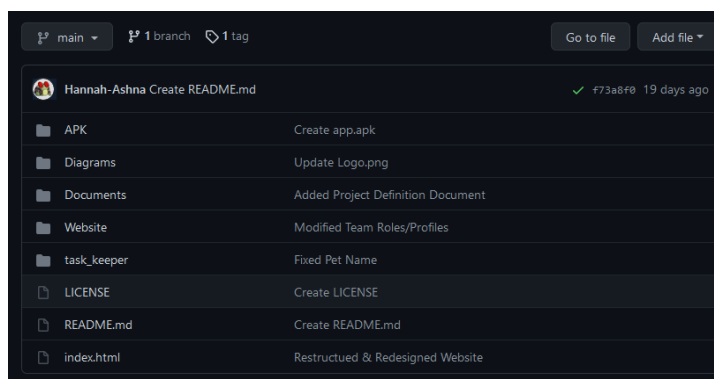


Figure 8: Screenshot of the Git Repository

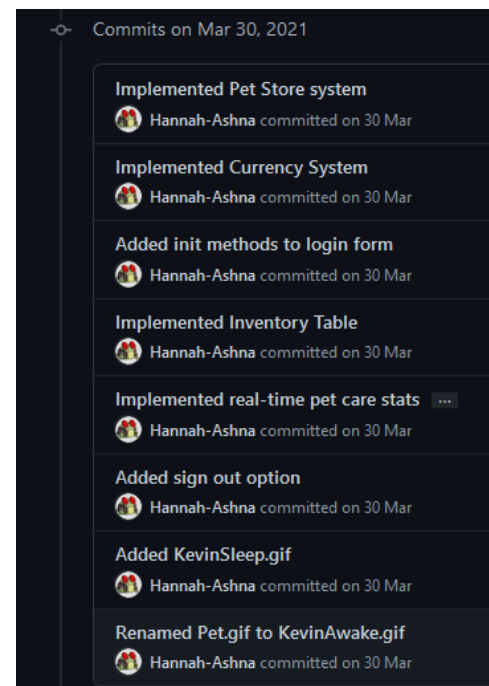


Figure 9: Screenshot of the Git Commit History

Documentation:

Flowchart Development Plan:

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:

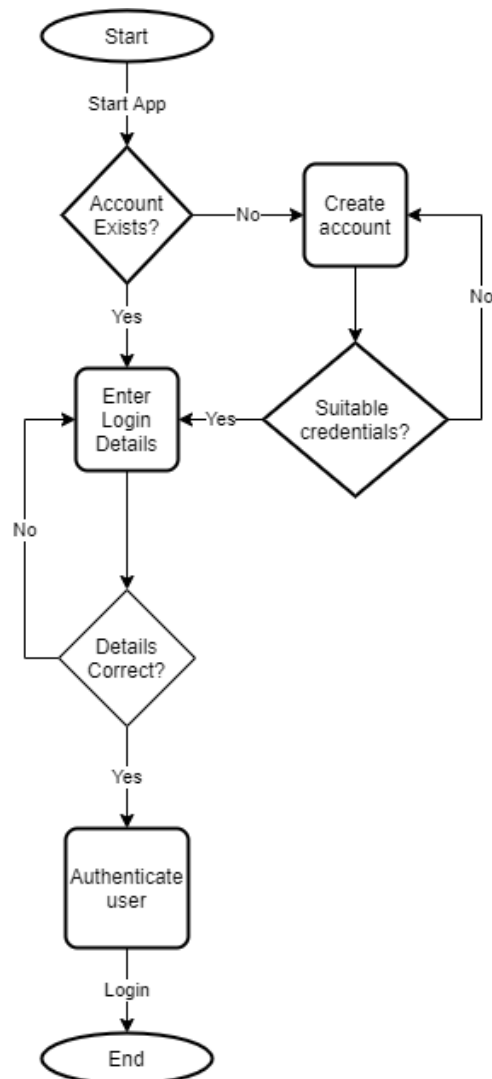


Figure 10: Login and Account Creation Flowchart

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:

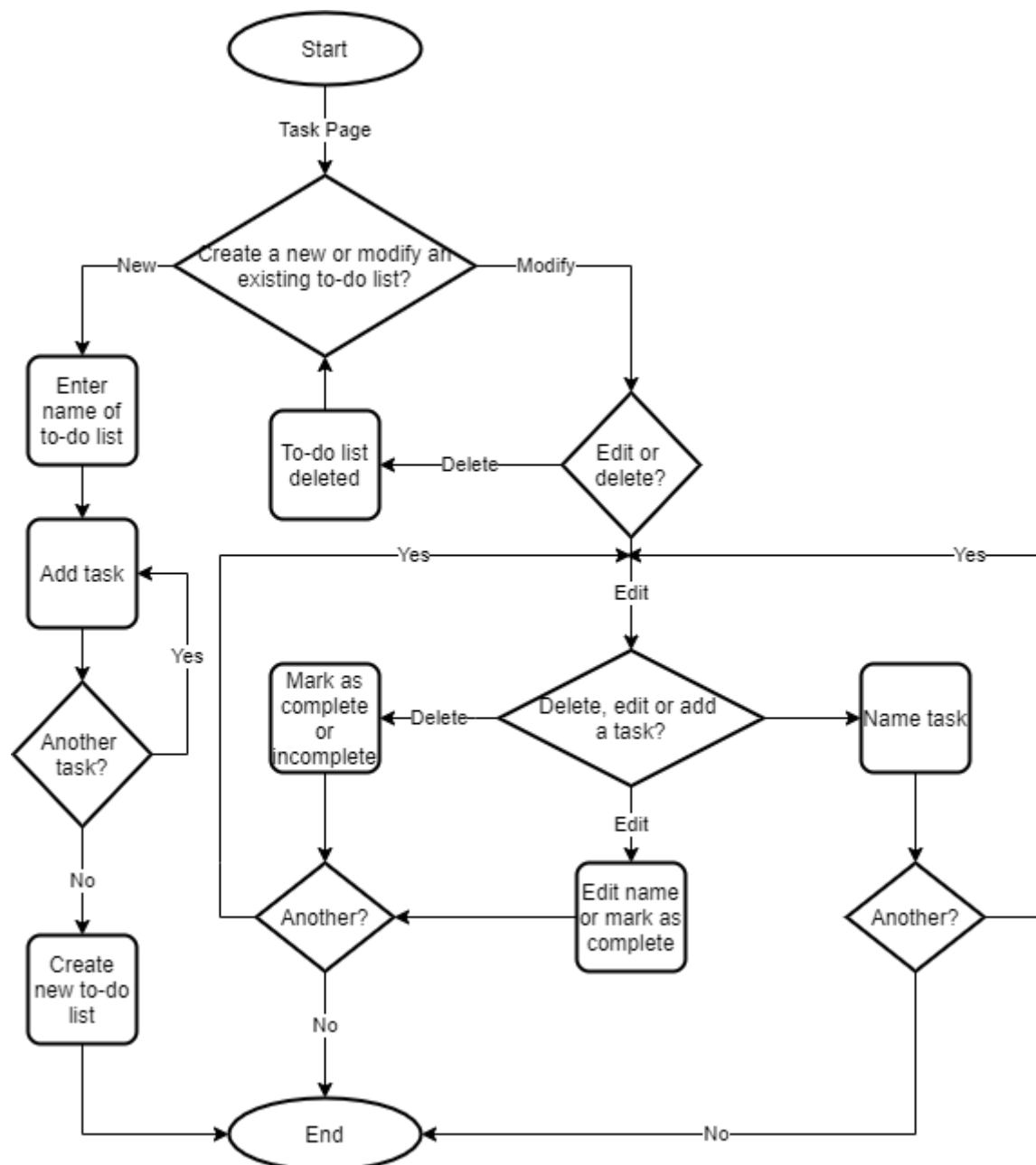


Figure 11: To-Do List Flowchart

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.

Pet Maintenance Functionality:

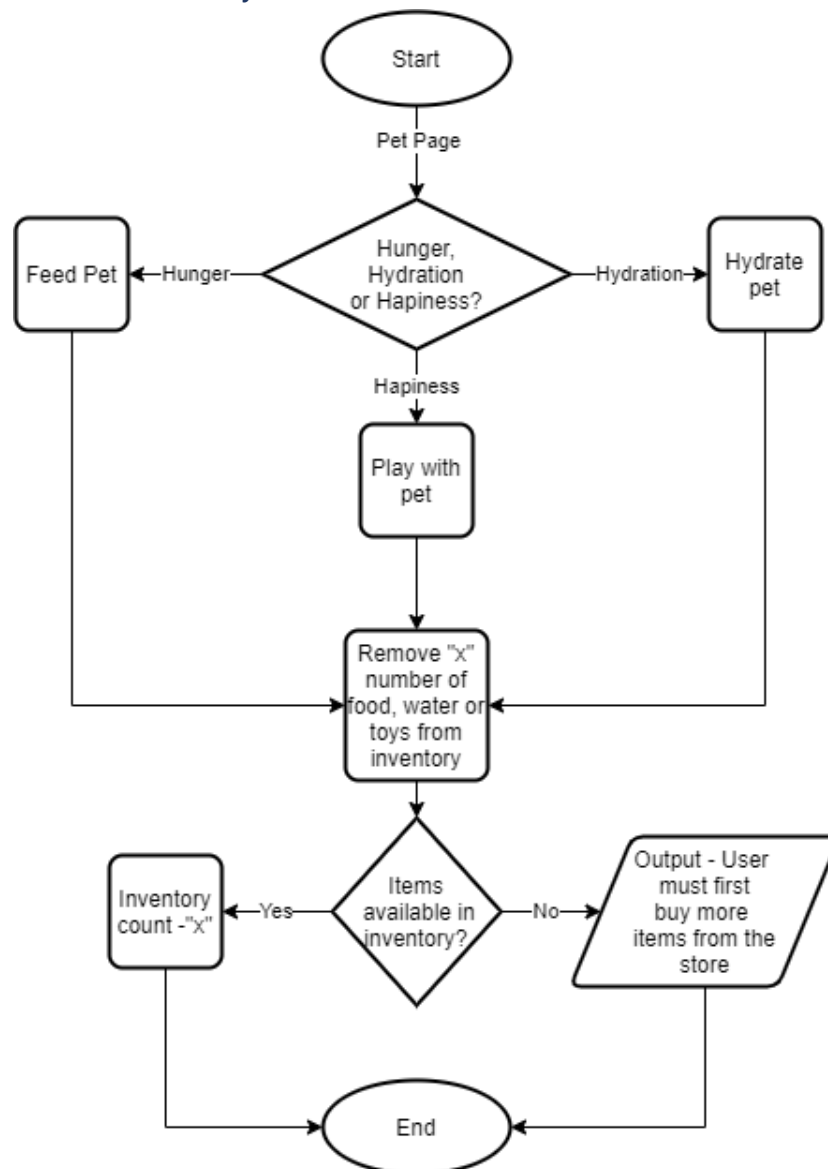


Figure 12: Pet Maintenance Flowchart

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):

In order to better visualise the relationships between the offline databases's entities, the team designed an Entity Relationship Diagram (ERD). Highlighted below 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 which 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 in relation to 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 particular 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 in relation to 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.

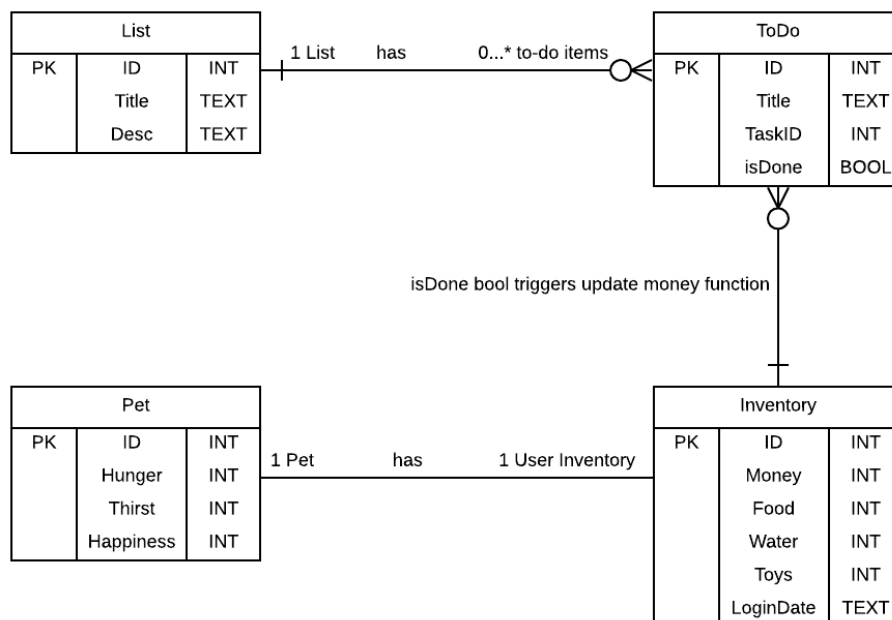


Figure 13: Entity Relationship Diagram

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.

The diagram below focuses on the use of the DatabaseTool class in relation to 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.

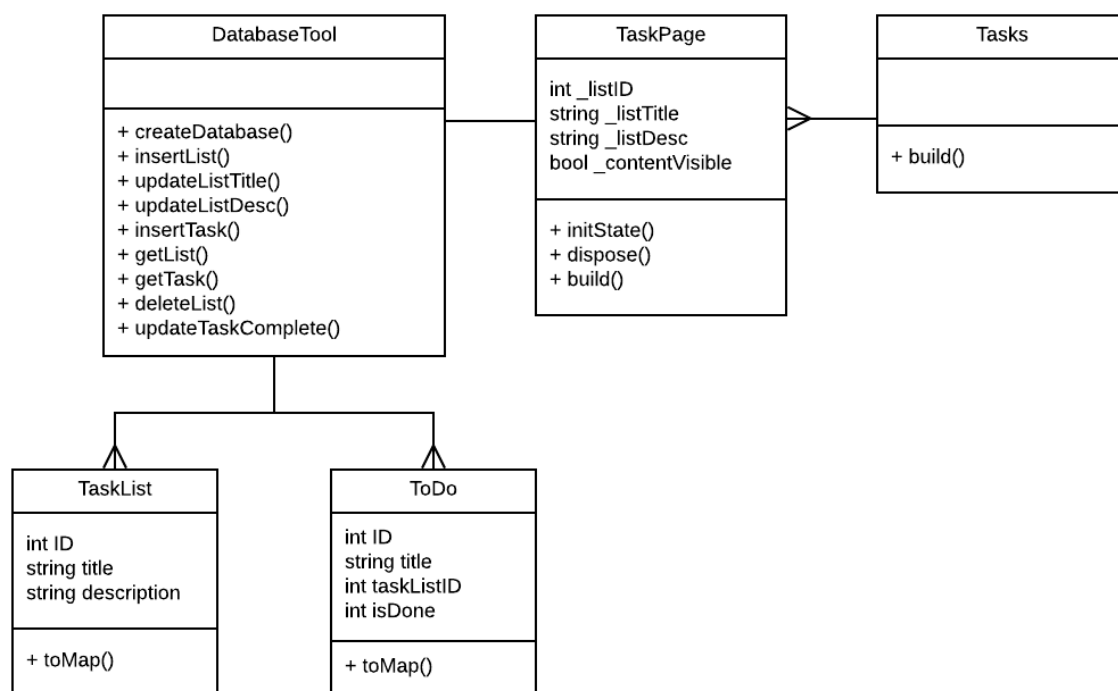


Figure 14: To-Do List Database Class Diagram

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 on a daily basis. 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.

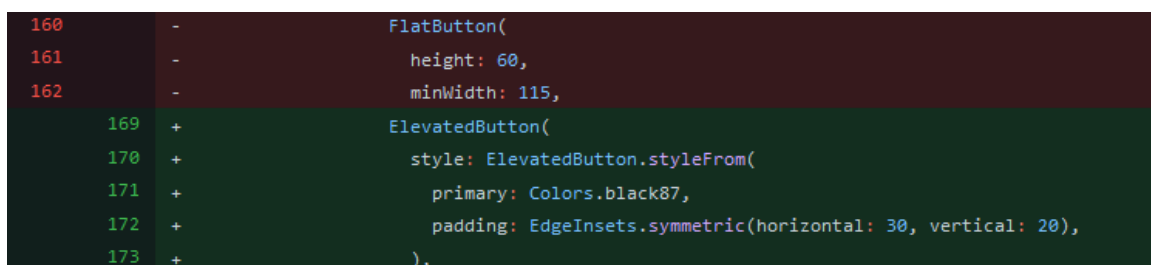
The solution involves storing the date of when a user loads up the app within the offline database. Whenever the user then opens the Pet page within the app, a check is first made to see how many days it has been since they last used the app. If it has been more than 1, the pet's statistics are updated 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 in order to fulfil the project's 'must have' requirement.

Deprecated Functions:

Midway through development, Flutter had released an update (Flutter SDK - Breaking Changes, 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. With the aim 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 the figure below:



```
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 15: Screenshot of Git Commit Log

Dynamic Pet Statistics:

For the display of Kenny 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 Kenny, for example, the hunger bar would update immediately with a simple animation. To do this, the team used the Flutter Charting (charts_flutter, 2021) external library.

```
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 16: Code Block of Model Data Format

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

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 16) when accessed. This meant that when the user opened the Pet page on the app, they would see Kenny'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.

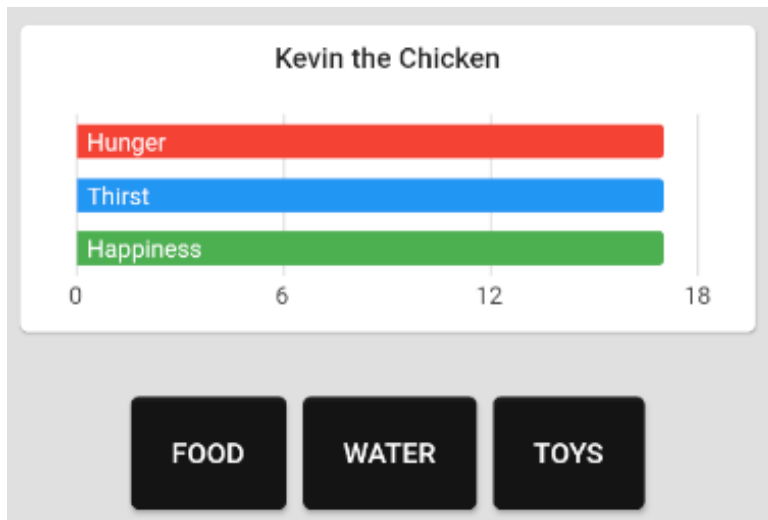


Figure 18: Screenshot of Pet Data



Figure 17: Screenshot of Pet Wake-up Screen

As a work around, the team implemented the 'wake Kevin up' button. 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, accompanied by an evaluation of its implementation. The test plan and reports referenced in this section can be found in the Appendices.

Objectives:

- Research into gamification
- Design and develop a new application
- Test and improve that application
- Release an alpha-stage of that application
- Design and develop a website

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 specific discussion of these features.

- The application should run on an Android device.

Through the use of Flutter and Android Studio, the team has been able to produce a functional APK version (<https://www.lifewire.com/apk-file-4152929>)(Add APK link?) 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 for the application as we're able to run it on our personal devices. 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 fulfill its aim. Through the testing process (test numbers here) the team were able to determine that the implementation of these features were 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 user's with an incentive to complete their tasks. Thus, it is also considered one of the team's must have requirements. In order 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 in order 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 as seen in Test No.X.

- 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 test no.x the team was able to successfully integrate this feature within the application.

- Develop a website to be used as a showcase for the team's capabilities as developers and to highlight the skills we've acquired during this project

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 is able to present themselves to employers. Hence, the team decided to use the website for this project as a way to showcase the team's capabilities in a simple and accessible way. Despite the project being available via the GitHub repository, the website (website link) allows non-tech savvy individuals to engage with the project and learn more about it without the technical barrier of Git-related knowledge.

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. A majority 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 as users may want to separate out their tasks based on specific categories; or features such as in-app alerts or daily notifications to keep users engaged with the application.

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 on a daily basis. Therefore, encouraging users to complete tasks for in-game currency as they have to 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.

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 personal to-do lists. The successful deployment of this feature within the app can be seen within (test no.x).

Finally, the application requires users to register an account prior to obtaining access to the application. The credentials are managed using Google's Firebase authentication console. 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 a future use of the account feature would be to combine it with a cloud-based database so that users are able to 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 (the 18th 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 C.

During development, the team made various decisions in 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 Studio, n.d.) and Flutter's Dart (Dart programming language, n.d.) due to the multitude of functionalities both tools provide. Moreover, Dart has a mildly similar syntax to HTML so the team agreed that it would be the quickest to adapt to and learn. This decision came down to the experience the majority of members in the team had done some level of website 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, 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 controls such as language translation and colour blind mode (among others) to make it accessible to anyone that may need it, in accordance with the (Equality Act 2010, 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, 2018) by making the database virtually unattackable, resulting in minimal issues in regards to 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, n.d.) to manage this project. The team communicated on a daily basis on the private Discord 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. If the project was going to be conducted over a longer period of time and required more efficient

resource management, the team would definitely 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 of the team's work being conducted online, it was paramount that backups were made and updated on a consistent basis 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 final outcome as almost everyone 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 on a consistent basis (as shown in Appendix C) and most members contributed when assigned with work to complete. It could be argued that the team's personal 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, there are a number of issues that need to be taken into account such as Professional, Social, Ethical and Legal Issues. The team has collated a list of issues that must be followed in order 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 (BCS, 2015) 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 take into account the Software Engineering Code of Ethics, published by the Institute of Electrical and Electronic Engineers Computer Society (IEEE-CS).

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. In the BCS Code of Conduct, it states that the team should "have due regard for public health, privacy, security and wellbeing of others and the environment" (BCS, 2015). 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), 2018) and (Data Protection Act, 2018).

When developing the project, the team had to ensure that the project did not breach the (Equality Act 2010, 2010). This act protects people, whether in the workplace or in the 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. In relation to 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 also 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 | ACM, 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, it was important that the team focused 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 cause the 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 BCS' 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 was 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 (BCS, 2015), 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:

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

The main legal issue was in relation to the collection of data required by the application. For instance, to provide leaderboard 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 is not limited to: usernames, task names, and email addresses. As a result of this, it is paramount that members in 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.

Appendices:

Appendix A - Survey

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

In the questionnaire, the team asked a variety of questions for the purpose of gaining 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 from 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 users favorite 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 in order 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 general 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 19: Survey Questions Part 1

Procrastination & To-Do lists

*Required

Task Tracking Apps

We're looking to gain insight into the market and to research some other options that we may have potentially overlooked ... your input here will help us better understand our 'competitors' and of how we can differentiate our work from the others.

What is the application's name? *

Your answer

Are there any features you like about it in particular? *

Your answer

What features would you like to add to it? *

Your answer

Any features you dislike?

Your answer

Figure 20: 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 21: Survey Questions Part 3

Appendix B - 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 principals, 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 so as 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.

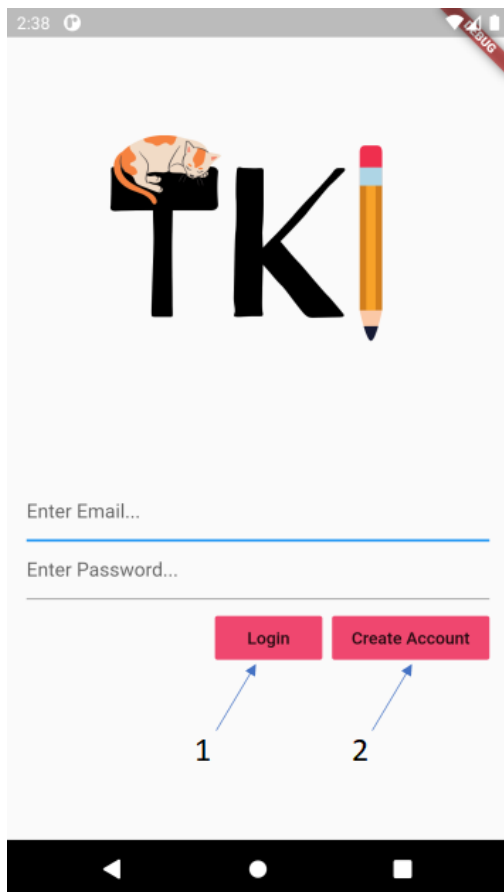


Figure 22: UI Sample 1

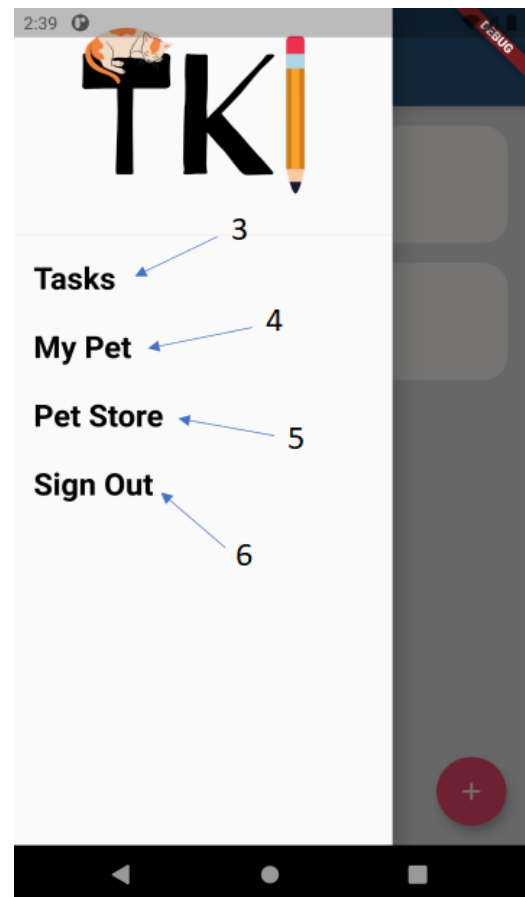


Figure 23: UI Sample 2

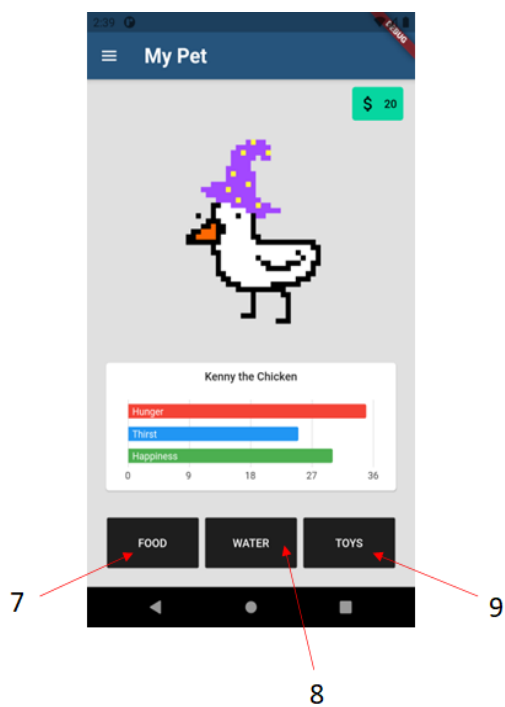


Figure 25: UI Sample 3



Figure 24: UI Sample 4

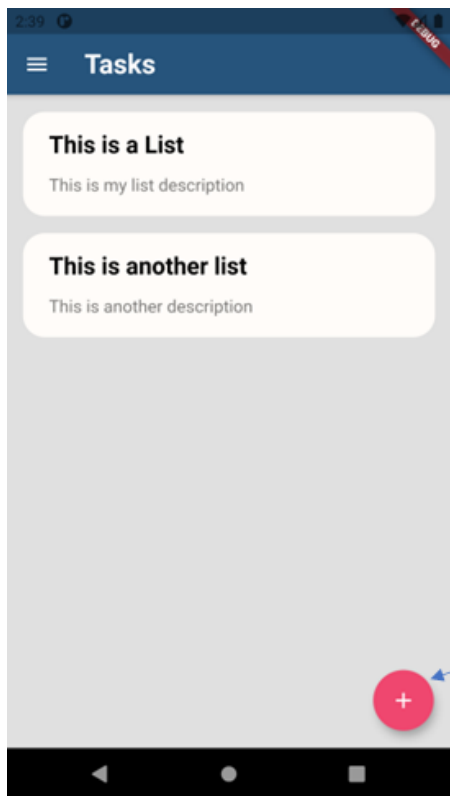


Figure 26: UI Sample 5

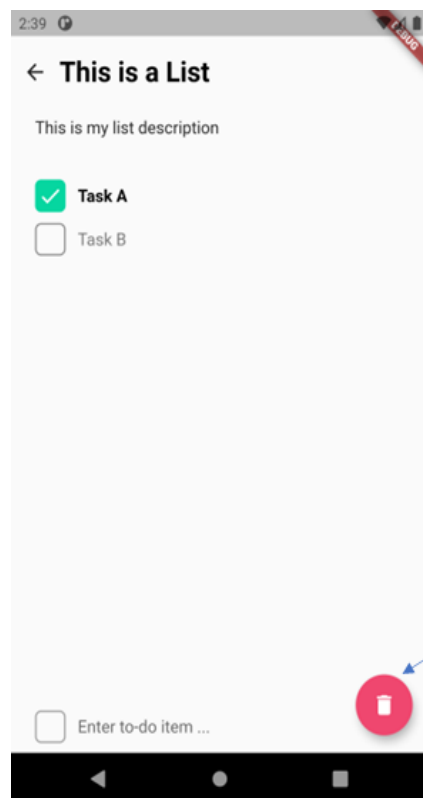


Figure 27: UI Sample 6

| 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 upon, triggers an action where a task can be added. |
| 12 | This button when clicked upon, triggers an action where a task can be removed. |

Appendix C - 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 28: Meeting Attendance Log

| | |
|----|--|
| 1 | 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 4PM. |
| 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 | 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 quality of work and ensured that everything was progressing in-line with expectations. Discussed the need to keep Simon inline to ensure that the work gets completed on time and to a good standard of quality. |
| 13 | Uploaded the first 3 sections of 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 D - Risk Analysis and Mitigation Plan Weighting Table:

| 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 a little to no impact on the project's progress |

Appendix E – Test Report: