# Software Engineering Coursework 2

# Team Plum

December 15, 2023

## Table of Contents

# 1. Introduction

Link to project repository: Here.

This report is a combined process and product documentation for a group software engineering project completed as part of the CM50109 module. The main section of the document is segmented and recorded by "sprints", which represents delivery iterations to the customer. Over eight weeks' time, seven MSc students, each from different academic and professional backgrounds, developed a serious game while implementing an agile software development processes and practices. The game "Bloom" is delivered as an Android mobile app with educational and practical purposes of teaching users how to take care of various indoor plants.

The following sections are structured as follows. Section 1 gives a high-level introduction of the project, including project overview, team composition, project timelines, a summary of agile process and practices implemented, and a note to readers on how to navigate this report. Section 2 details a project retrospective, which summarises retrospective thoughts and reflections of the project experience upon its completion. Section 3 is a full record of team processes and product development, organised by sprints. Section 4 concludes.

## 1.1. Project Overview

The project documented in this report entails the development of an edutainment serious game centred around the practical aspects of plant maintenance and plant care. "Bloom" is envisioned as an engaging learning platform, designed to teach users about indoor plant care through a blend of entertainment and education. Users will explore a diverse group of indoor plants through quizzes that progressively increase in complexity as users progress within various levels of "Plant Masteries". The difficulty in attaining a specific mastery will be dynamically tailored to reflect the real-life challenges associated with each specific plant. This is integral to the game's core functionality, as a primary objective is to aid players in the effective management of their real-world plants. The game will achieve this by seamlessly integrating reminders for crucial care activities, such as watering and other essential tasks. The core game mechanics encompasses the following components:

Sun Core Features:

- Oracle: A guiding character that provides advice and information.
- Educational Quizzes: Interactive quizzes that teach about a variety of indoor plants.
- Plant Maintenance: Virtual care of plants, mirroring the responsibilities of real-life plant maintenance.
- Interactive Hub Interface: A virtual bedroom where users interact with different elements of the game.
- Difficulty Levels: Categorization of plants into Easy, Medium, and Hard levels for progressive learning.
- Real-time Plant Management: Features like watering timers to remind players of real-life plant care tasks.
- Player Rewards and Achievements: Earning experience points, unlocking mastery levels, and receiving achievements for various milestones.

Technologies Used:

- Programming language: JavaScript (ES6+).
- Frameworks: React, React-Native, Expo.
- Additional Tools: Node.js, Git.

## 1.2. Team Plum Composition

The software engineering team comprises of seven students on the MSc Computer Science programme. The assignment of teams considers individual backgrounds as well as MBTI personality test outcomes. A summary of members' backgrounds and MBTI results are shown below.

| Name (A-Z) | Academic Background | MBTI Result |
|---|---|---|
| **Alec Mason** | BSc in Archaeology | Virtuoso (ISTP-T) |
| **(Claire) Siqi He** | BA in East Asian Studies,<br>MSc in Development Economics | Defender (ISFJ-T) |
| **Daniel-Favour Oshidero** | BSc in Architecture | Commander (ENTJ-A) |
| **(Eddie) Pang Hoi Chan** | BSc in Traffic Engineering | Virtuoso (ISTP-T) |
| **Marat Danyarov** | BSc in Finance<br>MA in Economics | Defender (ISFJ-A) |
| **Yan Chun (Ivan) Yeung** | BSc in Economics and Finance | Logistician (ISTJ-A) |
| **Udit Bhatia** | BSc in Information Technology | Mediator (INFP-T) |

## 1.3. Project Timeline and Process Overview

The project is organised into eight week-long sprints. The lab session on Wednesday 18th October is flagged as the start of Sprint 1. Customer meetings take place every Wednesday from 25th October onwards. The team decided that customer meetings will mark both the end of the previous sprint and the start of the next sprint. During each sprint (Wednesday to Wednesday), the team holds two stand-ups, usually on Fridays and Mondays. Sprint planning meetings take place every Wednesday immediately before or after the customer meeting.

The team keeps a product backlog and sprint backlogs. The product backlog lists tasks required as part of the overall product delivery, and sprint backlogs are weekly to-do lists for individual sprints. Sprint backlogs are typically created on Wednesdays during the planning meeting and updated after each team stand-up wherever necessary. No updates to the product backlog were made after its initialization in Sprint 3.

A misstep in the team's process design is the absence of official retrospective meetings. However, the team incorporated individual retrospectives in written form into the agile process design. After every sprint, each member is asked to complete a survey template briefly listing their achievements and indiscretions over the sprint, along with any thoughts, concerns, and exceptions they wish to raise. Each member has the option to keep their reflections confidential, in which case it is only visible to the documentation lead. Moreover, informal retrospective discussions happened inadvertently over most team meetings, all of which are recorded as "analysis of current standpoints" in the meeting minutes sections in the sprint documentation below.

The below figures visualise the project timelines as well as major processes adopted as part of an agile development team. As recorded and reflected in Sections 2 and 3, several practices such as team stand-ups, retrospectives, and record backlogs were only implemented from Sprint 3 onwards. This is deemed as the combined result of negligence plus lack of agile awareness in the early weeks of the project timeline.

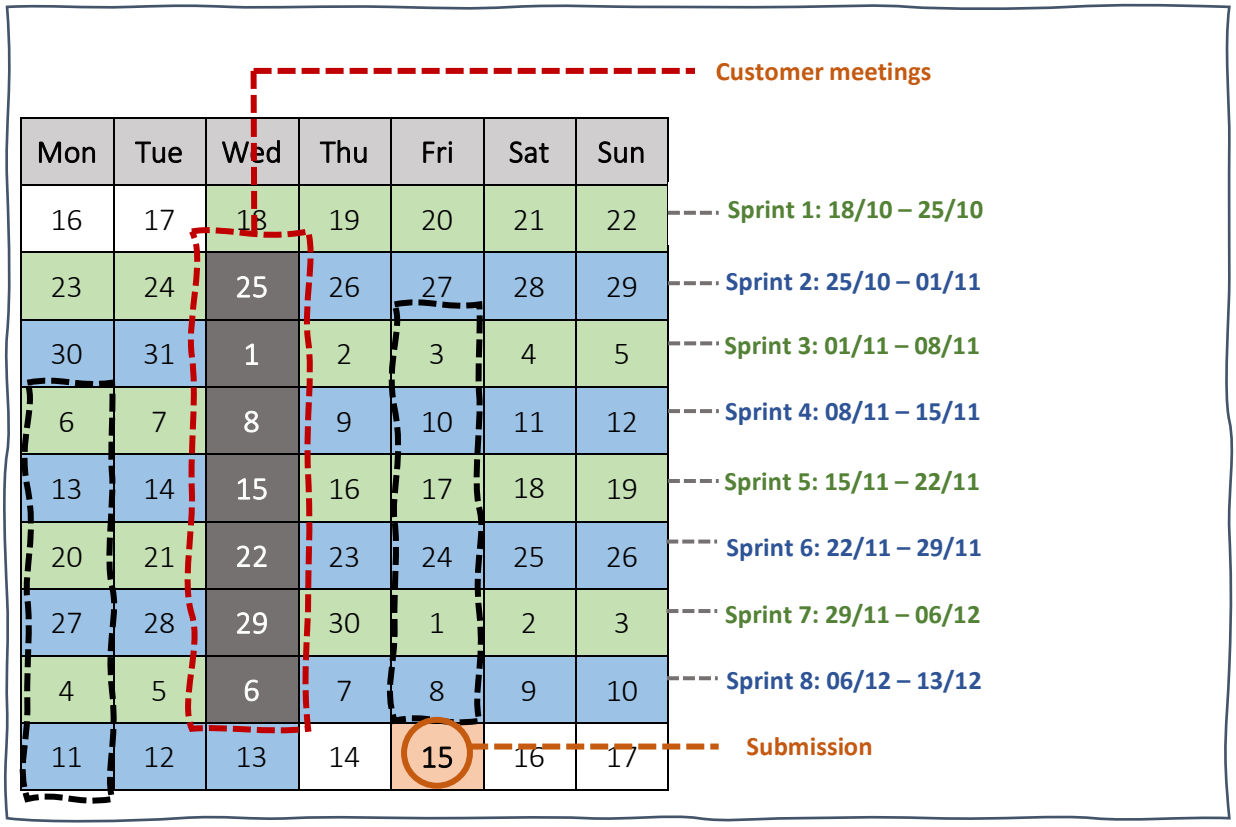



Figure 1.3 (a). Project timeline chart.

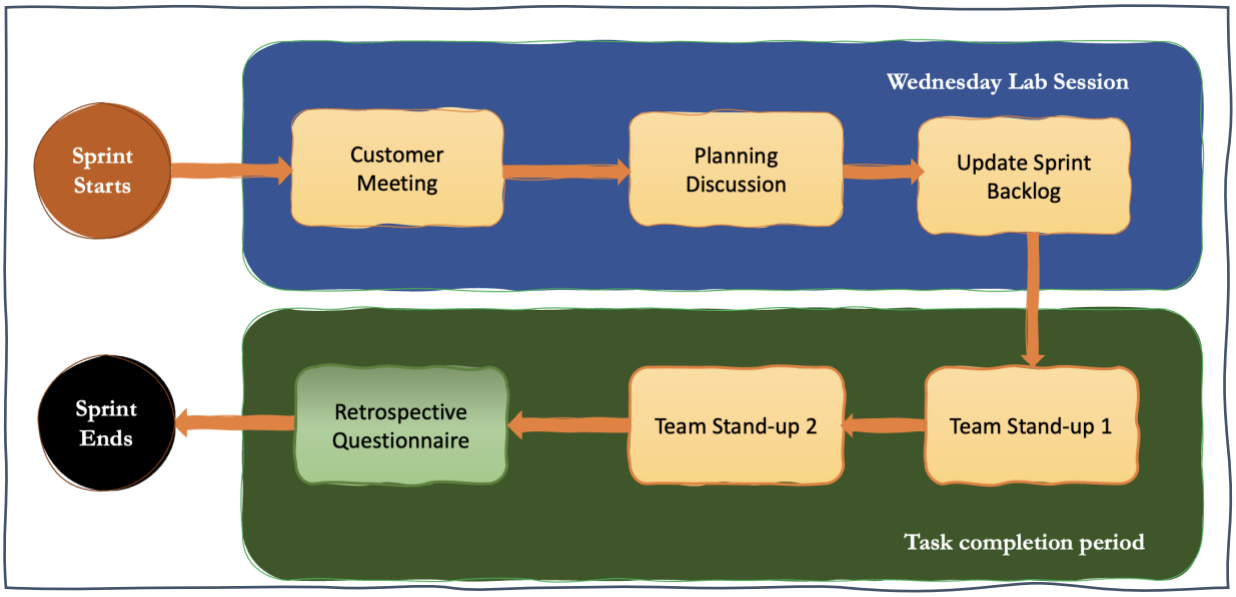
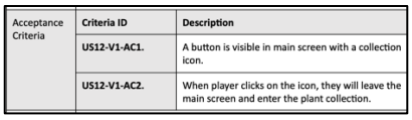
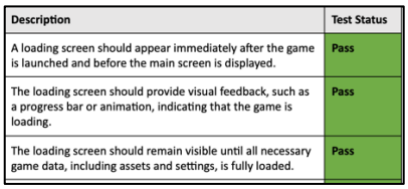
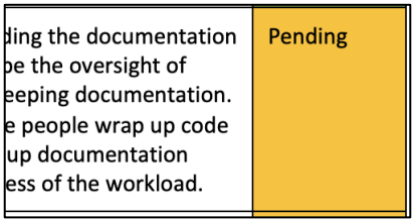
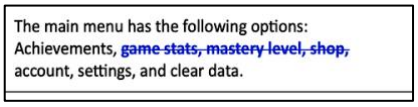
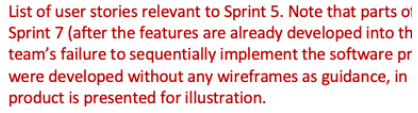
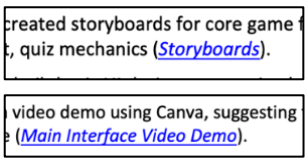
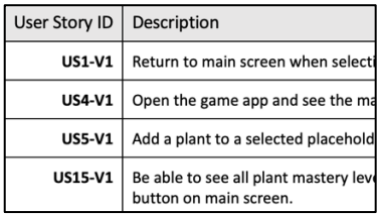
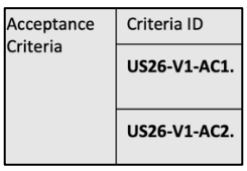



Figure 1.3 (b). Sprint internal structure.

## 1.4. Notations

The following notations applied in this documented are listed with examples to help readers better understand the presentation and structure of this report.

| Item | Purpose | Example |
|---|---|---|
| **Coloured cells in tables.** | **Light grey:** Usually the table header for differentiation. |  |
| | **Green:** Typically representing a complete status, such as pass state of a test, completion status of backlog item, successfully handled exception. |  |
| | **Amber:** Typically representing an incomplete status, such as an incomplete test, incomplete backlog item, or unhandled exception. |  |
| | **Dark grey with white foreground:** Typically denoting an item that has been deleted or voided. |  |
| **Bold blue text in User Story section** | To highlight the changes made to the previous version of the user story. |  |
| **Red text** | Used to add notation to bring attention to certain facts/incidents. |  |
| **Links** | To cross-reference between connected items and sections, and link to external file sources. |  |
| **Unique identifiers** | Each user story and acceptance criteria are given unique IDs.<br>User story IDs are formatted as:<br>"**US**XX-**V**X", denoting **U**ser **S**tory number and **V**ersion number.<br>Acceptance criteria IDs are formatted as:<br>"**US**XX-**V**X-**AC**XX", denoting **U**ser **S**tory number, **V**ersion number and **A**cceptance **C**riterion number. | User Story ID:<br>  <br>Acceptance Criteria ID:<br>  |

# 2. Project Retrospective

The section below provides a retrospective of the project from start to end, integrating thoughts and reflections of all members on the software engineering team. Points were collated and discussed over the second team stand-up of the final week (Sprint 8) on Monday 11th December. A series of informal discussions also took place over the week between documentation lead and individual members to gather any individual thoughts people wished to anonymously raise.

There are three main reasons for implementing the project retrospective process.

First, despite delivering a product that meets customer expectations, the team's agile process implementation has been worryingly chaotic throughout the eight weeks. One aspect is the absence of a collective retrospective process, the reasons for which will be discussed in this section as well. What the team has instead chosen to do is to gather individual retrospectives by conclusion of every sprint – implemented for Sprints 3 to 6. As will be shown in the discussion below, the lack of reflective awareness and limited open discussion on the team's mistakes was likely to be a contributor to the issues encountered in later stages of the project.

Second, the purpose of the project retrospective is to wholistically evaluate the team's performance over the course of the eight weeks and extract any lessons that could be taken to the next software engineering project in everyone's professional (or academic) careers. As many would agree, it is sometimes indeed very difficult to see what is happening in the bigger picture until the journey is complete. Hence, this conversation is particularly important to help everyone get the most out of the learning experience.

Third, as mentioned, individual retrospectives were only implemented for Sprints 3 to 6. This leaves two gaps over the eight-week period: The first 2 sprints (Sprint 1 and Sprint 2) and the last two sprints (Sprints 7 and 8). Hence, the project retrospective below also incorporates discussions around the periods previously neglected.

It is important to note that as the project retrospective took place in Sprint 8 after most deliverables were complete, the opinions in this section may have inconsistencies with individual sprint retrospectives (which were thoughts formulated immediately after the conclusion of each sprint). Such inconsistencies are expected, since one's wholistic views of the project over eight weeks' time should indeed come from a fresh perspective distinguished from what tend to be obscured views held at various points in time.

The two figures below visualise the project roadmap over eight weeks. Figure 2.1 (a) displays how broad areas of work were distributed across sprints. As shown, game design and preparation spanned across Sprints 1 to 4. Documentation began in Sprint 3 and lasted all the way to the project completion date. The game development (coding) period took only three weeks in Sprints 5, 6, and 7. Figure 2.1 (b) records the team's total number of hours dedicated to the project over each sprint. Sprint 2 has the least hours spent while Sprints 6, 7, and 8 all exceeds 140 hours. In total, the team committed approximately 700 hours to the completion of the project from 18th October to 15th December.

## 2.1. Project Roadmap

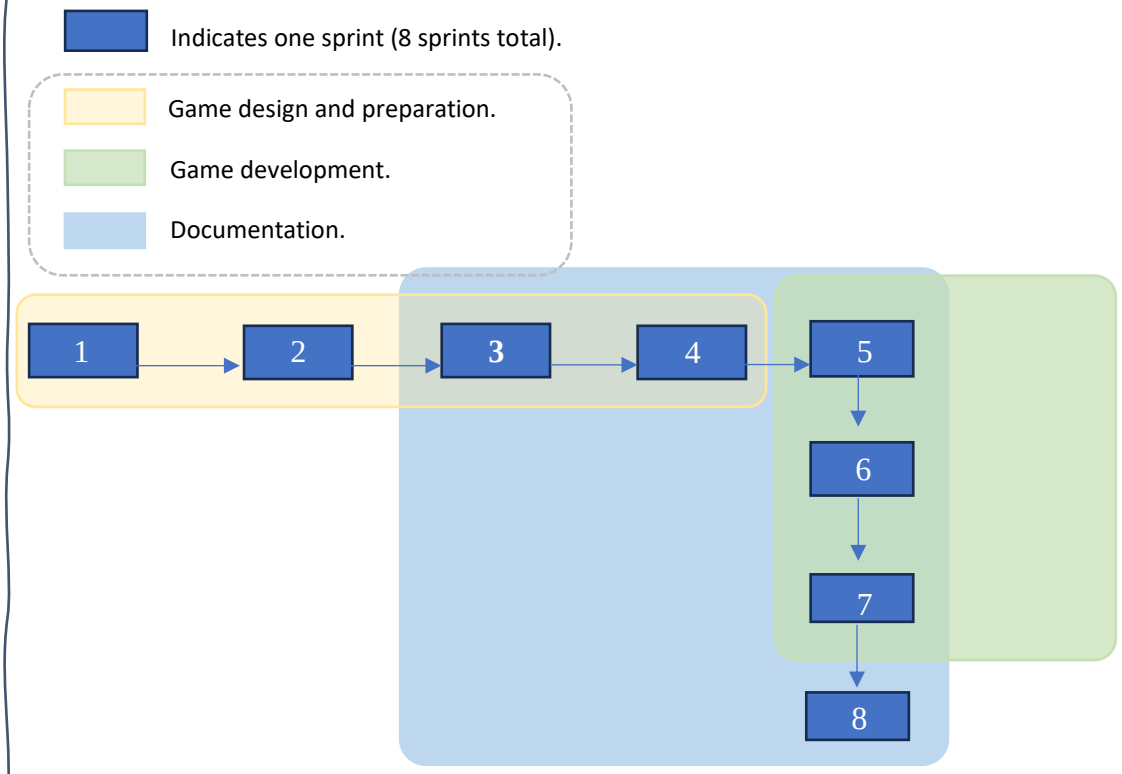



Figure 2.1 (a) Retrospective work distribution by streams.

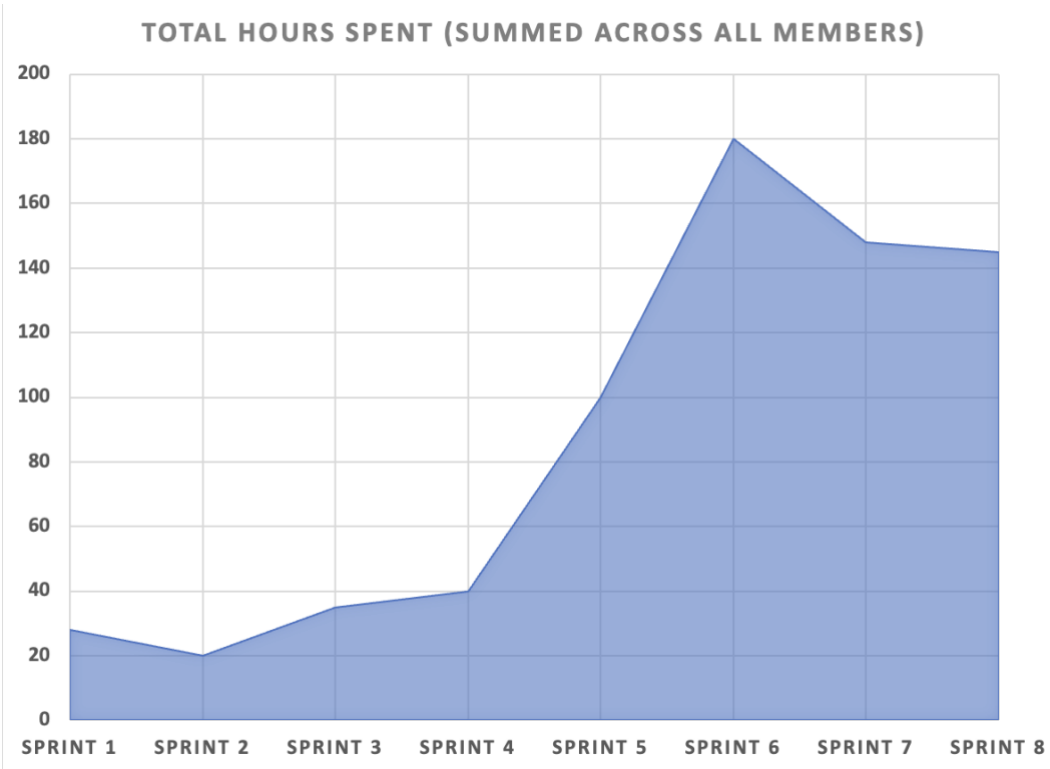



**Figure 2.1 (b) Total hours spent on delivery summed across all members.** Hours include Wednesday in-person lab sessions (customer meeting, TA meetings, planning discussions), team stand-ups, and individual efforts throughout the week.

## 2.2. Analysis of Project Trajectory

As shown in the graph above, progress throughout the period resembles bursts of energy from Sprint 5 onwards, rather than consistent efforts as one may expect across eight weeks. Contributing most significantly to the uneven distribution of work across time was the unfortunate reality that everyone had to balance several simultaneous coursework commitments until midway through Sprint 5. While this reason by itself was not decisive to the delay, its concurrence with the initial choice of using Unity as the game development engine was certainly adverse. As will be seen in the Sprint 4 documentation, the fact that only one member had any prior exposure to game development engines pushed the team into a danger zone. With everyone expecting a steep learning curve before getting familiar with the engine, the burden of doing so in everyone's spare time became frankly unrealistic with concurring commitments to other coursework tasks both within and outside of this module.

The initial appeals when choosing Unity 3 with C# was a combination of Unity having readily available templates online and the team's willingness to learn a new tool. The first appeal was very quickly nullified by the end of Sprint 3 after discovering that no templates seem to be a good fit for the game design. However, it was retrospectively a mistake for the team to persist with this decision for yet another week even after knowing the entire game will need to be built from scratch. In hindsight, members have pointed out that perhaps what caused the blind optimism is everyone's belief that as soon as other coursework is complete, all members can then spare a few solid days to fully immerse in Unity training, which will then get everyone onboard with the engine. It was, however, the customer meeting in Sprint 5 that brought unprecedented panic to the team as the customer flags our progress to be severely behind. Moreover, after Alec gave a high-level tutorial of the Unity functionalities which had so far been used to develop the interface, everyone realised that the engine itself is more complex than what people had imagined. It became apparent that building a full-fledged game with Unity requires a deep understanding of the engine's internal logic and the plethora of functionality it offers, rather than a simple mastery of C#.

Hence, it was only then a conversation finally took place regarding a switch to toolkits which are more intuitive to master, and in a few days' time the decision was made to use JavaScript with React Native instead. Favour and Ivan collectively built a game prototype whilst everyone else was still powering through the CM50260 midterm coursework in Sprint 5, and only in Sprint 6 did game development fully kick off with everyone assigned coding tasks to complete. By end of Sprint 6, the game arrived at a stage of 80% completion, after which some members then moved back to documentation while Favour, Ivan, and Udit continued to build the last remaining features. By the end of Sprint 7, the game reached full completion as everyone turned their focus to writing the documents required for submission.

Therefore, observing the overall trajectory, code development effort was highly concentrated in Sprints 5, 6, and 7. Although delivery was not impacted, benefiting from everyone's altruism and hard work, the highly concentrated workload has indeed made everyone's life quite tough over the final weeks. All of us admit that the positive delivery outcomes were largely the result of sheer luck that are unique to project circumstances – that the customer had no requirements on coding language, the team had Favour as an experienced React developer, and that the blessing that every feature developed in sprints 5, 6, and 7 met customer expectations without the need to revert and make any major changes. Many of these conditions may well not be the case with client projects in a commercialised industry setting.

What could the team have done to prevent this from happening? The key lesson to be learnt, is to base decisions on reality rather than perceptions. This is also aligned with the agile philosophy which encourages flexible decision-making according to evolving project requirements. At the point when Unity was chosen as the development platform, the team was drawn to perceptions which had little relevance to the project requirements and circumstances: Unity had available templates and

resources, and everyone is keen to learn a new tool. What should instead be the set of dominating factors are constraints such as project timelines, members' technical experience and learning abilities, estimated time and difficulty in learning new tools, estimated weeks needed for code development, and so on. All these considerations were not brought to the forefront when choosing between various toolkits. Even as the team approached end of Sprint 4, the dilemma between customer's asks of a coded prototype and the team's inability to immediately start coding was still not directly addressed. Instead, the burden was temporarily given to Alec and Ivan to start building code base and gather learning materials, presuming that everyone would eventually catch up with the necessary skills. All this happened *without* the team fully recognising that the poor choice of Unity itself was precisely the reason behind the delay.

It is however worth noting that had the team been given more time, things may well have worked out differently. The project is nonetheless part of an MSc course module with the primary purpose of acquiring new skills. Hence, the eagerness to learn new tools such as Unity is not a wrongdoing per se, but a suboptimal decision under the circumstances.

Aside from code development timelines, another fiasco which led to chaotic scenes was the lack of a structured project management framework. Although a set of processes and practices were implemented from Sprint 3 onwards (e.g. bi-weekly stand-ups, backlogs, weekly planning meetings, etc.), the team had little awareness of implementing systematic processes that are essential for structured product development. In a typical agile environment, the software process model should start from customer requirements (listing user stories and developing use cases), then proceed to product design (building CRC cards), and finally game development. With each iteration, components such as use cases and CRC cards should be revised and versioned according to evolving customer requirements, before being used as a blueprint for code development. However, the procedural steps were largely disregarded during implementation, where everything happened concurrently rather than in the prescribed sequenced manner. The problem started as early as the first few sprints when the whole team already demonstrated weak awareness for following the software process model. As pointed out by the customer in Sprint 4, the list of user stories presented over the customer meeting had very little detail and no acceptance criteria. It had appeared that people subconsciously assumed that no acceptance criteria were required at this stage, and that a simple list of intuitive user stories would be sufficient to guide development. Likewise, by Sprint 5, only a very small selection of basic user stories was substantiated along with no CRC cards whatsoever.

Sprint 6 was when the bomb dropped as the team found itself falling behind on all fronts. Coding had just started with only a stripped-down prototype, user stories were still in elementary form, process documentation had barely started with only a handful of meeting minutes and individual retrospectives. In this critical juncture, the team had no choice but to pick priorities. Because the team was already falling behind customer expectations on the product development front, coding was inevitably flagged as the topmost priority. The subsequent scene was six out of seven members focusing exclusively on feature development in Sprint 6 while Claire stayed on process and documentation to catch up from previous delays. The unintended consequences of this arrangement, however, was a temporary segregation between product development and document writing. On the one hand, in the absence of substantiated use cases, people on the coding stream had no choice but to rush into development following mentally conceptualised ideas of what features should look like. People had frankly no time to formally construct or document use cases / UI designs on the go. On the other hand, the separation between documentation and coding as two parallel streams under extremely tight schedules made it virtually impossible for use cases to be written in unison with code development. The problem was made worse by the disproportionate ratio of members allocated between the two streams and the difficult real-time communication. Consequently, by the end of Sprint 6, game development was mostly complete, but product documentation was lacking. At this critical point, once again, the team decided on another burst of documentation effort in Sprint 7 and 8 to create a quick fix for the situation. All members excluding Favour, Ivan, and Udit (who stayed on coding to complete remaining game features) spent Sprint 7 on reverse-engineering user stories, use

cases, CRC cards, and UI designs based on a combination of code files, chat records, GitHub commit messages, and collective memory.

It was agreed amongst all members that this was far from ideal. The chaotic arrangements described above tremendously benefitted from the simplicity of the game mechanics (which enabled development based on mentally conceptualised designs), and compact timelines (which allowed members to reverse-engineer use cases based on fresh memory of what happened only a week ago). Moreover, it was exceptionally fortunate that Favour had a proficient mastery of React, which allowed him to build contextual variables in Sprint 6 that reduced the dependencies between code files and React components. The team must understand that it is very unlikely for these blessings to extend to other projects in the future. Had the game mechanics been more complex, it would be far more likely for people to be lost when mentally mapping the game flows and dependencies during development. Likewise, if the project stretched over months, recalling on-the-spot decisions made three months ago for sake of building documentation would also be impractical. Hence, everyone learned the hard way why a structured software process flow is crucial to project success and the significance of having substantiated user requirements and designs as guidance in the code development phase.

Overall, it is nonetheless commendable that the team had successfully navigated through a range of exceptions to arrive at the completion stage. As evidenced in later sections, this is inseparable from the uncompromised work ethics of every member, as well as the supportive team culture built from day one. Fundamentally, everyone was committed to ensuring quality delivery at all costs, evidenced by the team collectively spending 140+ hours per week on the project over each of the last 3 sprints. The allegiance and reliability every member demonstrated have been vital to what the team was able to achieve despite the many hurdles and missteps encountered along the way.

## 2.3. Retrospective of Early Stage

As shown in Figure 2.1 (b), the glaring contrast between periods before and after Sprint 4 spurs contemplation on what could the team have done differently in the early weeks of the project. As mentioned, Sprints 4 and 5 had the biggest collision with other assignments, which was not the case for Sprints 1, 2 and 3. In fact, the only major assignment occurring in parallel with the first three weeks was the first coursework report on agile processes. This suggests that while many have indeed found the first coursework a heavy piece to deliver, it was nonetheless highly possible that the team could have spared more time over first three sprints for better project preparation.

The reality was, with the first report still ongoing, the team did not fully appreciate the urgency of the project at the point when Sprint 1 had begun. The mentality that the only task for the first two weeks was to decide on a game idea prevailed. For instance, as shown in the sprint documentation in Section 3, it was in Sprints 3 (second half) and 4 when discussions around individual responsibilities and team processes took place, while Sprints 1 and 2 kept both to a bare minimum. In retrospect, much of the administrative processes that took place over Sprints 3 and 4 could have been brought forward to allow the project to start on more solid grounds. A List of such processes are summarised in the table below.

| | |
|---|---|
| Regular team meetings | Earlier meetings in Sprints 1 and 2 may have provided the chance for deeper discussions around game mechanics specificities. For example, as shown in Section 3.4, some of these conversations aligning on crucial flows and functionalities in the game design was still happening in Sprint 4, at which point prototyping should have already begun. Moreover, Agile is fundamentally built around frequent interactions between teammates for a plethora of extended benefits. Hence had the team developed stronger awareness of this, collective productivity could have strengthened from day 1 rather than from Sprint 5 onwards. |

| | |
|---|---|
| Skills audit and role assignment | A procedure as simple as a skills audit could have been completed in the first week rather than after the game idea is finalised. For example, had the team been able to assign documentation stream leads earlier, tasks such as building a documentation structure could be brought ahead instead of taking place in Sprint 4 where things were rushed due to other commitments. Documentation leads may also have had more time to investigate the required components of product documentation and create a template early on for game design requirements (user stories, use cases, tests, and UI), as opposed to the chaos in reality relating to product documentation. Development tools and development platforms could also be shortlisted from the results of the skills audit, ranked by difficulty for team members and suitability for each game idea. |
| Set up documentation structure | With or without an assigned documentation lead, it should be imperative that a structure for process and product documentation is prepared as soon as possible. Particularly with the discord in product documentation that arose in reality. It could well have been a task which the team collectively completes in a mid-week stand-up, or picked up by a few members before flood of mid-term assignment due dates. |
| Discussion around game development platform and tools | By Sprint 2, the team has clearly formed a strong and justified preference for the plant game idea, and hence should be in a place to start having discussions around the tools and platforms suitable for game development. This would allow more time for learning any unfamiliar tools as well as avoid the panic when customer released requirements to see a coded game prototype in Sprint 4. |
| Building backlog | In early stages, the team failed to register the significance of a well-recorded backlog. This is partially due to the scarcity of activities in Sprints 1 and 2, based on which a backlog was deemed unnecessary. Despite so, the team could have devoted more effort to constructing the backlog structure and format in week 1, which would have spared more time in the following weeks for other tasks. |

Based on the proposals above, the team has collectively constructed a re-design of what could have been a more efficient project roadmap, where tasks are more evenly spread out across the full duration. A graphical representation of the proposal is shown below in Figure 2.3.

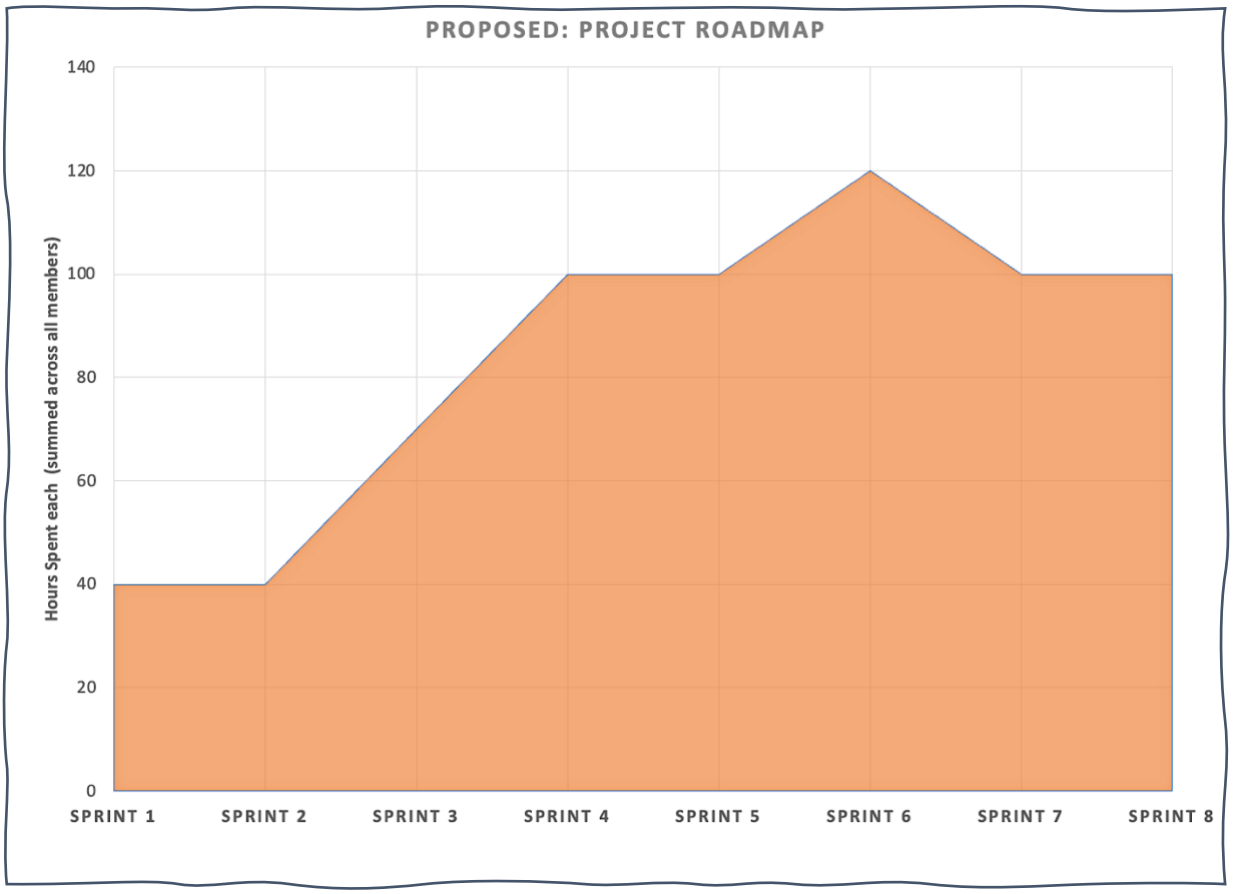



Figure 2.3 Retrospectively proposed revised roadmap in terms of hours spent per sprint.

Proposed distribution of tasks across eight sprints:

| No. | Administration | Coding (Game development) | Documentation |
|---|---|---|---|
| 1 | • Meeting set-up (planning meeting, retrospective meeting, sprint stand-ups).<br>• Backlog set-up.<br>• Teams channel and GitHub set-up.<br>• Individual weekly reflection set-up.<br>• Skills audit, potentially assign documentation leads. | • Shortlist the programming languages and packages according to skills audit.<br>• Make a rank of the difficulty and suitability of each toolkit. | • Build process document structure.<br>• **Build product documentation Masterfile (Excel spreadsheet, including sections for user stories, use cases, CRC cards, UI design).** |
| 2 | • Assign coding leads according to tools selected. | • Make preliminary decision on toolkit and platform.<br>• Organise knowledge sharing session for those experienced in development tool to give tutorial and learning guidance. | • *Process documentation.*<br>• *Product documentation.* |

| | | | |
|---|---|---|---|
| 3 | • *Processes implementation according to plan.* | • Code development (structure set-up).<br>• All members unfamiliar with development tools to allocate time for training. | • *Process documentation.*<br>• *Product documentation.* |
| 4 | • *Processes implementation according to plan.* | • *Code development.* | • *Process documentation.*<br>• *Product documentation.* |
| 5 | • *Processes implementation according to plan.* | • *Code development.* | • *Process documentation.*<br>• *Product documentation.* |
| 6 | • *Processes implementation according to plan.* | • *Code development.* | • *Process documentation.*<br>• *Product documentation.* |
| 7 | • *Processes implementation according to plan.* | • Code development (complete). | • *Process documentation.*<br>• *Product documentation.*<br>• *Installation guide and maintenance guide.* |
| 8 | • Project retrospective. | • Prepare installation file for submission. | • Process documentation (Complete).<br>• Product documentation (Complete). |

The revised roadmap enables code development to start by Sprint 3, with a more than a week for members lacking prior experience to learn new tools. Product documentation would also kick off as early as Sprint 2 where conceptualised game functionalities would form a preliminary user story catalogue. Early construction of team infrastructure not only yields more time for other workstreams in later periods, but crucially implements safeguards to collaborative progress and productivity. Early presence of appropriate agile processes such as backlogs, stand-ups, weekly reflections, interactive collaboration channels lubricate teamwork and boosts collective efficiency, hence the benefit multiplies throughout the duration of the projection. Moreover, adoptions of new processes and practices tend to have an adjustment period where members gradually become used to the addition. Therefore, the earlier the implementation are, the more time members have to be adapted to following them on a daily basis, as well as to suggest changes if any are discovered to be unsuited for project circumstances. This is a prominent lesson team has learnt from the laxity in the project starting stage.

## 2.4. Retrospective Analysis of Team Processes and Practices

Individual processes implemented the team were analysed as part of the project retrospective discussion, summarised below.

- Team environment and dynamics

**Egoless team morale:** Everyone was focused on collective achievements throughout. No one avoided any responsibilities. All members have at some point over the project volunteered to take on additional tasks to help other teammates. This was key to ensuring the final delivery despite the many obstacles we encountered along the way.

**Mutual respect and understanding:** The team cultivated a culture of mutual support and understanding from day one. No blame occurred between any teammates, and everyone was

extremely understanding to any individual circumstances that caused delays in progress. At both critical junctures during the project (the switch to JavaScript in Sprint 5 and the refocus to product documentation in Sprint 7), no one expressed frustration toward any members. Instead, each time a problem occurred, everyone took it to their duty to think of solutions, rather than investigating who is at fault.

Bonds outside of coursework context: Everyone made the effort to get to know each other on a personal level, which contributed to seamless collaboration over the weeks. All peers added to team banter and informal conversations on topics other than the coursework itself. This happens not only over Wednesday lab sessions but also WhatsApp group chats and regular team meetings.

Playing to individual advantage: Everyone gave their best selves to the team. According to MBTI personality test outcomes, it is hard to miss that the group is made up of six introverts with Favour as the only extrovert. As the weeks went by, the team subtly formed a dynamic where Favour would guide the flow of most meetings, while everyone else generously contributed their thoughts and suggestions. We all agree that the dynamics is a good fit for the team, enabling everyone to find a comfort zone in the team composition. Moreover, the significant proportion of "observant" members may have contributed to the practical mindset of the team. Including the persistent focus on task completion and success in handling a series of exceptions with realistic practical solutions.

- Communication: Meetings, messages, and stand-ups

The set of communication channels worked well for the team, combining informal as well as formal platforms. A development point some have highlighted is the absence of meeting agendas which led to some stand-ups spinning off topic with discussions that went down a rabbit hole. Most of the times people brought things up as they came to mind, which meant that important topics could have been easily forgotten.

- Backlog keeping

Most members believe that more weight could have been placed on keeping backlogs. This is largely due to the project having only an eight-week schedule, making tasks and responsibilities for each week easily memorable for every member. Hence it almost came to a point where everyone had to make the conscious effort to keep a backlog instead of relying on memory. As shown in later sections, the backlogs only started getting longer when the team began code development in Sprint 5 onwards. Many have note that the backlog did not function as well as it should have, given the boundaries between tasks were not always clear. For instance, the task to build a "Collection" feature should in practice be broken down into granular components, many of which overlap/interact with other backlog items. Subsequently, several sub-tasks emerged to resolve such dependencies, all of which were voluntarily taken up by members without recording them into the backlog.

Moreover, there was notable negligence toward the product backlog as a result of procedural missteps. The product backlog should in theory be a collection of user stories and other product requirements from which items are moved to the sprint backlog in the planning meeting. However, due to the team's unawareness of building requirements prior to starting game development, the product backlog ends up only having deliverables in the coursework specifications. It certainly did not fulfil the designated benefits in an agile software engineering team.

- Collaborative tools

GitHub was the main collaborative tool used for the project. Everyone demonstrated proficient mastery of Git except for an inadvertent merge conflict that occurred in Sprint 5. Subsequently, the rule was enforced to make an announcement in the group chat and receive everyone's confirmation before making a push. No conflicts had arisen henceforth.

- Workstreams and role assignments

There is shared doubt about whether splitting the workstream into code development and documentation was the best choice, particularly regarding product documentation. As discussed, the

team failed to register that product documentation should be recording developer decisions throughout the project, and therefore should not be segregated from the code work. The team however saw them as two independent tasks happening in parallel, hence assuming that people can "jump back" to writing documentation after coding is complete. We suggest perhaps a more realistic approach would be to make everyone responsible for designing a subset of user stories, as well as building these user stories into code. In this way no additional effort is required on communication between those recording the story in documentation with those developing the story into the code base.

# 3. Process and Product Documentation by Delivery Iteration

## 3.1. Sprint 1: 18th October – 25th October

### 3.1.1. Overview

Sprint 1 kicked off prior to submission of Coursework 1, when everyone has had the chance to brainstorm some serious game ideas relevant to their experience and background. Start of Sprint 1 is marked by the first team meeting on 18th October, officially launching the game development project. Two goals of were identified for the sprint: getting to know one another in the team and preparing a game idea pitch to customers in a week's time. As the everyone was still immersed in the workload of Coursework 1, the team expected Sprint 1 to be a week where the only focus was on producing pitch slides for game ideas.

### 3.1.2. Review

In the first team meeting, the team brainstormed several game ideas covering a range of topics. Notes were recorded and everyone was assigned to choose their favourite idea over the week and develop it into a one-page slide for presentation to the customer. Specifically, two important selection criteria were highlighted in the idea generation process: the appeal of the backstory (whether the value of the game in the real world) and game continuity (the difficulty building game tasks into progressive levels). Aware that all members were still spending significant time on Coursework 1, no formal processes or practices were designed or implemented in Sprint 1, except for creating a WhatsApp group for day-to-day communications. By end of the week, four game ideas were developed into slides, consisting of:

- An Antiquities game highlights the effects of the entire antiquities trade and teaches the player to apply laws which curb illegal antiquity sales.
- A Plant learning and management game informs people about how to take care of a variety of indoor plants through the form of gamified quizzes, and helps player manage any plants they may have in real life.
- A budgeting game which gamifies personal finance management by setting tasks and goals in the budgeting process.
- A social awareness game that teaches players how to identify social wrongdoings via animated video acts.

### 3.1.3. Summary of Key Events

- Team formulation.
- Initial game idea brainstorm.
- Four shortlisted game ideas to pitch to customer.
- Limited time commitment to the project and absence of formal processes.

### 3.1.4. Sprint Preparation: Brainstorming Session and Planning Meeting

| | |
|---|---|
| **Date** | 18th Oct 2023 |
| **Location** | Campus – Chancellor Building 5.13 |
| **Attendees** | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |
| **Brainstorming Session** | **Introduction:** See team composition table.<br>**Game idea brainstorm:**<br>**Idea 1: Antiquities game** |

| | |
|---|---|
| | • **Context:** Illegal antiquities market.<br>• **Mechanism:** Play as an agent investigating smuggling ring, examine artefacts and check legitimacy of documents for items sold.<br>• **Criticism:** Difficulty of designing game assets without highly specialised domain knowledge, particularly those at higher levels.<br>**Idea 2: Educational game**<br>• **Context:** Educational contents dull to learners.<br>• **Mechanism:** Gamified educational game (theme undecided) to help learners with a subject.<br>• **Criticism:** Some subjects may be hard to gamify.<br>**Idea 3: Coding game**<br>• **Context:** Increased public interest in learning to code.<br>• **Mechanism:** Inspired by the CSS game FlexBox Froggy, which helps player learn CSS flexbox by visualising flexbox arrangements with frogs and water lilies. Player needs to enter correct code to reach next level.<br>• **Criticism:** The game self can perhaps be too simple as the main development task is to write programme tests with bare minimum game architecture needed.<br>**Idea 4: Social awareness game**<br>• **Context:** Rise of global inclusivity and cultural diversity in the 21st century.<br>• **Mechanism:** A series of animated videos showing the interaction of different characters. Player is asked to mark down the timestamp whenever they realize the in-game characters are doing something wrong (behaviours of micro-aggression, discrimination, etc.).<br>• **Criticism:** Designing game levels can be difficult<br>**Idea 5: Professional simulation game**<br>• **Context:** Interest in learning about the day-to-day of working in a certain profession.<br>• **Mechanism:** Player pretends to be a professional in a particular industry (e.g., teacher, lawyer, bartender) and are asked to complete work tasks assigned to them. XP is awarded if tasks are completed correctly.<br>**Idea 6: Budgeting game**<br>• **Context:** Students and early-careers professionals often find it difficult to manage their budget.<br>• **Mechanism:** Player sets budgeting goals for a certain period (weekly, fortnightly, monthly, etc.) and records all income and expenses in the game. XPs are awarded if player completes their goals.<br>• **Criticism:** Difficult to design levels and measure success as people may have individual unforeseen circumstances. |
| **Planning meeting** | **Analysis of current standpoint:**<br>Although Sprint 1 started with everyone's focus still very much on Coursework 1, it was crucial that everyone acknowledges the decisive role of the first two sprints in the scope of the project. Having a sound and feasible game idea is of paramount importance, particularly as the game itself is of educational and practical nature. It was fortunate that many ideas were formulated in the brainstorming session with everyone making valuable inputs based on their academic and professional backgrounds. |

| | |
|---|---|
| | As everyone had just met each other, another implicit yet pivotal assignment for the team was to get to know each other on a professional level, i.e., learning each other's personalities, strengths, weaknesses and working styles. The survey of MBTI results show that most members seem to be tilted towards the introverted and practical side. It was certainly a good sign that members were able to offer insightful comments on each other's proposals, as well as prompt each other with questions that help to substantiate primitive ideas. Meanwhile, it was interesting to observe that Favour, who was the only extrovert identified by the MBTI results, naturally took on a leading role in the brainstorming session, facilitating between discussions, making sure everyone has a chance to express their thoughts, while keeping notes on a whiteboard. Everyone else had no problem with the leadership, and all were actively participating in the conversation without any hint of dominance or fear. By sheer luck, the team's personalities so far seem to be working out perfectly and no one seems to "stand out" in ways that requires reconciliation efforts.<br>**Sprint priorities:**<br>**Everyone** to select a game idea from the list (or come up with new idea) and elaborate into a slide for the pitch.<br>The slide should include a backstory, game mechanism, conceptualised difficulty levels, rewards.<br>**No stand-ups are scheduled for this week.**<br>**The Sprint 1 backlog can be found in Section 3.1.8.** |

### 3.1.5. User Stories, Acceptance Criteria, Use Case, and UI Design

N/A – No user stories were developed in Sprint 1 as the game idea is not confirmed.

### 3.1.6. CRC Cards

N/A – No CRC cards are deemed relevant for Sprint 1

### 3.1.7. Team Stand-up Minutes

No meetings were scheduled in Sprint 1.

Informal progress update on 24th October 2023 (via WhatsApp): Four out of 7 members had chosen a game idea and created presentation. (Link to file)

### 3.1.8. Sprint 1 Backlog and Completion Status

| TASK | DUE | PRIORITY | ASSIGNED TO | STATUS |
|---|---|---|---|---|
| Select one game idea and elaborate into 1-page slide. | 25 Oct | High | All | Complete |
| Combine slide into customer pitch presentation. | 25 Oct | High | All | Complete |

### 3.1.9. Exception Handling

No exceptions were raised in Sprint 1.

### 3.1.10. Individual Retrospectives

Individual reflection process started from Sprint 3 onwards.

## 3.2. Sprint 2: 25th October – 1st November

### 3.2.1. Overview

After receiving customer feedback around the four pitches, the goal for Sprint 2 was to narrow down on two ideas and elaborate them according to customer suggestions. As these was the only item in the Sprint 2 to-do list, the week remained uneventful as members continued to focus on Coursework 1 without yet fully registering the criticality of building a strong start to Coursework 2.

### 3.2.2. Review

The team very quickly shortlisted two game ideas immediately at the Sprint 2 planning discussion, held immediately after the customer meeting: The plant management and learning game (referred to as "**the plant game**" henceforth) and the gamified budgeting application (referred to as "**the budgeting app**" henceforth), among which the plant management game was highly preferred. Prior to the customer meeting on 1st November, the team pulled a vote on WhatsApp where everyone was asked to make select two options out of the four ideas pitched on 25th October. The plant game and the budging app received respectively the highest and second highest number of votes (7/7 and 6/7). Just before speaking to the customer on 1st November, the team had an in-person discussion around the customers' comments received for both game ideas last week and recorded bullet points for the presentation. It was agreed that plant game was significantly preferred over the budgeting app due to realistic considerations regarding effective game design and implementation.

### 3.2.3. Summary of Key Events

- Two shortlisted game ideas: the plant game and the budgeting app
- Strong inclination toward the plant game
- Limited time commitment.

### 3.2.4. Sprint Preparation

#### 3.2.4.1. Customer Meeting and Planning Discussion

| | |
|---|---|
| **Date** | 30th Oct 2023 |
| **Location** | Campus – Chancellor Building 4.7 |
| **Attendees** | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |
| **Customer Meeting** | **Presentations:**<br>Slide content in PowerPoint file: Link<br>**Customer feedback on each game idea:**<br>• **Antiquities:**<br>Simulating the role of an Interpol agent can be a fun and stimulating context for player.<br>Difficulty in creating actual information and data to make this game, i.e., information about the artifacts and documents.<br>• **Plant learning and management:**<br>Specification needed for what exactly is required to complete a level.<br>Other engagements, such as adding friends, high-score table.<br>How to deal with rare problems that happens in real life (e.g., diseases) but not covered in the game.<br>• **Gamified budgeting app:** |

| | |
|---|---|
| | Are users receiving advice on how they should budget, or if the game lets them set their own goals?<br>How to make it gamified?<br>Is it a game or a support mechanism, or both?<br>• **Social awareness clicker game:**<br>Interesting and niche idea.<br>Creating enough videos may be challenging. |
| **Planning Discussion** | **Discussions around product design based on customer meeting:**<br>• Antiquities and social awareness game difficult to implement due to tough requirements on game assets.<br>• Plant management game the most realistic due to straightforward game mechanics.<br>• Budgeting app still needs thought on the mechanics of gamified features and continuity, especially how to set standards for quantifying player performance (acknowledging that different people can have highly individualised budgeting priorities and goals).<br>**Analysis / reflections of current standpoint:**<br>From the outcomes of the customer meeting, the team has already formatively decided on the two most favoured ideas. This seems to progress ahead of schedule for Sprint 2 customer requirements. Nonetheless, as discussed in project retrospective, it was a pity that the team had no early plans to start agile process design this week, as doing so will undoubtedly better prepare everyone for the full engagement of product design in Sprint 3.<br>**Sprint priorities:**<br>No other priorities besides narrowing down on two game ideas and addressing customer concerns were identified at this point of time.<br>**No stand-ups are scheduled for this week.**<br>**The Sprint 2 backlog can be found in Section 3.2.8.** |

### 3.2.5. User Stories, Acceptance Criteria, Use Case, and UI Design

N/A – No user stories were developed in Sprint 2 as the game idea is not confirmed.

### 3.2.6. CRC Cards

N/A – No CRC cards are deemed relevant for this sprint.

### 3.2.7. Team Meeting Minutes

After selecting two game ideas on the WhatsApp group poll, the team had a brief discussion prior to the customer meeting on 1st November to address concerns raised by the customer relating to the shortlisted ideas.

| | |
|---|---|
| Date and Time | 1st Nov 2023, Before Sprint 2customer meeting |
| Location | On campus – Chancellor Building 5.13 |
| Attendees | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |

| | |
|---|---|
| Meeting | **Addressing concerns and brainstorming game features**<br>• **Plant management game:**<br>"Specification needed for what exactly is required to complete a level."<br>Player first learns about the plant in the form of knowledge cards, after which they complete a quiz designed for that level.<br>"Other engagements, such as adding friends, high-score table."<br>Both would be fun features to incorporate, but will need to be evaluated based on the team's skillset and the difficulty of building them.<br>How to deal with rare problems that happens in real life (e.g., diseases) but not covered in the game.<br>This can be incorporate into quiz content so the player can learn about them, but the in-game plant only requires watering and fertilising.<br>Other considerations – game continuity.<br>• Include a shop functionality where the player can purchase plant "skins" with XPs.<br>• Special plants such as a lucky clover plant can be a form of in-game reward.<br>• Player can unlock different backgrounds as their advance through mastery levels.<br>• Potentially partnership with plant shops and sending physical rewards: if player reaches certain levels (e.g., achieving full mastery for a predefined number of plants), a partner company can post a real plant to the customer as a gift.<br>• **Gamified budgeting app:**<br>Are users receiving advice on how they should budget, or if the game lets them set their own goals?<br>How to make it gamified?<br>The game oracle can generate periodic insights on players' spending habits, as well as give tips on areas for improvement.<br>If player completes target set for given periods, they earn XPs.<br>Is it a game or a support mechanism, or both?<br>The blurred boundaries between a game vs. support mechanism cast doubt on the feasibility of implementing the game idea. If it is a support mechanism to daily life, then setting quantifiable metrics to measure player performance would go against its purpose completely. Individual budgeting goals can be constantly revolving according to what's happening in their lives, such as an expected expenditure due to significant life changes may turn one's financial situation upside down overnight. Unless the game mechanics can account for a multitude of real-life complexities (which is beyond the team's capacity as conversional Computer Science students), the game can end up very much detached from people's actual budgeting needs. Therefore, such an attempt to gamify personal finances is unlikely to be suited for a serious game.<br>The team has reached the conclusion that the budgeting game is considerably less realistic to build in 6 weeks compared to the plant game. |

### 3.2.8. Sprint 2 Backlog and Completion Status

| TASK | DUE | PRIORITY | ASSIGNED TO | STATUS |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| Set up vote in WhatsApp group to shortlist 2 game ideas. | 1 Nov | High | All | Complete |
| Everyone to vote 2 options. | 1 Nov | High | All | Complete |
| Everyone to brainstorm game design responses to customer feedback. | 1 Nov | High | All | Complete |

### 3.2.9. Exception Handling

No exceptions were raised in Sprint 2.

### 3.2.10. Individual Reflections

Individual reflection process started from Sprint 3 onwards.

### 3.2.11. Sprint Retrospective

Sprint retrospective meetings started from Sprint 3 onwards. Reflections on Sprint 1 and Sprint 2 have been carried out as part of the Project Retrospective, documented in Section 2 of this report.

## 3.3. Sprint 3: 1st November – 8th November

### 3.3.1. Overview

The start of Sprint 3 is announced by the final decision to implement the plant learning and management game idea. With approval from customers, the team went into the third sprint aiming to build collaborative infrastructure, explore feasibility of game features, and begin creating demos that describes game mechanics. On the administration front, the goal is to set up processes and practices which the team will follow for the rest of the project, form team structure, and select the tools for game development based on project requirements and members' backgrounds. To advance the game idea into a tangible concept, the team simultaneously launched several streams of work: shortlisting plant species, plant quiz development, compiling user stories, creating video demo, drawing basic user interface, and building simple storyboards for core features.

### 3.3.2. Review

The team made commendable progress on all fronts in the third sprint. Over the mid-week catchup, several key decisions were made regarding team processes, practices, and tools. This includes assigning workstream leads, choosing development platform (Unity with C#), and setting up collaboration channels. Favour and Claire were assigned to co-lead documentation while Alec and Ivan co-leads code development. Several members contributed to creating demo visualisations. Alec made storyboards showcasing the core mechanics of the game, Udit created a video that presents the main screen layout, and Eddie constructed a user interface design with descriptions of each game element and their purpose. In the second half of the sprint, these creations were merged into a presentation for the next customer meeting. On top of this, Alec proceeded to compiling segment-wise user stories to crystalise the game composition, while Eddie moved onto drawing granular wireframes of UI design in conjunction with the game flow. Ivan dived into research on Unity training material and available templates the team can leverage for building mobile applications. Despite the absence of a clear management structure, everyone was able to take on and deliver on a good proportion of tasks contributing to team administration and game design.

### 3.3.3. Summary of Key Events

- Completion of team skills audit and subsequent role assignments: Alec and Ivan co-leading code development and Favour and Claire co-leading documentation.
- C# and Unity selected as development toolkit.
- Implementation of key processes, practices, and tools for agile teamwork: bi-weekly stand-ups, Microsoft Teams channel, WhatsApp group, backlog, etc.
- Development of storyboards, user story list, UI wireframe, and video demo for next customer meeting.
- Early signs of an inclusive, egoless team culture.

### 3.3.4. Sprint Preparation

#### 3.3.4.1. Customer Meeting and Planning Discussion

| | |
|---|---|
| **Date** | 1st Nov 2023 |
| **Location** | Campus – Chancellor Building 4.7 |
| **Attendees** | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |
| **Customer Meeting** | **Presentations (two shortlisted game ideas):** |

| | |
|---|---|
| | *"*" indicating that customer has made a comment, recorded in the "Customer concerns" section.*<br>• **Budgeting app:**<br>Continuity: XPs and awards are given when users achieve their goals.<br>Oracle: Give insights on spending habits and advice on better manage finances.<br>Drawbacks: It can be difficult to measure "success" as people can have very individualised budget goals*.<br>• **Plant management game:**<br>Continuity (If player finishes learning all the plants): A virtual in-game shop can "sell" different variations of the plants such as plant skins (pink cactus, or a very large cactus one might find in the dessert*)<br>A lucky clover plant can be included as a reward.<br>Game producers can potentially partner with plant shops and send the player a real plant matching the ones they have in the game*.<br><br>**Customer feedback:**<br>*(Linked with proposed changes to product design in planning discussion.)*<br>• Customer agrees that the plant game idea is more coherent to put together as a game, whereas the budgeting app is too much connected to and impacted by the multi facets of life.<br>• Customers suggest the idea of crowd-source plants, where players can upload plant profiles of their choice with facts of the plant. **[A1]**<br>• Customer encourages exploring partnership with companies with in-game ads (e.g., for brands of soil) for the feature of sending players real plants as rewards. **[A2]**<br>• Customer points out there could be potential mismatch between plant knowledge learned in game and the general plant species. For example, people might believe that facts learned about basic indoor cacti also apply to large dessert grown cacti (if sold in shop). Therefore, the game should make sure to match learning content with the plants provided in the game. If any rare species are offered, they should be accompanied by learning sessions on those species in question. **[A3]** |
| Planning Discussion | **Additions/changes to product design based on customer meeting:**<br>**[A1]** Consider adding crowdsource plants to user story shortlist.<br>**[A2]** Consider adding real-plant postage to user story shortlist.<br>**[A3]** Reconsider products in the "shop" feature – limit products to only those variations that do not require different caring procedures.<br>**Analysis / reflections of current standpoint:**<br>With the game theme approved by customer, the project has entered the game-design phase where the team must align on a coherent game flow and a reasonable set of functionalities before starting to code. The period is crucial to the success of the coursework as it maps out overall expectations for the deliverables. Although extensive planning is not encouraged as part of Agile, the team should, nevertheless, form an unambiguous understanding of customer expectations, and align on the list of features and logics to be incorporated in game design. Given the reality that the team cannot be collocated for 7 hours a day as agile methodologies prescribe, it is therefore critical for everyone to engage in team stand-ups and share thoughts wherever possible to boost communication and build an agile team environment. |

| | |
|---|---|
| | The same guidelines also apply to the project management aspect, where teamwork mechanisms must be established in Sprint 3 to best ensure collective productivity.<br>**Sprint priorities:**<br>As everyone in the team is still yet to submit Coursework 1 for this module on Friday, it was agreed that for the first half of Sprint 3 the focus was on setting up an administrative framework rather than any tangible deliverables, keep workload lightweight. Nonetheless, taking customer feedback onboard, everyone should indeed spend time thinking about how to develop the game idea into a roadmap of game mechanics. The team will regroup on Monday to collate and merge individual ideas and embark on systematic product design. In the second half of Sprint 3, effort should be made to begin producing presentation materials that showcases key functionalities to customers in the next meeting. This should include at the minimum user stories, storyboards, basic user interface design.<br>**Next meeting scheduled:** Monday 6th Nov.<br>**The Sprint 3 backlog can be found in Section 3.3.11.** |
| **Task Allocations** | Everyone to complete skills audit table (created by Favour during lab).<br>Everyone to make a start on substantiating product requirement and product design. |

### 3.3.4.2. Product Backlog

| TASK | DUE DATE | PRIORITY | ASSIGNED TO |
|---|---|---|---|
| Installation Guide | 15 December | - | - |
| User manual | 15 December | - | - |
| Maintenance guide | 15 December | - | - |
| Sprint 1 process documentation | 15 December | - | - |
| Sprint 2 process documentation | 15 December | - | - |
| Sprint 3 process documentation | 15 December | - | - |
| Sprint 4 process documentation | 15 December | - | - |
| Sprint 5 process documentation | 15 December | - | - |
| Sprint 6 process documentation | 15 December | - | - |
| Sprint 7 process documentation | 15 December | - | - |
| Sprint 8 process documentation | 15 December | - | - |
| Sprint 1 product documentation | 15 December | - | - |
| Sprint 2 product documentation | 15 December | - | - |
| Sprint 3 product documentation | 15 December | - | - |
| Sprint 4 product documentation | 15 December | - | - |
| Sprint 5 product documentation | 15 December | - | - |
| Sprint 6 product documentation | 15 December | - | - |

| | | | |
|---|---|---|---|
| Sprint 7 product documentation | 15 December | - | - |
| Sprint 8 product documentation | 15 December | - | - |
| User story development | - | - | - |
| Use case development | - | - | - |
| User story tests | - | - | - |
| User case tests | - | - | - |
| UI design by user story | - | - | - |
| Game development | 08 December | - | - |
| Testing | 08 December | - | - |
| Demo video creation | 22 December | - | - |
| Team questionnaire | 15 December | - | - |

### 3.3.5. Initial Development of User Stories

Based on the current conceptualised game mechanics, the following user stories have been tentatively shortlisted. The stories are organised hierarchically, starting with overarching epics (wide scope, low detail) which are broken down into stories with increasingly narrow scope and high detail.

Initial User Story List

**Epic 1 As a player, I want to learn about different types of houseplants.**

| | |
|---|---|
| I want to add new houseplants to my collection, so I can learn about them. | |
| 1. | I want a list of the available houseplants, so I can choose some to collect. |
| 2. | I want to choose houseplants from the list, to add to my collection.<br>• I want a certain number of slots in the **hub** which can hold houseplants. |
| 3. | I want a certain number of slots in the **hub** which can hold houseplants. |
| 4. | I want those houseplants to appear in the **hub**. |
| I want to be quizzed about the houseplants in my collection, to help me learn. | |
| 1. | I want to choose houseplants to be quizzed about.<br>• I want to select a houseplant, so I can decide if I want to attempt its quiz.<br>• When I select a houseplant, I want to see its species name and my mastery for that species. |

| | |
|---|---|
| 2. | I want a series of multiple-choice questions.<br>• I want to be presented with one question at a time, so I can focus.<br>• I want to view a list of possible answers and select one.<br>• I want to be told whether my answer was correct, so I can learn from it.<br>• If my answer was correct, I want to move on to the next question.<br>• If my answer was incorrect, I want the quiz to end, so I can review the learning material. |
| 3. | I want my **mastery** to increase if I pass the quiz, so I stay engaged and I'm encouraged to attempt more quizzes.<br>• I want the plant to grow, so there's tangible proof of my mastery. |
| I want information about the houseplants in my collection, so I can pass quizzes. | |
| 1. | I want to choose a species of plant in my collection. |
| 2. | I want the oracle to give me information about that species.<br>• I want a multimedia presentation, e.g., text and pictures, so it's memorable.<br>• I want the important/quizzable information to be emphasised, e.g., in bold, so I know what to commit to memory. |
| 3. | I want to review information I've already seen, so I can re-attempt quizzes. |

Epic 2 As a player, I want to be reminded when my real-world houseplants need watering.

| | |
|---|---|
| I want to add my real houseplants to the Hub by linking them to an in-game plant. | |
| 1. | I want a separate section of the **hub** which holds real houseplants, so I can clearly see whether a houseplant is 'real' or simulated. |
| 2. | I want to choose from a list of sprites to represent my real houseplant, so it's quick and easy to see which is which. |
| 3. | I want to input names for my real houseplants, in case I forget which is which. |
| 4. | I want to input how often the plant needs watering and feeding. |
| 5. | I want to be able to edit all of this information at any time, in case I realise later that I entered it wrong. |
| 6. | I want to delete houseplants I don't have any more. |
| I want to see a timer above each 'real' houseplant that counts down to its next watering and feeding. | |
| 1. | I want push notifications when it's time to water or feed my plant (might be beyond our abilities) |
| I want to tell the game I've watered/fed my plant, so it will stop reminding me and start counting down to the next watering/feeding. | |

Epic 3 As a player, I want to gain rewards as I progress through the game.

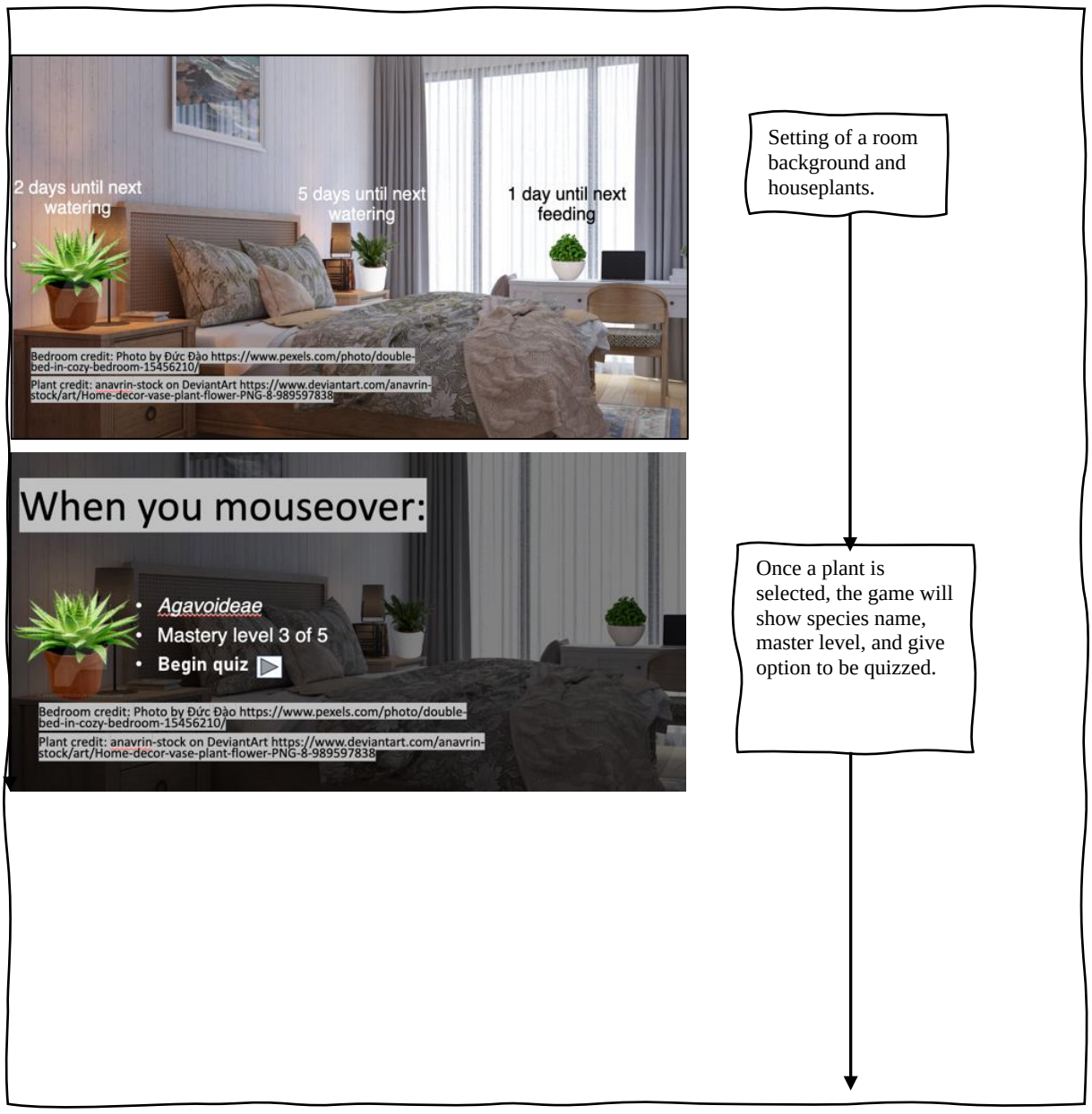
| | |
|---|---|
| A. I want to earn coins as I progress through the game. | |
| 1. | I want to earn a certain number of in-game coins each time I complete a quiz. |
| 2. | I want to earn a certain number of coins each time a water/feed an in-game plant. |

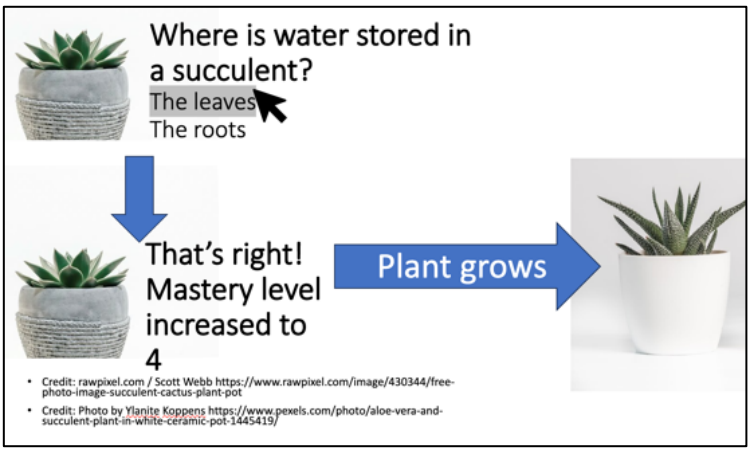
| | |
|---|---|
| 3. | I want to earn a certain number of coins each time I water/feed my real-life plant according to push notifications and informed the oracle I have done so. |
| B. I want to unlock new features of the game. | |
| 1. | I want to be able to unlock new backgrounds (rooms) as I advance through the game. |
| 2. | I want to purchase new "skins" in a virtual shop with the "coins" I earned. |
| C. I want to receive a physical plant as a reward from the game producers after I have reached a certain level. [ABORTED] | |

Epic 4 As a player, I want to connect with friends when playing the game. [ON HOLD]

### 3.3.6. Storyboards

High-level tentative storyboards for core functionalities are created as part of Sprint 3.






Quizzes are sets of multiple-choice questions, if the player correctly answers questions, the plant will grow.

*Link to full presentation: [2023-11-04 Demo Slides.pptx]*

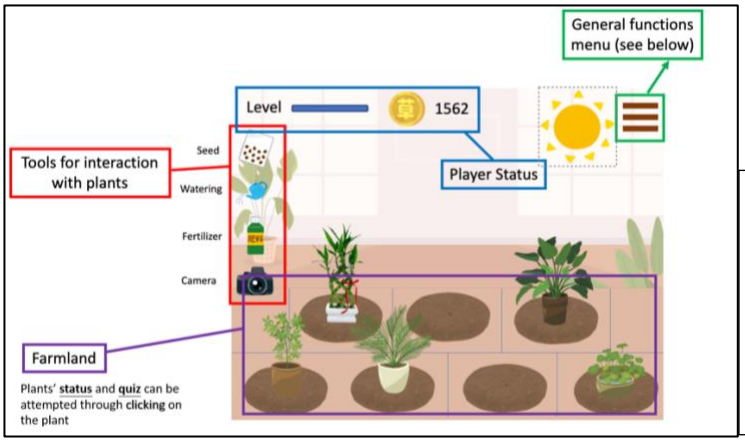
### 3.3.7. Tests

N/A – Testing not started as no acceptance criteria has yet been set for user stories.

### 3.3.8. UI Design

Tentative versions of UI designs produced in Sprint 3 are shown below.

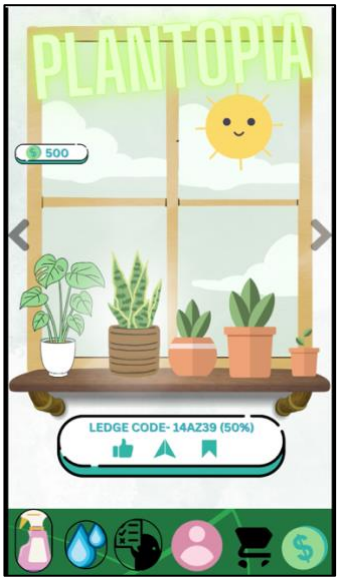
#### *3.3.8.1. UI Design version 1: Main Interface*




Explanations: Indoor setting, plant space in the centre to display plants. Plant interaction menu to water, fertilise, etc. The oracle (sun) is placed next to a collapsed menu bar e.g., achievements, collections, shops, player icon showing player details.


*Link to full presentation: [2023-11-04 Demo Slides.pptx]*

#### *3.3.8.2. UI Design version 2: Main Interface Video Demo*




*Link to full video: [2023-11-04 Demo Video.mp4]*

#### *3.3.8.3. Game Assets: Plant Examples*


*Link to all assets (mixed with additional generations in Sprint 4): [Plants]*

### 3.3.9. CRC Cards

N/A – No CRC cards are deemed relevant for this sprint.

### 3.3.10. Team Stand-up

#### 3.3.10.1. Team Stand-up

| | |
|---|---|
| Date and Time | 6th Nov 2023, 12:00 noon |
| Location | Microsoft Teams |
| Attendees | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |
| Meeting | **Game mechanics development:** |

**Alec** created storyboards for core game features: main screen plant display, mouseover effect, quiz mechanics (*Storyboards*).

**Eddie** built basic UI designs presenting layout of the main screen and key functionalities (*UI Design Version 1: Main Interface*).

**Udit** made a video demo using Canva, suggesting that the in-game "sun" can take the role of the oracle (*Main Interface Video Demo*).

**Marat** generated some game assets of plants and backgrounds using Dall-E from OpenAI (*Game Assets: Plant Examples*).

Everyone agrees with the logic and concepts presented by Alec, Eddie, and Udit. Everyone is very happy with art styles of Mara's assets.

**Game features discussion:**

**Aborted feature:**

**Partnering with companies:** Too ambitious to achieve for a 6-week student project as it most likely requires business negotiation and legal contracting, can be something to experiment with after submission (*Epic 3-C*).

**Features on hold:**

**Sharable code to find friends:** A nice-to-have but challenging to implement (*Epic 4*).

**Features confirmed:**

**Earning coins to buy skins from virtual shop:** Important for game continuity, but still requires thought on what skins to include to keep the game realistic. For example, a pink cactus may not exist in real life hence may not be suitable to sell in shop (*Epic 3-A*).

**Game name ideas:** BotanIQ, Bloom, GroWise, Xylem, PlantMe, Inflorescence. After voting in WhatsApp chat there is a tie between BotanIQ and Bloom.

**Skills audit table results and subsequent decisions:**

| Name | Programming Langua | Other Technical Skills - Visualisation Software, Databases, etc | Work Experiences | Background | ROLE PREFERENCE |
|---|---|---|---|---|---|
| Claire | Python, Java, JavaScript (rusty), HTML, CSS | Git, VBA, SQL, SQLite | Data Analyst | Political Economy / Finance | Professional experience was a mix of model dev and documentation. Generally better at documentation (despite how much i hate it) than coding. Would be happy to help with either. |
| Ivan | Python, C++, C | Bash/Shell Scripting, Photoshop, Git | | Economics/ Finance | Prefer to deal with coding and happy to learn a new language. Could also help with documentation. |
| Eddie | Python, C++, VBA | VBA, CAD | Traffic engineer | Transportation Engineering | Anything but documentation (i mean real typing cause shit in english) |
| Udit | JavaScript(ES6 or React), HTML, CSS, C,C++ | Git,Jira,SQL(Beginner),Frontend design | Software engineer,QA tester and cloud security(temporary) | Information Technology | Bend of both Coding and research work |
| Alec | Python, C# | Unity, Blender | | Archaeology | Coding |
| Favour | JavaScript(ES6 or React), HTML, CSS, Python, C, SQL | Bash/Shell Scripting, VBA, MongoDB, Photoshop, Indesign, Illustrator, Git, Unreal Engine, Twinmotion | Data Analyst, Data Science Intern, and Data Researcher | Architecture/Design | Would prefer to oversee documentation but would also like to engage in a bit of coding. |
| Marat | Java | | Auditor, Enterpreneur | Economics/ Finance, Agriculture | Prefer to code, but can do whatever is needed. If we choose Unity for the game, C# syntax is similar to Java. |

**Selected game development platform and package:**

Mobile phone – as the purpose of the game is to assist players with daily plant care, the mobile platform is easily accessible to players at any point of time, enabling push notifications reminding player to water plants.

C# - 2/7 members have experience, everyone else happy to learn a new language.

Unity – much online resources and templates available.

| | |
|---|---|
| | **Role assignment:**<br>Consideration:<br>• Diverse skillset across the team, with limited specialisation in either software development or documentation.<br>• Short timeline.<br>• Flat team structure (no "manager – junior" hierarchy)<br>• People should not feel pigeon-holed into doing certain task due to limitations in their existing skillsets but should instead see the project as a learning process.<br>Decision:<br>• Two simultaneous workstreams: Code and documentation<br>• In charge of documentation workstream – Favour and Claire<br>• In charge of code development workstream – Alec and Ivan (both with C# experience)<br>• Documentation stream includes development of both product and process documentation.<br>• Coding stream includes coding along with any other technical aspects of game development.<br>• Everyone else will work under one or more workstreams depending on interests and project needs at different phases.<br>**Other process design:**<br>• Daily communications and updates in WhatsApp group.<br>• Bi-weekly stand-ups on every Monday and Thursday.<br>• During stand-ups everyone to go through what they have completed and explain their approach, other members should give comments and feedback.<br>• All finalised versions should be peer reviewed.<br>• All files save in Teams channel.<br>• Backlog as primary means of task records and assignments.<br>• Each sprint set at length of 1 week, starting and ending at each Wednesday customer meeting.<br>• Everyone to do their share of admin where possible, e.g., organising folders and files. |
| Task allocations | **Claire** to set up process documentation structure.<br>**Favour** to set up versioned backlog structure for sprint records. The backlog should be divided into Product Backlog and Sprint Backlog, allow the team to set task due dates, priorities, and make allocations. The backlog for current sprint should be reverse engineered according to meeting records.<br>**Favour** to set up bi-weekly sprint stand-up on Mondays and Thursdays.<br>**Claire** and **Favour** to take over user story development from Sprint 4 onwards.<br>**Alec, Udit, Eddie** to liaise and combine storyboards, UI, and video demo into presentation for next customer meeting.<br>**Alec** to make a tentative list of basic user stories (*User Story List Version 1*). |

<table>
<tr><td></td><td>Ivan to research portrait template in Unity for building mobile applications.<br><br>Eddie to develop basic drawings of the UI design into a tree diagram to visualise the “flows” of the game (UI Game Flow Tree).<br><br>Marat to continue generating assets for plants and game backgrounds.</td></tr>
<tr><td>Meeting reflections and analysis</td><td>Analysis / reflections of current standpoint:<br><br>Although the initial plan for up until this meeting was for everyone to brainstorm rather than deliver, it is a pleasant surprise that half the team (Alec, Eddie, Udit, and Marat) all made a head start with building materials for the next customer meeting. This has certainly helped to set a collaborative and “egoless” team environment for the project where everyone is willing to actively contribute wherever they can. While deciding coding platforms and role allocations, it was highly notable that all members demonstrated the willingness to step out of their comfort zones to take on responsibilities, as shown in the “Roles Preference” column in the skills audit table.<br><br>Moreover, it was clear in the meeting discussion that the team was aware of the importance of setting up necessary processes and practices as early as possible (e.g., bi-weekly stand-ups, backlogs, role assignment and reporting lines, communication channels, etc.). Although this list may not have had an extensive coverage, it certainly reflects the spirit of agile thinking, and the attention to implementing mechanisms that lubricates fast and frequent delivery iterations.<br><br>Team was quick to acknowledge the factors that prevent by-the-book implementation of the Scrum methodology, namely the distinction between a team of seven premature developers studying a conversion MSc versus a professional team of full-time software engineers with industrial experience. Below is a summary of the limitations and the explicit and implicit approaches taken so far to address them.<br><br>
<table>
<tr><th>Limitation</th><th>Solution</th></tr>
<tr><td>Short delivery timeline (6 weeks until due date) making conventional Scrum schedules impractical.</td><td>Setting the length of iteration (sprints) at one week rather than conventional practice of 2 weeks or more.</td></tr>
<tr><td>Concurring commitments to other course modules and personal life, rather than mandatory 9-5 working hours.</td><td>Scheduling bi-weekly stand-ups rather than daily as there no guarantee that all members can spend time on the project every single day. This is to ensure that everyone would have some time to work on their tasks before touching base.</td></tr>
<tr><td>A mix of members with different backgrounds and very limited software engineering experience, rather than a team of mature engineers with specialised skillsets.</td><td>Blurring the boundaries between individual responsibilities by having two workstreams (documentation and code development) rather than distinguished roles. Each workstream have two people co-leading the work for quality assurance, but members have flexibility to contribute to more than one area, depending on the skills required and individual interest/capacity.</td></tr>
<tr><td>Absence of a hierarchical management reporting line which is inherent to the corporate structure, hence the absence of mentorship and management oversight.</td><td>Everyone taking turns to manage team administration: organising directories, following up on individual progress, etc.<br><br>Everyone peer-reviewing everyone’s work.<br><br>Workstream leads overseeing the progress of their streams and raising any issues in team meetings.</td></tr>
</table>
</td></tr>
</table>

| | |
|---|---|
| | **Priorities until next customer meeting:**<br>No new priorities raised. Everyone to work on their assigned tasks. |

### 3.3.11. Sprint 3 Backlog and Completion Status

| TASK | DUE | PRIORITY | ASSIGNED TO | STATUS |
|---|---|---|---|---|
| Complete team skills audit table. | 6 Nov | High | All | Complete |
| Individually brainstorm ideas on product design | 6 Nov | High | All | Complete |
| Set up Git repository for the game, invite all other members as collaborators | 8 Nov | High | Favour | Complete |
| Set up product and sprint backlog files | 8 Nov | High | Favour | Complete |
| Set up template structure on Word for Sprint style documentation | 8 Nov | High | Claire | Complete |
| Find templates / tutorials for working with C# and Unity for mobile game development.[1] | 8 Nov | High | Ivan | Pending |
| Generate drawings and diagrams of initial interface ideas.[2] | 8 Nov | High | Eddie | Pending |
| Begin user story development to begin feature development. | 8 Nov | High | Alec | Complete |
| Develop a demo presentation and demo video for initial interface ideas. | 8 Nov | High | Udit | Complete |
| Continue game asset generation | 8 Nov | High | Marat | Complete |
| Begin to research and develop ideas regarding different plants and difficulty for taking care of them. | 8 Nov | Medium | Marat | Complete |
| Begin to research and develop trivia questions for the various plant difficulties. | 8 Nov | Medium | Marat | Complete |

**Notes to completion status:**

1. No suitable Unity template was found for the plant game design.
2. Eddie was ill throughout the week, hence delivery delayed to Thursday 8 November.

### 3.3.12. Exception Handling

No exceptions were raised in Sprint 3.

### 3.3.13. Individual Retrospectives

Each member reflected on their experience during Sprint 3 by completing the individual weekly retrospective template created in Sprint 4, key points are summarised as follows.

First, there is unanimous agreement the project is off to a good start on multiple fronts. All members express satisfaction with the game idea and shares a sense of relief that the customer is onboard with the conceptual design. Furthermore, several reflections have noted the welcoming, collaborative environment the team has cultivated so far, commending teammates on their enthusiasm and inclusivity.

Second, despite a promising game idea, concerns are present around unknown challenges yet to come in the game-building phase. The difficulty of learning the Unity game engine was recognised as potentially an underestimated obstacle to future progress. This trend seems to align with the team's MBTI composition (Team composition table) in which there is a remarkable predominance of the "Observant" element, often associated with practical and realistic working styles. A few members have also suggested that although informative presentation materials have been built for the next customer meeting, there remains substantial uncertainty on many aspects of the game. For instance, the team has not yet reached an agreement on user interface design, with several versions built into the presentation. Meanwhile, the storyboards and demo only cover the core functionalities of the game (learning about the plants through quizzes), leaving other features (such as link to real life plants) unexplained.

Overall, contribution for Sprint 3 averaged 4 hours across all members with low variance. The table below summarises how everyone is feeling on a scale of 1 to 5 (1= Worst, 5= Best) regarding this Sprint as well as the project overall. The attitude is generally positive but with reservations with all members, accurately reflecting the summaries above.

| Band | Sprint 3 | Project |
|---|---|---|
| 5 (Best) | - | - |
| 4 | 3 | 2 |
| 3 | 4 | 5 |
| 2 | - | - |
| 1 (Worst) | - | - |

## 3.4. Sprint 4: 8th November – 15th November

### 3.4.1. Overview

In Sprint 4, the focus shifts from groundwork to crafting a prototype that showcases the game's key functionalities. In particular, the customer anticipates a stripped-down version of the game with a functioning code base by the end of the sprint. Since most of the team is new to C# and Unity, the priority for this sprint should be for everyone to install the technical software and start learning the tools. Alec and Ivan, co-leading the development stream, are assigned to initiate development of the user interface, as well as continue identifying useful resources that can help accelerate the team's learning curve for Unity.

Simultaneously, in preparation for code development, the team also needs to organise and articulate conceptualised ideas into structured product flows and user interfaces. This entails refining user stories and generate additional game assets. On the documentation front, the team aims to formulate a plan for effective record-keeping throughout the development timeline. This responsibility is assigned to Favour and Claire who are co-leading the documentation workstream. Although there is an early consensus that delivering a functional prototype of the game by end of Sprint 4 may be ambitious (with Alec being the only one who's familiar with Unity), everyone nonetheless commits to trying their best and to reassess progress near the end of the week.

### 3.4.2. Review

As Sprint 4 coincided with the two mid-term deadlines (CM50258 and CM50259), several members of the team (especially those new to computer science) were struggling to balance the workload. As a result, progress was sluggish throughout the week on the coding front, and no one was able to learn enough of Unity to assist Alec with interface and prototype creation. By end of Sprint 4, it was clear that the team is falling behind, and delivering a prototype to the customer by start of Sprint 4 was unachievable. The challenge of learning Unity with piecemeal time has escalated to become a substantial obstacle to game development, exacerbated by the truth that no one has any experience with professional game development engines whatsoever.

Aside from the challenges, the team continued to generate a broader set of game assets, encompassing plant images at different levels, game backgrounds, game oracle images, and game name icon design. A game flow tree was also created for visualising the main game flow, complemented by UI wireframes. Furthermore, Favour and Claire were able to set up a weekly reflection template containing a list of questions for everyone to complete by end of every sprint to recording individual efforts, contributions, and retrospective insights throughout the project. Several key user stories representing the game's basic functionalities were also developed in detail to showcase the nitty-gritties of core game mechanics.

### 3.4.3. Summary of Key Events

- Prototype development kick-off – Alex and Ivan building stripped-down UI and everyone else beginning to learn Unity.
- Completion of all the key assets required to kick-start development.
- Completion of a game-flow tree to visualise core functionalities.
- Incorporation of weekly individual reflections into team processes.
- Developed subset of user stories for main screen game components.
- Unsatisfactory progress as most peers is struggling with overlapping deadlines.
- GitHub repository created for collaboration; Unity demo uploaded.

- "Bloom" decided as the final game name.

### 3.4.4. Sprint 4 Preparation: Customer Meeting and Planning Discussion

| | |
|---|---|
| **Date** | 8th Nov 2023 |
| **Location** | Campus – Chancellor Building 4.7 |
| **Attendees** | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |
| **Customer Meeting** | **Presentations:**<br>*"*" indicating that customer has made a comment, recorded in the "Customer concerns" section below.*<br>**Storyboard*:** To demonstrate game mechanics only, not a real representative of how the game would look like (*Storyboards*).<br>**Main Interface Design:** See *UI Design Version 1: Main Interface*.<br>**Video demo:** See *Main Interface Video Demo*.<br>**User stories*:** See *User Story List Version 1*.<br>**Game name*:** Currently a tie between "BotanIQ" and "Bloom".<br>**Choice of platform:**<br>**Mobile Phone:** Allow push notifications when a plant requires attention (water / fertilise)<br>**Unity with C#:** Mature game engine with plenty of readily available resources and templates for developer support*.<br>**Customer concerns:**<br>• Customer advice to be careful with version control as Unity does not have built-in Git. A video will be shared on how manually link Unity with GitHub.<br>• Customer requests more detail on user stories, in particular acceptance criteria.<br>• Customer overall happy with the preparation work, noting that the concept of the game is a lot clearer than last week.<br>• Customer requires more detail on the functionalities and features at each stage of the game, e.g., how will the oracle interact with the player, how to quantify a player's mastery level, etc. **[A1]**<br>• Two customers each added vote to game name poll, yet still a tie between "BotanIQ" and "Bloom". |
| **Planning Discussion** | **Additions/changes to product design based on customer meeting:**<br>**[A1]** To add granularity user stories.<br>**Analysis / reflections of current standpoint:**<br>Customer feedback has so far stayed positive, implying that work is on the right track. Nonetheless, the team is on the verge of entering a danger zone as several obstacles are about to emerge. First, game development has not yet started, and it will soon be announced on the forum the requirement to deliver a prototype by end of Sprint 4. Second, everyone except Alec is entirely unfamiliar with the development toolkit chosen, and there is a great deal of uncertainty around how long it takes people master enough Unity to start developing the game. Third, over the past few weeks most members in the team have been focused exclusively on delivering CW1 of this module, meaning that progress for other modules have been put on hold. Now with CW1 submitted, many members are rushing to |

| | |
|---|---|
| | complete other coursework deadlines, which makes learning Unity from scratch in addition to everything else will be excruciatingly challenging.<br><br>As a result, the team should be aware that if appropriate measures are not taken, progress can easily go south in the upcoming weeks.<br><br>**Determining Sprint priorities:**<br><br>Mindful of the remaining time until submission, the team determined that Sprint 4 should transition into active game development following the prior weeks of preparatory effort. Ivan raised in the planning discussion that there doesn't seem to be any existing Unity templates suitable for the game. The news brought about an overwhelming sense of urgency, flagging out the possibility that everything will need to be coded from scratch. Ivan also shared some Unity training videos he has been going following, bringing everyone aware of the steep learning curve for those with little knowledge of game engines.<br><br>As the concurring theme in customer feedback during this week's meetings was to crystalise game features and mechanics, another task for Sprint 4 was to refine the works already started in Sprint 1, i.e., asset creation, user story development, UI development, storyboard refinement, video prototype refinement. Meanwhile, at the point of the meeting, the team was not fully aware of the deliverables for the next customer meeting (which was published the next day to include a functional stripped-down game prototype). Therefore, when making the initial backlog on 8th November, it was hard to say whether everyone was fully aware of the pressing need to start coding. Hence, the Sprint 4 backlog as of Wednesday 8th November did not have an obvious focus on coding game prototype.<br><br>**The Sprint 4 backlog can be found in Section 3.4.11.** |

### 3.4.5. Development of User Stories, Acceptance Criteria, Use Cases

Based on the tentative user story list compiled in Sprint 3, the team began finalising user stories for game development. User stories that were deemed relevant to the stripped-down prototype were listed and substantiated in detail, incorporating use cases, tests, and UI design. However, as the prototype was built in Sprint 5 rather than Sprint 4, the corresponding user stories are all presented as part of the Sprint 5 documentation.

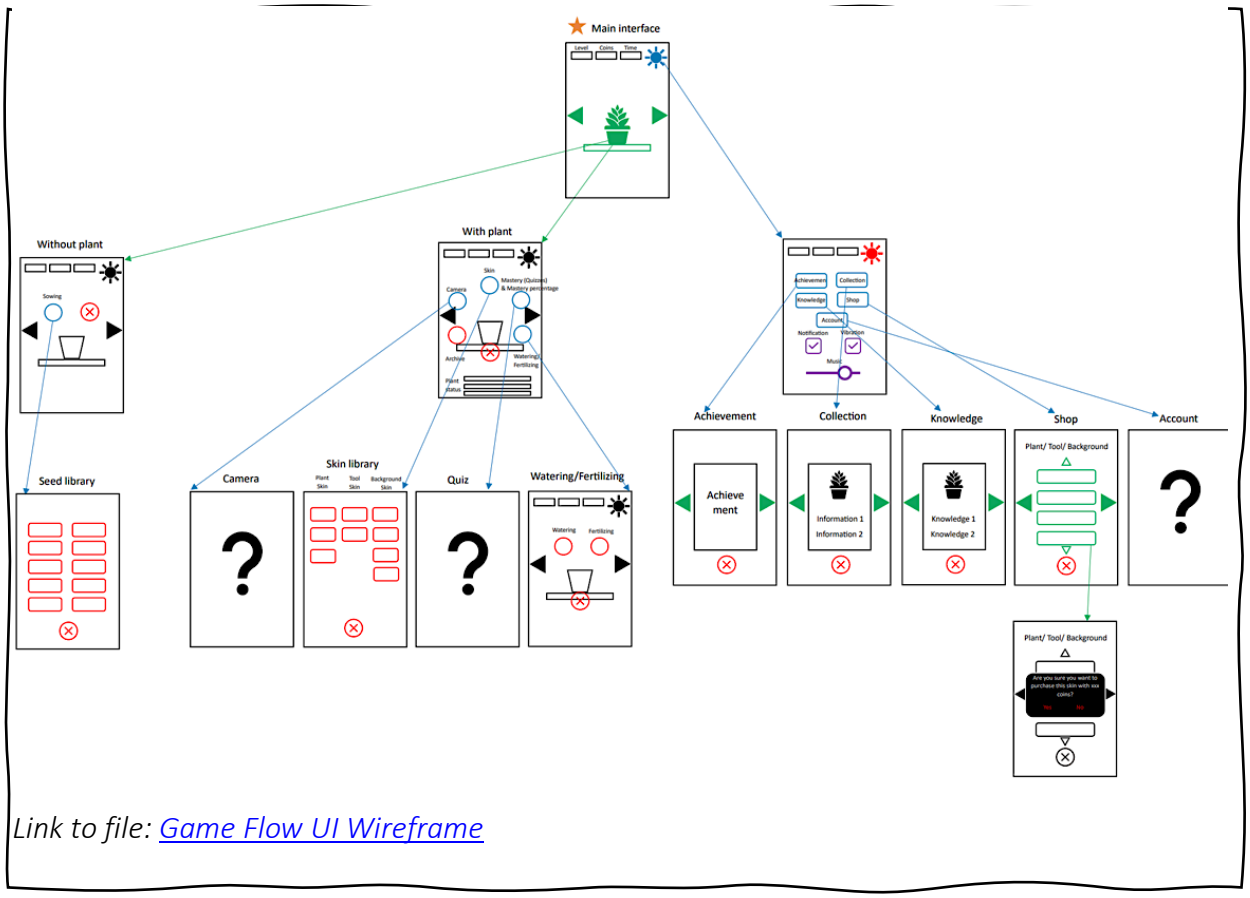
### 3.4.6. Tests

As no prototype was built over Sprint 4, no tests are deemed relevant to this week's tasks.

### 3.4.7. UI Design

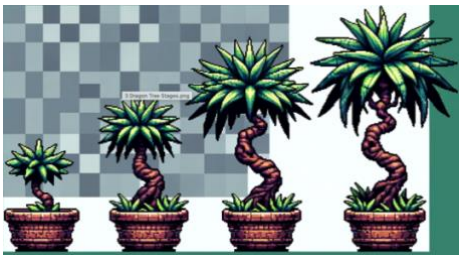
#### *3.4.7.1. Game Screens Wireframes*

UI wireframes presented in the game flow tree-diagram is shown below.




*Link to file:* [Game Flow UI Wireframe]

### *3.4.7.2. Game Assets: Plants and Backgrounds (examples shown below)*

| | |
|---|---|
| Dragon Tree |  |
| Aloe Vera |  |
| Olive Tree |  |

| | |
|---|---|
| Background |  |
| Background |  |

*Link to all assets (mixed with previous generations in Sprint 3): [Assets]*

### 3.4.8. CRC Cards

N/A – No CRC cards are deemed relevant for this sprint.

### 3.4.9. Team Stand-up

#### 3.4.9.1. Team Stand-up 1

| | |
|---|---|
| Date and Time | 9th Nov 2023, 12:00 noon |
| Location | Microsoft Teams |
| Attendees | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |

| | |
|---|---|
| Meeting | **Updates on task-wise progress:**<br>**Eddie** went through in detail the game process flowchart which outlines a conceptualised UI design for each screen (Game Flow Tree for UI). The camera and quiz functionalities are left as question marks as not yet sure of their exact format and complexity. The team is unsure whether to implement an account feature as it may potentially be beyond the team's current level of expertise.<br>**Alec** showed the current stage of the interface in Unity. Eddie's flowchart will be incorporated into prototype development.<br>**Marat** generated more plant assets representing four stages for each plant. Everyone happy with the art style and agree that the variety of the plant is now sufficient for development. More backgrounds pictures are needed to align with the plant style and perspective.<br>**Game features discussion:**<br>The team carried out a comprehensive discussion on the mechanics between in-game plants and real-life plants. Previously the assumption was that real-life plants added to the game flow will be shown separate to in-game plants, and for a player the task is to both keep the in-game plants alive by virtually "watering" or "fertilising" them and to water their real plants according to timed push notifications received from the app. Some of the team was unsure of the need to care for in-game plants at all (by virtual "watering" and "fertilising"), if the purpose of the game is to help players take care of their real-life plants. The final decision is to have two simultaneous threads:<br>(1) In-game plants for which the user can take quizzes and achieve mastery.<br>(2) Instead of keeping all real-life plants in a separate section, the user can link an in-game plant with a real-life plant at any point in the game.<br>In addition:<br>(1) For linked plants the app will give push notifications when it needs care.<br>(2) In-game plants will not generate any push notifications unless linked to a real plant.<br>Moreover, the team has determined that in the shop feature, only skins representing real-life scientific variations of the plant will be sold, i.e., no more pink cactus.<br>**New process change:**<br>The team decides that the Thursday stand-up will be moved to Friday from next week onwards, hence creating longer intervals between stand-ups for members to work on their task allocations. Wednesdays will be both a retrospective meeting for the previous print and the planning meeting for the next print.<br>**Documentation update:**<br>Claire and Favour to arrange a separate chat to align on documentation structure and task division.<br>**Goal setting (entering danger zone):**<br>New announcement in course forum expects by the next customer meeting a basic working game prototype with start, middle and end. The announcement brought heightened sense of urgency within the team had barely started to code and is not yet even in a position to divide up code work. Everyone is forced to acknowledge that the goal itself is beyond the team's capacity for this sprint as everyone is just about to start learning Unity and Alec so far being the only person involved in game development. While Alec continues to develop the coded demo, everyone else should start making time for Unity as soon as possible. The team will touch base again in the next stand-up on Monday to discuss deliverables for the customer meeting. |
| Task allocations | **Eddie** to share flowchart on Teams for **Alec** and **Ivan** to incorporate into demo.<br>**Marat** to try generating alternative backgrounds in the same style as potted plants. |

| | |
|---|---|
| | **Favour** to create GitHub repository and **Alec** to upload demo file to repository.<br>**Claire** and **Favour** to liaise on documentation structure, and to update the team on Monday.<br>**Everyone** to download and start learning Unity. |
| Meeting reflections and analysis | **Analysis / reflections of current standpoint:**<br>While this meeting overrun the 1-hour time slot by 30 minutes, it was a productive discussion as the team gained clarity on the main features of the game (better late than never). Specifically, the team reached an agreement on the relationship between two core functionalities, achieving mastery by completing quizzes, and linking a plant to a real-life plant to give the user notifications on when to water their plants.<br>With the customer requesting a basic game by Wednesday, it is by now inevitable that the team would be failing expectations for Sprint 4. Although mastering a new tool is not a hefty task per se, the 6-week timeframe of the project suggests spending so much as one week on Unity training could lead to disastrous consequences. It could have been a good idea to start thinking of plan Bs at this stage, or even as early as when Ivan pointed out the absence of applicable Unity templates (as the access to templates was the primary reason why Unity was chosen in the first place). However, the team at this point still held an unjustified positivity that the plan would eventually work out when people have had some time to solely focus on this project after completing all mid-term assignments.<br>**Priorities until next customer meeting:**<br>Everyone to use all available time for the project on learning Unity. |

### 3.4.9.2. Document Alignment Meeting

| | |
|---|---|
| Date and Time | 11th Nov 2023, 11AM |
| Location | Microsoft Teams |
| Attendees | Claire, Favour |
| Meeting | **Key points discussed:**<br>**Process document:** Record keeping is essential to documenting process throughout a 6-week project.<br>All team meetings should be recorded.<br>**Claire** will set up a weekly reflection template for everyone to record their contributions as well as any reflections for each sprint.<br>**Favour** is happy with Claire's process document structure.<br>**Claire** will start transcribing meeting minutes for past meetings.<br>**Question for TA:** If the markdown template saved under Coursework 1 on Moodle page is sufficient for the process document.<br>**Product document:** May not be able to start as the product is not yet of tangible form.<br>**Installation guide and video:** To be started when majority of development work is complete. |
| Post-meeting reflections and analysis | **Analysis of current standpoint:**<br>This meeting has given both oversights of the documentation stream a better understanding of what needs to be done throughout the project to deliver on the documentation side. Nonetheless, this meeting should have been scheduled way earlier than Sprint 4, i.e., at least before the start of Sprint 3. The reality was that everyone had jumped into the project in a rush without full awareness of the procedures, and only after everyone had submitted the Coursework 1 report had the team been able to pause and |

| | |
|---|---|
| | think if the necessary processes are set in place.<br>In retrospect, a notable failure of this meeting is the negligence towards product documentation. Both Favour and Claire took to the assumption that product documentation was something to discuss further down the road when the product has taken form, rather than an innate part of the project that requires consistent effort from the beginning. The impact materialised in Sprint 6 when everyone realised that much of what is required as part of product documentation was missing and had to subsequently embark on a week of reverse engineering versioned components. |

### 3.4.9.3. Team Stand-up 2

| | |
|---|---|
| Date and Time | 13th Nov 2023, 12:00 noon |
| Location | Microsoft Teams |
| Attendees | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |
| Meeting | **Updates on task-wise progress:**<br>Claire created a Sprint reflection template for everyone on Teams. The purpose of this template is to keep record of individual contributions and retrospectives for every sprint. It is advised that no more than 20 minutes is spent on populating each template with simple bullet points. Template questions are presented below:<br>• Sprint Number.<br>• Date.<br>• Name.<br>• Tasks assigned to you for this sprint.<br>• Which ones did you complete?<br>• Were there any particular causes that you think helped you in completing them?<br>• *e.g., support from a peer, an online forum with handy resources, good time-management and planning, etc.*<br>• Which ones were you unable to complete?<br>• Were there any particular causes that you think prevented you from completing them?<br>• e.g., Heavy coursework load from another courses, sickness, friction with a teammate, or any other circumstances, etc.<br>• Estimated number of hours spent on the project, including team stand-ups (2.5h total)<br>• Overall, what do you think went well this week? (could be from an individual or team point of view, or both)<br>• Overall, what do you think didn't go so well this week? (could be from an individual or team point of view, or both)<br>• Any thoughts on the collaborative teamworking process? –<br>• e.g., practices and processes, technical tools, communication with teammates, and *anything else you would like to highlight in the report.*<br>• On a scale of 1-5 (1 = worst, 5 = best) how are you feeling about your experience of this sprint?<br>• Any elaborations you want to add to your rating above? |

| | |
|---|---|
| | • On a scale of 1-5 (1 = worst, 5 = best) how are you feeling about the progress of the project so far?<br>• Any elaborations you want to add to your rating above?<br>*Link to template file: Individual Weekly Reflection Template*<br>Marat generated more game assets for the Oracle (the sun) and new game backgrounds. Background number 9 is voted by far the most suitable with a consistent left to right ledger (which helps with building the infinite scrolling function), and absence of excessive light and shadow. However, this means asset ratio and perspective for potted plants may need to be re-adjusted, as the image is a 90-degrees perpendicular view into the screen.<br>**User story development:** Claire developed some user stories for basic functionalities that should be included in game prototype.<br>**Code development:** Favour created GitHub page for the project and linked Unity. Everyone has downloaded Unity, and several people has tried running the demo locally. Ivan has shared some Unity tutorials in the WhatsApp group. However, attempts to learn Unity while completing other coursework has been challenging for most people, especially as members were generally unfamiliar with the intuitions behind the mechanics of game development engines. Alec has agreed to run through the main functions of Unity on Wednesday's lab.<br>**Working demo:** As no one has yet been able to help Alec with code development, the demo is still in elementary form with various elements not working as expected. The team is inclined towards scrapping the idea of presenting the working demo to the customer on Wednesday 15th. Since most people do not expect to have much capacity until after the CM50258 and CM50259 due dates, dividing up coding tasks is subsequently postposed to the next sprint, when hopefully everyone would have had some time to explore Unity.<br>**Game name:** One member changed their vote from BotanIQ to Bloom, which makes Bloom the winner. |
| Task allocations | **Everyone** to complete reflection template for Sprint 1 and Sprint 4 by Wednesday 15th.<br>**Everyone** to try running demo on Unity.<br>**Everyone** to continue learning Unity.<br>**Marat** to try generating some alternative background assets that aligns with current plant perspective. |
| Meeting reflections and analysis | **Current standpoint:**<br>Many members have expressed the concern that the preparation phase of the game has been dragging out for way too long (four weeks since first team meeting in Week 3 of Semester 1), with not even a bare minimum completed for game development. It is imperative that code development must escalate to be the priority for the next sprint to deliver a complete product by 15th December.<br>The team has entered a state of collective panic.<br>**Priorities until next customer meeting:**<br>Complete CM50258 and CM50259 coursework as soon as possible so the focus can move onto building code. |

### 3.4.10. Exception Handling

| Exception | Actions | Outcome |
|---|---|---|

| | | |
|---|---|---|
| Confusion around how to synchronise in-game plants and real-life plants. Specifically,<br>Where are real-life plants displayed in the game?<br>Does the user have to achieve full mastery before allowed to link a plant?<br>Do unlinked in-game plants need virtual care (i.e., watering and fertilising) and whether they generate push notifications? | The team has agreed that the player will only receive push notifications for plants that have been linked with a real-life plant, since simultaneous notifications from a mix of in-game and real-life plants can become chaotic to the user.<br>Players are allowed to link an in-game plant with a real-life plant at any point in the game, without needing to achieve full mastery.<br>Once players have achieved full mastery for an in-game plant, they can "water" and "fertilise" the in-game plant by selecting the "water" and "fertilise buttons and earn XP / coins. However, only linked plants will generate push notifications reminding players to water their real-life plants. | Success |
| No available ready-to-use Unity templates for the game design. | The main reason for choosing Unity was to take advantage of available templates. This implies that the entire game needs to be coded from scratch, which the team is not yet prepared for. | Pending |
| CM50258 and CM50259 coursework due dates overlapping with Sprint 4. Many members are struggling with overlapping assignments. | As most members come from non-computer science backgrounds, the issue seems to be affecting most if not all members of the team. Everyone is trying their best to make piecemeal time for this project, and it is evident that everyone is still able to deliver on a good proportion of their allocated tasks. However, this still inevitably means that Sprint 4 has put the team behind on progress . All peers have committed to spending significant blocks of time to the project as soon as other deadlines are cleared. | Pending |
| Unity proving to be much more difficult than everyone had previously imagined. | All members have downloaded Unity by end of Sprint 4, yet no one was yet able to become a self-taught Unity user. This is a combined result of overlapping deadlines as well as the difficulty of Unity itself for anyone with no prior experience in game development engines. The immediate plan is for Alec to run through basic functionalities of Unity on Wednesday and to have the whole team dedicate a week to Unity training. | Pending |
| As a result of the three exceptions above, the team has failed to deliver a working game demo as per customer request. | The team recognises the urgency but there is not much anyone can do since no one would be willing compromise on CM50258 or CM50259, both of which are core units in the programme. It was agreed that as soon as competing deadlines are finished, everyone must commit to spending longer periods on this project to compensate for Sprint 4. | Pending |

### 3.4.11. Sprint 4 Backlog and Completion Status

| TASK | DUE | PRIORITY | ASSIGNED TO | STATUS |
|---|---|---|---|---|
| Update user stories with acceptance criteria and use cases (listed in Sprint 5 User Story Section as they were built into the code base in Sprint 5) | 15 Nov | High | Claire | Pending |

| | | | | |
|---|---|---|---|---|
| Update presentation with more accurately annotated interface diagrams. | 15 Nov | High | Eddie | Complete |
| Update demo video with features and styling closer to discussed expectations. | 15 Nov | High | Udit | Pending |
| Development of background image assets and sprites to be used in Unity. | 15 Nov | High | Marat | Complete |
| Learn Unity and C# | 15 Nov | High | All | Pending |
| Build stripped down coded game demo containing core features. | 15 Nov | High | Alec | Pending |
| Divide up coding tasks into individual portions. | 15 Nov | High | Alec | Pending |
| Continue research and development of plant types, and trivia questions for the various plant difficulties, with more questions depending on difficulty. | 15 Nov | Medium | Marat | Complete |

### 3.4.12. Individual Retrospectives

Each member reflected on their experience during Sprint 4 by completing the individual weekly reflection template, key points are summarised as follows.

First, several peers commented on a positive team culture so far established in the team. Communication was open and honest, and everyone was understanding of any individual circumstances, such as sickness and personal commitments, that may have interrupted individual work. Over team meetings, people felt at ease when expressing questions or doubts, and any conflicting opinions were handled with courtesy, respect, and grace. The team has so far adopted the implicit process of giving everyone the chance to raise any thoughts, each time someone presented on a piece of work. If concerns are raised, the team would then have an open discussion to break down the problem and arrive at a solution accepted by all. The inadvertent effect of this is that at every step of the way the team is aligned as a united front, with everyone agreeing with the goals as well as the approaches taken. Consequently, since no one's views are ignored and everyone feels they have a place in the team, an encouraging and selfless team culture hence emerges out of frequent interactions and collaborations.

Second, almost everyone noted that customer seem to enjoy the game's concept, especially after presentation of the story boards and core game mechanics. However, the worry that game development will be an insurmountable challenge in the next few weeks is deeply bothering everyone. As mentioned, the immediate deadlines for courses CM50258 and CM50259 has placed progress on hold due to shortage of time for learning Unity, and many peers are anxious about whether they can master a complex game development engine in a short period of time. For the time being, however, it seems there is nothing much the team was able to except focus on completing other coursework as early as possible to yield extra time for the project.

Overall, contribution for Sprint 4 was evenly distributed across members, averaging at a total of 5.5 hours. The table below summarises how everyone is feeling on a scale from 1 to 5 (1=worst, 5=best) regarding this Sprint as well as the project overall. Everyone is clustered in band 3 and band 4, accurately reflecting the team's current situation combing clearly conceptualised game mechanics with a distressing absence of a prototype code base.

| Band | Sprint 4 | Project |
|---|---|---|

| | | |
|---|---|---|
| 5 (Best) | - | - |
| 4 | 3 | 2 |
| 3 | 4 | 5 |
| 2 | - | - |
| 1 (Worst) | - | - |

## 3.5. Sprint 5: 15th November – 22nd November

### 3.5.1. Overview

Sprint 5 brings drastic changes in team structure and project progress. With Sprint 4 severely falling behind in coding due to various reasons, the primary goal for Sprint 5 was to experiment with changing to JavaScript with React Native, which a few members of the team have prior exposure to. The team planned to confirm the decision by the first sprint Stand-up, and to devise a structured blueprint for the upcoming weeks on how to proceed with development tasks.

In the second half of the sprint, the priority is to move ahead with as much coding completed as possible before the next customer meeting and deliver the product prototype that was due in Sprint 4. The game should be in a state where the skeleton structure is set in place for substantial coding tasks (relating to core game features) to be divided and conquered over the next sprint. On the documentation side, the focus was to continue developing user stories according to the template shown in Week 7's Lab, as well as continue progressing with the process document. By end of this sprint, the team expects everyone to "hit the ground running" and settle into a steady pace of product feature delivery.

### 3.5.2. Review

The team began by experimenting with the platform change to JavaScript with React Native. With much progress made in just a couple days, all quickly agreed that this was for the best and proceeded with the decision. While majority of the team was powering through the last remaining coursework from CM50260, two members of the team who were ahead with it (Favour and Ivan, between which Favour has much prior experience with using React Native), took to building a game skeleton and setting up the React Native framework. By the customer meeting, majority of User Stories on the main game page was completed. Overall, the team was able to get back on track with progress in terms of developing code and meet customer expectations. On documentation, progress was paused as a result of team restructuring (Favour moving from co-leading documentation over to heading up game development) in addition to CM50260 impacting most of the team.

### 3.5.3. Summary of Key Events

- Change of development platform from Unity to JavaScript (React Native).
- Team restructuring – Favour moving from co-leading documentation to leading game development.
- Completion of a stripped-down coded demo of game interface and basic versions of core functionalities.
- Part of the team still struggling with other course deadlines.
- Falling behind with documentation.

### 3.5.4. Sprint Preparation: Customer Meeting and Planning Discussion

| | |
|---|---|
| **Date** | 15th Nov 2023 |
| **Location** | Campus – Chancellor Building 4.7 |
| **Attendees** | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |

<table>
<tr><td>Customer Meeting</td><td>

**Presentations:**

*"*" indicating that customer has made a comment, recorded in the "Customer concerns" section below.*

- The team showed to the customer game interface flow created by Eddie (Game Flow Chart with UI). Highlighting:
  - Interactive start screen (main game screen), demonstrating how the ledges are placed, from which the user can select plants to place in their virtual room*.
  - Option to select from a plant-specific menu, amongst which one option is to go to plant quiz, as well as water, fertilise, etc.
  - Option to select from main menu functionalities from a navigator on main game screen, including achievements, settings, account profile, shop, etc.
- Game assets generated of background, oracle, plants, etc.
- The game prototype right now only has a basic interface, within no functionalities built in.

**Customer concerns:**

- Customer asks to clarify on how to navigate back to main screen from quiz.

  Response: A backward button is present on every screen.
- Customer asks to clarify on whether the interactive start screen is the main gameplay screen or the starting screen.

  Response: Main gameplay screen is also the first screen player lands at after opening the game.
- Customer asks whether any points are needed before adding any plants to collection.
  Response: All plants in the seed library are available to choose from.
- Customer asks for incentive to learn more plants if you cannot buy more plants.
  Response: XPs and coins can be used to buy different skins in the shop for plants in collection.
- Customer asks to clarify how do players learn how to care for a plant.

  Response: Each plant will have a health bar indicating whether they are receiving the necessary care (regularly watered etc.). However unhealthy plants will look "sad" instead of dying.
- Customer asks how players tell apart that plants in imaginary skins are not representative of plants in real life.

  Response: The skins will now be restricted to realistic and scientific variations of the same plant i.e., white monstera.
- Customer asks that a working demo must be delivered next week as this was originally the goal for this week.

</td></tr>
<tr><td>Planning Discussion</td><td>

**Additions/changes to product design based on customer meeting:**

No additions or changes have stemmed from this meeting.

**Analysis / reflections of current standpoint:**

The team agreed that game development progresses is lacking behind and emphasises that next week's focus should be pulling ourselves out of the "danger zone". Unfortunately, several members are still far from completing the CM50260 coursework, which was previously placed on hold as everyone was focused on CM50258 and CM50259. It seems that until 20th November the outlooks are still looking grim for people to head deep into Unity training. The situation has reached a severity where a plan B must be prepared to fulfil

</td></tr>
</table>

| | |
|---|---|
| | the bare minimum coursework requirements.<br>**Determining Sprint priorities:**<br>Two urgent issues were raised when compiling the backlog for Sprint 5.<br>**(1) Switching platforms:**<br>The team discussed the reasons why people have not been able to proceed with code development. While overlapping deadlines was a key factor, the concern was why no one besides Ivan and Alec (who were leading the coding stream) was able to use piecemeal time to make progress. Everyone concluded that another contributing factor is the difficulty of learning to use Unity, which is a monster in and of itself before even starting on any game feature development. After Alec went through a brief introduction of the Unity interface and some functionalities, everyone became aware that even with other coursework deadlines out of the way, there still the need of at least another few days and substantial amount of googling for everyone to become functioning users of Unity.<br>Therefore, several members of the team suggested the possibility of switching platforms to JavaScript and React, of which both Favour and Udit are already proficient users and some other members of the team also have had brief exposures. Both Favour and Udit also agree that getting the game up and running in React would be a relatively quick and easy processes. The plan for the next two days until the first team stand-up on Friday is to research the feasibility of building a mobile game with JavaScript and React.<br>**(2) Hosting on mobile:**<br>So far, the assumption was to build the game as an IOS app, but a concern was raised that hosting an app on the Apple Store requires a licence fee, and until now the installation of the app from Unity to an iPhone has proved to be a cumbersome process. While the easy solution was to move to Android, only one out of seven peers in the team has an android phone, which could make testing a lot harder. The team urgently needs to devise a solution to this problem.<br>**The Sprint 5 backlog can be found in Section 3.5.9.** |

### 3.5.5. Development of User Stories, Acceptance Criteria, Use Cases, UI Design

In Sprint 5, the following user stories have been built into the game prototype.

List of user stories relevant to Sprint 5. Note that parts of the stories below are reverse engineered in Sprint 7 (after the features are already developed into the game) as an unfortunate outcome of the team's failure to sequentially implement the software process model. Therefore, some UI designs were developed without any wireframes as guidance, in which cases a snapshot of the UI in the end-product is presented for illustration.

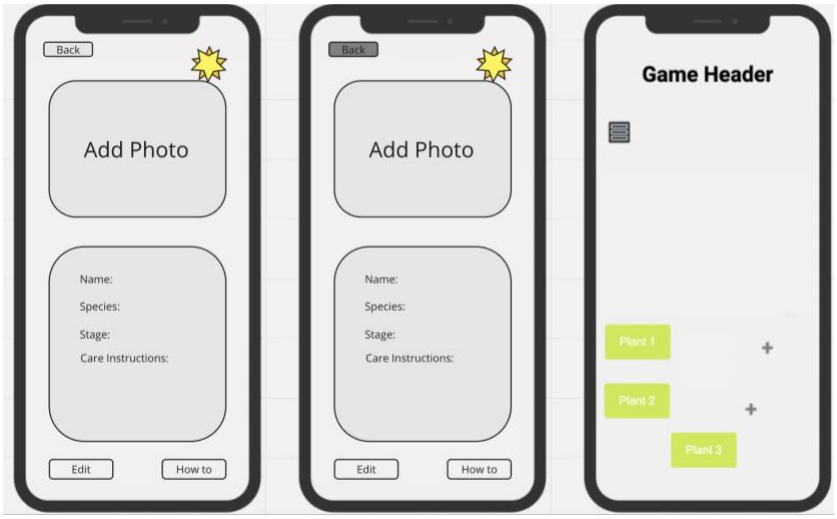
**Full list of user stories relevant to Sprint 5**

*Overwritten versions are greyed out

| User Story ID | Description |
|---|---|
| US1-V1 | Return to previous screen when selecting a "Back" button from any non-main screen. |
| US4-V1 | Displaying main screen elements |
| US5-V1 | Add a plant to a selected placeholder spot on virtual room from the plant inventory. |
| US15-V1 | View plant mastery levels. |
| US17-V1 | Expand plant action menu. |

| | |
|---|---|
| US46-V1 | Expand main game menu. |
| US25-V1 | Level selection screen. |
| US26-V1 | Proceed to next question if a question is answered correctly. |
| US98-V1 | Receiving oracle guidance throughout the quiz experience. |
| US45-V1 | Oracle displaying facial expressions |
| US41-V1 | Game settings screen |
| US42-V1 | Clear game data |
| US37-V1 | Close plant action menu. |

| | | | |
|---|---|---|---|
| US1-V1 | Return to previous screen when selecting a "Back" button from any non-main screen. | | |
| Description | **As a player, I want to have a "back" button for every secondary screen that returns to the previous screen after pressing.** | | |
| Acceptance Criteria | Criteria | Description | Test Status |
| | **US1-V1-AC1.** | I can see the “Back” button on every screen other than the main screen. | Pass |
| | **US1-V1-AC2.** | Once I clicked the button, it will take me back to the previous screen in my game flow | Pass |
| Requirement Use Case | **Use Case:** Returning to the previous screen by pressing the button on the screen.<br>**Scope:** Secondary game screens<br>**Level:** Basic game set-up<br>**Context:**<br>(1) When the player arrives at any screen except the main screen, they can press a button to return to the previous screen in their game flow.<br>(3) The button should be placed somewhere easy to spot.<br>**Frequency of occurrence:** Every time the user wishes to return to previous screen from secondary screens.<br>**Open issues:** N/A | | |
| Design Use Case | **Scope:** Secondary game screens.<br>**Level:** Basic game set-up<br>**Primary actors:** Player<br>**Description:** Player can press the “back” button on the screen to return to the previous screen in their game flow.<br>**Dependencies:**<br>(1) The user has successfully installed the game, and able to run the game on their device.<br>(2) Secondary screens have been built into the game<br>(3) User is has arrived at a secondary screen (non-main screen). | | |

| | |
|---|---|
| | **Assumptions:** There are features that require user to go to secondary screens from the main screen.<br><br>**Preconditions:** Player has pressed any functional buttons that brings the game to secondary screens.<br><br>**Main Flow:** Player can see a “Back” button on the upper left corner of the screen displaying text “Back”. User presses the button if they wish to return to the previous screen. The user successfully returns to the previous screen.<br><br>**Post conditions:** After the player pressing the "back" button on the screen, the game should return to the previous screen in their game flow.<br><br>**Alternative Flows:**<br><br>(1) Valid:<br><br>Player arrives at non-main screen where a back button is correctly displayed but player does not wish to return to previous screen.<br><br>(2) Invalid:<br><br>Player arrives at non-main screen but no "back" button appears.<br><br>Player presses the "back" button but unable to return to previous screen.<br><br>**Frequency of occurrence:** Every time the user is on a secondary screen and wishes to return to previous screen.<br><br>**Open issues:** N/A. |
| UI Design | **Key UI components:** "Back" icon19<br><br>**Wireframe:**<br><br>  <br><br>**Interaction Flow:**<br><br>User clicks on the "back" button.<br><br>Returns to the previous screen in the player’s game flow.<br><br>**Data Entry and validation: N/A**<br><br>**Error Handling: N/A**<br><br>**UI Behaviour:** The "back" button should have a hollow animation when player pressed it.<br><br>**Visual Design Guidelines:**<br><br>The "back button should be in the corner of the screen.<br><br>The button should be on the highest layer of the screen. |

| | |
|---|---|
| | The button should not cover other functional items except the background. |

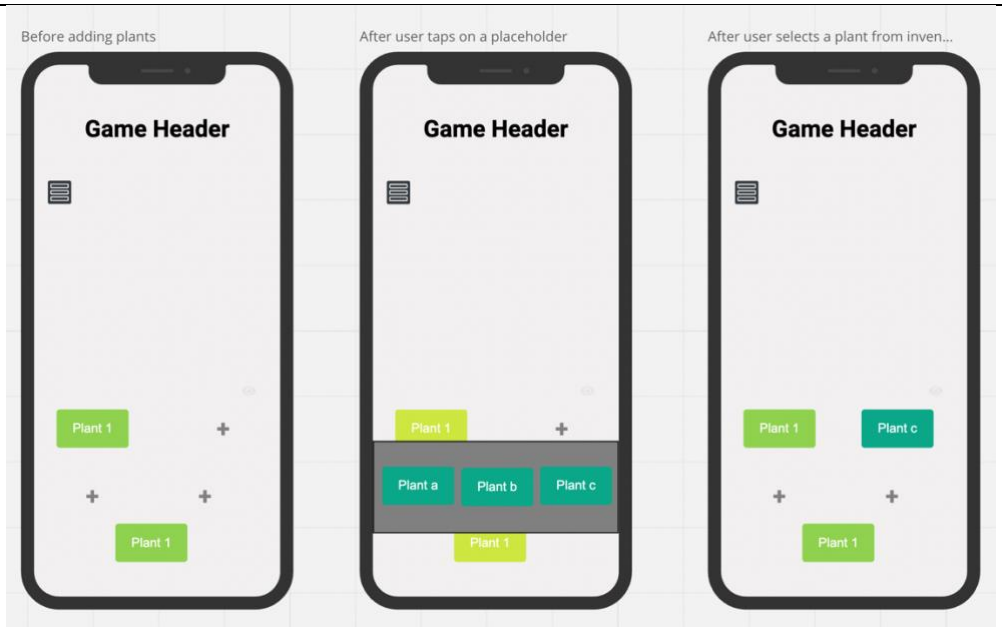
| US4-V1 | Displaying main screen elements | | |
|---|---|---|---|
| Description | As a player, I want to open the game app and see the main screen. The main screen should have the following elements: game header, a sun-shaped oracle, collapsed main menu icon, a background room, in-game plants the player currently has in room, and any empty placeholder spots for new plants. | | |
| Acceptance Criteria | Criteria ID | Description | Test Status |
| | US4-V1-AC1. | I can see the main game page when I enter the game app. | Pass |
| | US4-V1-AC2. | The main page shows all listed elements above. | Pass |
| Requirement Use Case | **Use Case:** Loading main game screen and its elements.<br>**Scope:** Main screen.<br>**Level:** Basic game set-up.<br>**Context:**<br>(1) When the user arrives at the main screen, they should be able to see a variety of game elements, some of which can navigate to other game features screens.<br>(2) These elements are game header, a sun-shaped oracle, collapsed main menu icon, a background room, in-game plants the player currently has in room, and any empty placeholder spots for new plants.<br>**Frequency of occurrence:** Every time player is at the main screen, either just after they open the game app, or after then return to main screen from other screens.<br>**Open issues:** The set of objects rendered on the main screen may be subject to changes. | | |
| Design Use Case | **Scope:** Main screen.<br>**Level:** Basic game set-up.<br>**Primary actors:** Player<br>**Description:** Player can see a set of game elements on the main screen.<br>**Dependencies:**<br>(1) The user has successfully installed the game, and able to open the game to the main game screen.<br>(2) Game assets (images and icons) are added appropriately to code file.<br>**Assumptions:** N/A<br>**Preconditions:** N/A<br>**Main Flow:** After the user arrive at main screen (either by opening the app, or returning to main screen from other game screens), the user should be able to see a number of game components, including: game header, a sun-shaped oracle, collapsed main menu icon, a background room, in-game plants the player currently has in room, and any empty placeholder spots for new plants.<br>**Post conditions:** All elements listed above are visible on the main screen and positioned according to UI design.<br>**Alternative Flows:** One or more of the components are not visible or unaligned with the UI design (invalid).<br>**Frequency of occurrence:** Every time the player is at the main screen. | | |

| | |
|---|---|
| | **Open issues:** The set of objects rendered on the main screen may be subject to changes. |
| UI Design | **Key UI components:**<br>Main game screen,<br>Game header,<br>Sun-shaped oracle<br>Collapsed main menu,<br>In-game plants and placeholders,<br>**Wireframe:**<br>  <br>**Interaction Flow:**<br>User arrives at the main screen with all elements displayed as expected.<br>**Data Entry and validation: N/A**<br>**Error Handling: N/A**<br>**UI Behaviour:** Components should take roughly similar time to load with no significant lagging.<br>**Visual Design Guidelines:**<br>Styling should align with the game's defined typography and colours.<br>Elements should be able to stand out from the background image colour / patterns.<br>All elements should be clearly visible without overlap. |

| US5-V1 | Add a plant to a selected placeholder spot on virtual room from the plant inventory. | | |
|---|---|---|---|
| Description | As a player, I want to add a plant from the in-game inventory to a selected placeholder in a virtual room (either in the default room on the main screen or rooms unlocked with XP level progression). | | |
| | Criteria ID | Description | Test Status |

<table>
<tr><td rowspan="6">Acceptance Criteria</td><td>US2-V1-AC1.</td><td>After I tap an available location in a virtual room, a plant inventory list should appear on the screen.</td><td>Pass</td></tr>
<tr><td>US2-V1-AC2.</td><td>I can scroll left and right to see all the plants available in the inventory.</td><td>Pass</td></tr>
<tr><td>US2-V1-AC3.</td><td>If I tap on a selected plant, an instance it should be added to the placeholder spot I had previously selected the virtual room.</td><td>Pass</td></tr>
<tr><td>US2-V1-AC4.</td><td>The size of the plant instance should match the player’s mastery level achieved for that type of plant.</td><td>Pass</td></tr>
<tr><td>US2-V1-AC5.</td><td>The plant inventory list should then disappear as soon as I have chosen a plant.</td><td>Pass</td></tr>
<tr><td>US2-V1-AC6.</td><td>The plant should remain in the spot for the rest of the game until I clear data or move it to archive.</td><td>Pass</td></tr>
<tr><td>Requirement Use Case</td><td colspan="3">Use Case: Adding plants to player’s virtual room.<br>Scope: Main screen gameplay.<br>Level: Core game feature.<br>Context:<br>(1) When the user arrives at a virtual room, they can choose from an inventory of in-game plants to learn about / care for in the game.<br>(2) The inventory is by default archived.<br>(3) In the virtual, there are five available plant spots with a "+" icon, each representing a placeholder for an in-game plant. To add a plant to any one of these spots, the player should first click on the "+" sign, after which the plant inventory containing all available plant species will appear as a horizontal scrollbar on screen.<br>(4) The player can select any one of the plant species in inventory by tapping on the plant, after which an instance the plant will occupy the placeholder spot. The inventory option should also go back into archive each time a plant is selected.<br>Frequency of occurrence: Every time player is in a virtual room (either in the default room on the main screen or rooms unlocked with XP level progression) and wishes to add a new plant to their room.<br>Open issues: What if the user wishes to delete a plant?</td></tr>
<tr><td>Design Use Case</td><td colspan="3">Scope: Main screen gameplay.<br>Level: Core game feature.<br>Primary actors: Player<br>Description: Player can view and choose from an in-game plant inventory and add selected plants to placeholder spots on the main screen.<br>Dependencies:<br>(1) The user has successfully installed the game, and able to open the game to a virtual room.<br>(2) The plant inventory data has been developed and rendered.<br>Assumptions: Player wishes to add new plants.<br>Preconditions: There are still empty placeholders in the room available, otherwise player must move one or more plants into archive or wait until new room is unlocked with XP level progression.</td></tr>
</table>

| | |
|---|---|
| | **Main Flow:** After the user arrive at a virtual room (either in the default room on the main screen or rooms unlocked with XP level progression), the user should be able to see placeholder spots for in-game plants, to which the user can add plants from the in-game plant inventory.<br>**Post conditions:** The player should first be able to see the complete inventory of plants appear as a horizontal scrollbar on the main screen once they select a placeholder. Player can tap on a plant to add it to the spot. After selection the inventory scrollbar would disappear (go back into archive until player decides to add another plant). The plant should also be added to the designated placeholder spot.<br>**Alternative Flows:**<br>**(1) Valid:**<br>Player arrives at virtual room, but no empty placeholders are available to hold any more plants.<br>Player arrives at virtual room but do not wish to add a new plant.<br>**(2) Invalid:**<br>Player arrives at a virtual room with remaining space but does not see any "+" icons.<br>Player arrives at the main screen, selects a placeholder, but inventory does not appear.<br>Player arrives at the main screen, selects a placeholder, the inventory list appears but is not correctly displayed - scroll function not working, layout inappropriate, not all plants are listed, etc.<br>Player selects a plant, but it is not added to the placeholder.<br>Player selects a plant, the plant is added to the placeholder, but inventory list does not go back into archive.<br>**Frequency of occurrence:** Every time the user wishes to add another plant to a virtual room.<br>**Open issues:** What if the user wishes to delete a plant if they are selected by mistake? |
| UI Design | **Key UI components:**<br>Virtual room (either default room on main screen or unlocked rooms with XP level progression),<br>Plant placeholder icon,<br>Plants (in inventory),<br>Inventory list as a scroll bar.<br>**Wireframe:** |




Interaction Flow:

User clicks on a placeholder icon "+".

Plant inventory appears on screen as a horizontal scrollbar on the bottom half of the screen.

User can select any plant in inventory.

Selected plant will be added to the placeholder.

Once a plant is selected, the inventory goes back into archive.

Data Entry and validation: N/A

Error Handling: N/A

UI Behaviour:

The inventory list should have a smooth animated effect when appearing on screen.

The inventory should by default be archived on the main screen, only to show when the player clicks on a placeholder icon.

The player should be able to view all inventory plants by scrolling left and right in the form of a scrollbar.

The scrolling animation should be seamless.

Visual Design Guidelines:

Styling should align with the game's defined typography and colours.

The expanded inventory list should be displayed centre in the bottom half of the screen.

All plants should show as an image icon with its name labelled next to the icon.

The font size and style of plant name labels should ensure it is eligible but not consuming too much space.

The size of each plant icon should be enough for the player to tell what it is.

Each plant icon should be of similar size.

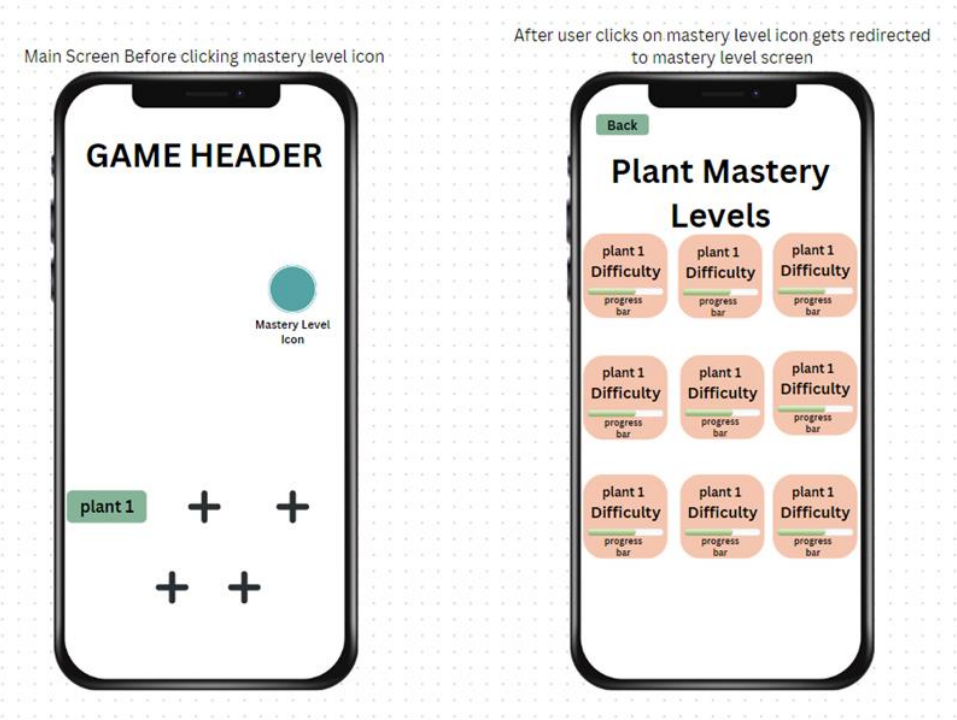
The size of each image icon should be sufficient for the user to effortlessly select.

Each option should have an icon with an appropriate image that matches the style of the game.

| | |
|---|---|
| | There should be enough spacing between each option for the player to choose and make selection. |

| | | | |
|---|---|---|---|
| US15-V1 | View plant mastery levels. | | |
| Description | As a player, I want to be able to see all plant mastery levels when clicking "Plant mastery" button | | |
| Acceptance Criteria | Criteria ID | Description | Test Status |
| | US15-V1-AC1. | When I tap on the "Plant Mastery" button on the main screen, a new screen should appear displaying all plant names along with their mastery levels. | Pass |
| | US15-V1-AC2. | The plant mastery levels should be presented in a card format, making it visually appealing and easy to navigate.<br>I should be able to go through all the plant mastery level cards to explore and understand my progress in the game. | Pass |
| | US15-V1-AC3. | Each card should contain name of the specific plant, its difficulty level, along with a bar of mastery level which increases as the quiz levels complete. | Pass |
| | US15-V1-AC4 | I should be able to go through all the plant mastery level cards to explore and understand my progress in the game. | Pass |
| Requirement Use Case | **Use Case:** Display all plants along with their difficulty and mastery levels in a card format when selecting "Plant mastery" button on main screen.<br>**Scope:** Standard gameplay.<br>**Level:** Core game feature.<br>**Context:**<br>(1) The player is actively engaged with the game and is currently on the main screen, where they have access to various features and options.<br>(2) The "Plant Mastery" button is prominently displayed on the main screen, indicating to the player that there is a dedicated section for tracking and exploring plant mastery levels.<br>(3) The player has completed various quiz levels in the game, contributing to their overall mastery of different plants.<br>(4) The player is interested in understanding their proficiency across different plants, ranging from familiar ones to more challenging ones, and is curious to explore the mastery levels associated with each.<br>(5) The player anticipates that the mastery level cards will provide insights into their progress and get insight of difficulty categories.<br>**Frequency of occurrence:** Every time player is at the main screen and wishes to see it's progress regarding plant mastery.<br>**Open issues:** N/A. | | |
| Design Use Case | **Scope:** Standard gameplay.<br>**Level:** Core game feature.<br>**Primary actors:** Player | | |

| | |
|---|---|
| | **Description:** Player can view and choose from the plant option menu for a plant by clicking that plant on the main game screen.<br>**Dependencies:**<br>(1) The app has been perfectly installed without any errors with optimum rendering.<br>(2) Plants and plant mastery levels are rendered in the game.<br>**Assumptions:** N/A<br>**Preconditions:**<br>(1) The player is on the main screen of the game.<br>(2) The "Plant Mastery" button is visible and accessible.<br>**Main Flow:**<br>(1) The player taps on the "Plant Mastery" button.<br>(2) A new screen opens, displaying plant mastery levels in a card format.<br>(3) The player sees the cards to view all available plants along with their difficulty and mastery level progress bar.<br>**Post conditions:** After Completing a quiz player can see the progress of their plant mastery level which increases as the player completes a quiz. Once satisfied with the progress user can click back button to navigate back to main screen.<br>**Alternative Flows:**<br>**Valid:**<br>(1) Once at the mastery screen, player taps on a specific card to view detailed information instead of returning to main screen.<br>(2) Player navigates back to the main screen from the plant mastery levels screen.<br>**Invalid:**<br>(1) The "Plant Mastery" button is not visible or accessible on the main screen.<br>(2) Tapping on the "Plant Mastery" button does not open the expected screen.<br>(3) The mastery level cards are not displayed as expected.<br>**Frequency of occurrence:** Every time the player wants to check plant mastery levels or view plant information from the main screen.<br>**Open issues:** N/A. |
| UI Design | **Key UI components:**<br>Main game screen<br>Plant Mastery Level Icon<br>Plants Cards<br>Plant name, difficulty level and progress bar<br>Back Button<br>**Wireframe:** |




**Interaction Flow:**

User clicks on a plant mastery level icon.

Plant Cards appear on the screen.

Plant cards with names along with difficulty levels and progress bar visible on the screen.

Back Button to navigate back to the main screen.

**Data Entry and validation:** N/A

**Error Handling:** N/A

**UI Behaviour:**

(1) Intuitive Navigation: The user should be able to navigate through the plant mastery cards effortlessly. Tapping on a specific card should provide additional details, and a clear back button should facilitate a seamless return to the main screen.

(2) Real-time Progress Update: As the player completes quiz levels, the mastery level progress bar on each plant card should update in real-time, providing instant feedback on the player's achievements.

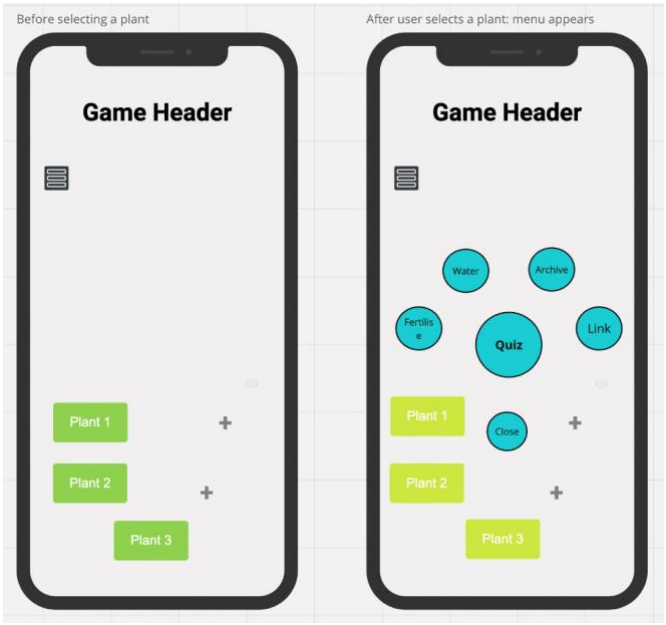
**Visual Design Guidelines:**

(1) Consistent Styling: Maintain a consistent design language with the rest of the game, using the established typography, colour palette, and overall theme. This consistency contributes to a cohesive and polished visual identity.

(2) Card Layout: The plant mastery levels should be presented in a card format with a visually pleasing layout. Ensure each card includes the plant name, difficulty level, and a progress bar, creating an organized and informative display.

(3) Engaging Progress Bar: Design the mastery level progress bar to be visually engaging and easy to interpret. Use colours or animations to signify progress, providing a quick visual indicator of the player's achievements.

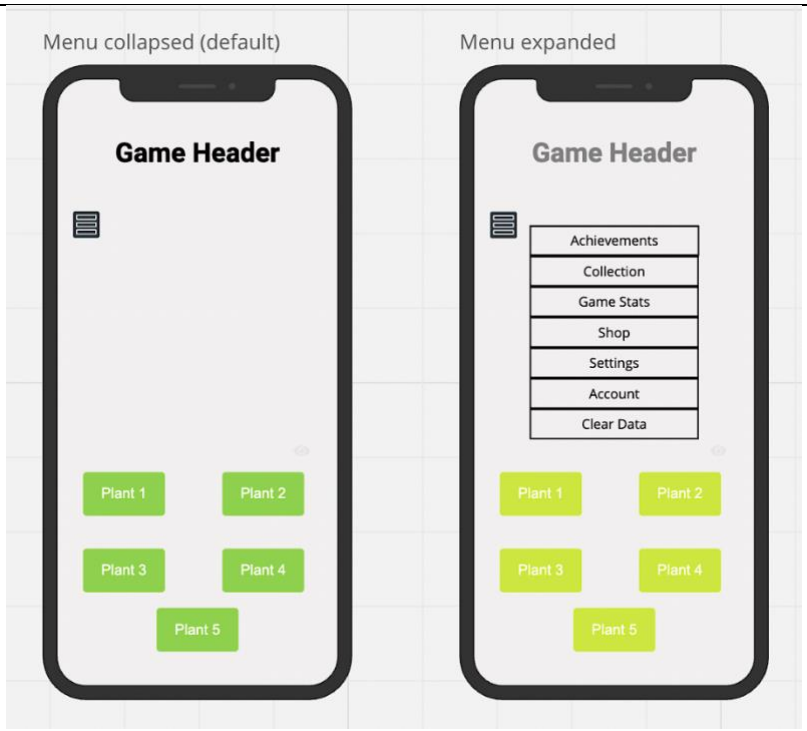
<table>
<tr><td>US17-V1</td><td colspan="3">Expand plant action menu.</td></tr>
<tr><td>Description</td><td colspan="3">As a player, I want to expand the plant action menu for a plant when I click on a plant in a virtual room.</td></tr>
<tr><td rowspan="4">Acceptance Criteria</td><td>Criteria ID</td><td>Description</td><td>Test Status</td></tr>
<tr><td>US17-V1-AC1.</td><td>After selecting a plant, the plant action menu should expand and show on the main page.</td><td>Pass</td></tr>
<tr><td>US17-V1-AC2.</td><td>The plant action menu has the following options: Fertilise, water, archive, link to real plant, plant quiz and close.</td><td>Pass</td></tr>
<tr><td>US17-V1-AC3.</td><td>The menu should no longer be visible (back into archive) if I click on the "close" option in the menu.</td><td>Pass</td></tr>
<tr><td>Requirement Use Case</td><td colspan="3">Use Case: Accessing plant action menu for a selected plant from main screen.<br>Scope: Standard gameplay<br>Level: Core game feature.<br>Context:<br>(1) When the user arrives at the main screen, they can choose the actions they wish to perform on any plant currently in their "room".<br>(2) The menu is by default archived, meaning that the user cannot see the options each time they arrive at or return to the main screen (or other virtual room screens).<br>(3) As part of the game's core functionalities, the player can perform a set of actions on a plant, including taking its quiz, water, fertilise, link the plant to a real-life plant they own, and place the in-game plant into archive (when the UI is holding too many plants). To perform these actions, the player needs to first select the targeted plant, click on it, and the plant option menu is expanded and shown on the main screen, listing all options available, from which the player can select.<br>Frequency of occurrence: Every time the user is at in a virtual room screen and wishes to perform certain actions on a plant.<br>Open issues: Are these the final list of menu options? What is their style design (font, colour, alignment etc.)?</td></tr>
<tr><td>Design Use Case</td><td colspan="3">Scope: Standard gameplay<br>Level: Core game feature.<br>Primary actors: Player<br>Description: Player can view and choose from the plant option menu for a plant by clicking that plant on a virtual room screen.<br>Dependencies:<br>(1) The player has successfully installed the game.<br>(2) The player can open the game to a virtual room screen (by default the main screen).<br>Assumptions: The player has at least one plant currently in the virtual room screen.<br>Preconditions: N/A<br>Main Flow: After the user arrive at a virtual room screen that contains plants in their main screen (either by opening the app, or returning from other game screens), the user should be able to view and select from an option menu containing actions to a particular plant by clicking on that plant in their main screen.<br>Post conditions: The user should be able to see the expanded plant option menu show up in the centre of the screen. The menu should include the following options: Quiz, water,</td></tr>
</table>

| | |
|---|---|
| | fertilise, archive, link to real plant and close. Player can either select an option from the expanded menu to go to another game screen or collapse the menu by clicking on the "close" option in the menu.<br>**Alternative Flows:**<br>**(1) Valid:**<br>Player arrives at a virtual room screen but does not wish to perform any actions to plants. Instead, player proceeds to other functionalities (e.g., shop, view mastery).<br>**(2) Invalid:**<br>Player arrives at a virtual room but have not yet added any plants.<br>Player clicks on a plant, but the plant action menu does not appear.<br>Player clicks on a plant; the plant action menu expands but are not properly displayed.<br>**Frequency of occurrence:** Every time the user is at a virtual room screen and wishes to perform one of the actions (take quiz, water, fertilise, archive, link) to a plant in their room.<br>**Open issues:** Are these the final list of menu options? What is their style design (font, colour, alignment etc.)? |
| UI Design | **Key UI components:**<br>Virtual room screen (by default the main screen)<br>Plants (already added to the player's in-game collection)<br>Expanded plant action menu.<br>**Wireframe:**<br>  <br><br>**Interaction Flow:**<br>User clicks on a plant displayed on a virtual room screen.<br>Expanded menu appears on screen as 6 round icons, the quiz icon is placed in the centre with the others surrounding.<br>User can select any item in the list to go to the corresponding game page.<br>User can click anywhere on the "close" option to collapse the menu and stay at main page.<br>**Data Entry and validation: N/A**<br>**Error Handling: N/A** |

| | |
|---|---|
| | **UI Behaviour:**<br>The plant option menu should have a smooth animated effect when expanded and collapsed.<br>The menu should by default be archived on the main screen, only to show when the player clicks on a plant.<br>The menu should always be placed in the centre of the screen regardless of the plant's relative position in the room.<br>**Visual Design Guidelines:**<br>Styling should align with the game's defined typography and colours.<br>The expanded menu should be displayed at the centre of the main screen.<br>The expanded menu option list should appropriately sized, taking up the centre space in the main screen.<br>When menu is expanded, the original images in the main screen should not obstruct its visibility, i.e., the player should be able to clearly see all option icons.<br>Each option should have an icon with an appropriate image that matches the style of the game.<br>There should be enough spacing between each option for the player to choose and make selection. |

| US46-V1 | Expand main game menu. | | |
|---|---|---|---|
| Description | As a player, I want to expand the main game menu when I select the menu button on the main page. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US46-V1-AC1.** | After selecting the collapsed menu widget, the main game menu should expand and show on the main page. | **Pass** |
| | **US46-V1-AC2.** | The main menu has the following options: Achievements, collection, game stats, shop, settings, account, and clear data. | **Pass** |
| | **US46-V1-AC3.** | The menu should go back into archive if I click anywhere else on the screen. | **Pass** |
| Requirement Use Case | **Use Case:** Accessing main game menu on main screen.<br>**Scope:** Standard gameplay.<br>**Level:** Core game feature.<br>**Context:**<br>(1) When the user arrives at the main screen, they can enter other game screens through the main menu.<br>(2) The menu is represented by a menu widget in the shape of three horizontal lines on the main screen. The default menu state is collapsed, meaning that the user cannot see the options by default when entering main screen.<br>(3) The menu is un-collapsed when user clicks the menu widget, from which the user can select from a series of options.<br>**Frequency of occurrence:** Every time the user is at the main screen and wishes to access main menu options. | | |

| | |
|---|---|
| | **Open issues:** Are these the final list of menu options? What is their style design (font, colour, alignment etc.)? |
| Design Use Case | **Scope:** Standard gameplay<br>**Level:** Core game feature.<br>**Primary actors:** Player<br>**Description:** Player can view and choose from the main game menu on the main game screen.<br>**Dependencies:**<br>(1) The user has successfully installed the game, and able to open the game to the main game screen.<br>(2) The user has successfully returned to the main screen from other screens in the game.<br>**Assumptions: N/A**<br>**Preconditions:** Player has been able to successfully install and run game on a mobile phone.<br>**Main Flow:** After the user arrive at the main screen (either by opening the app, or returning to main screen from other game screens), the main menu widget should be visible on the upper left side of the screen. The user should be able to un-collapse the menu by clicking on the widget icon.<br>**Post conditions:** The user should be able to see the expanded main menu show up in the centre of the screen. The menu should include the following options: Achievements, collection, game stats, shop, settings, account, and clear data. Player can either select an option from the expanded menu to go to another game screen or collapse the menu by clicking anywhere else on the screen.<br>**Alternative Flows:**<br>**(1) Valid:**<br>Player arrives at main screen but does not wish to view main game menu. Instead, player proceeds to other functionalities (e.g., shop, view mastery, plant actions).<br>**(2) Invalid:**<br>Player arrives at the main screen but cannot see the menu widget.<br>Player clicks on the menu widget, but menu does not expand.<br>Player clicks on the menu widget; menu expands but are not properly displayed.<br>**Frequency of occurrence:** Every time the user is at the main screen and wishes to access main menu options.<br>**Open issues:** Are these the final list of menu options? What is their style design (font, colour, alignment etc.)? |
| UI Design | **Key UI components:**<br>Main game screen<br>Main menu widget icon (collapsed icon in the shape of three horizontal lines).<br>Expanded menu.<br>**Wireframe:** |




Interaction Flow:

User clicks on the Main menu icon button on main screen.

Expanded menu appears on screen in a list.

User can select any item in the list to go to the corresponding game page.

User can click anywhere on the main screen to collapse the menu and stay at main page.

Data Entry and validation: N/A

Error Handling: N/A

UI Behaviour:

The main menu should have a smooth animated effect when expanded and collapased.

The menu should by default be collapsed on the main screen,

The collapsed menu icon should always be placed along the upper left edge.

Visual Design Guidelines:

Styling should align with the game's defined typography and colors.

The collapsed menu icon should be visible, not obstruct any ther widgets on the main screen.

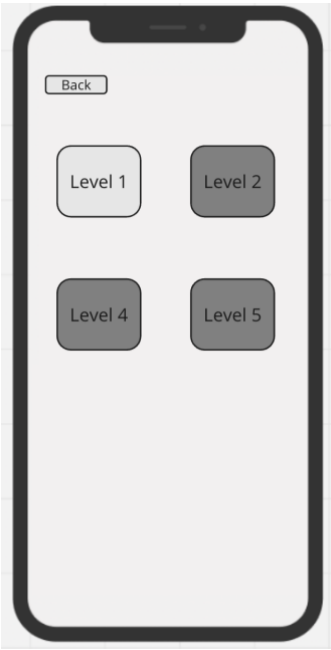
The expanded menu should be displayed at the center of the main screen.

The expanded menu option list should not fill up the entire main screen, leaving space for the user to click if they want to collapse the menu and return to main game.

When the expanded menu is displayed, the main game screen background colours should be dimmed, enabling the menu to stand out to the player.

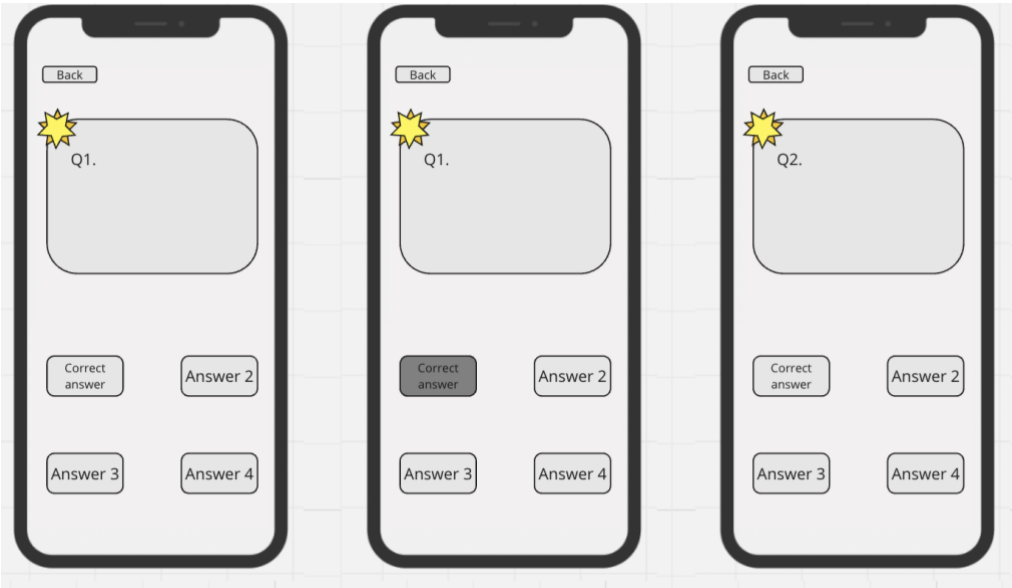
There should be enough spacing between each option for the player to choose and make selection.

<table>
<tr><td>US25-V1</td><td colspan="3">Level selection screen.</td></tr>
<tr><td>Description</td><td colspan="3">I want to enter the level selection screen via the quiz button and see the level selection screen. The level selection screen should have the following elements: header back button and all available level buttons (unlocked/locked).</td></tr>
<tr><td rowspan="5">Acceptance Criteria</td><td>Criteria ID</td><td>Description</td><td>Test Status</td></tr>
<tr><td>US25-V1-AC1.</td><td>I can see the level selection screen when I enter the level selection screen via the quiz button.</td><td>Pass</td></tr>
<tr><td>US25-V1-AC2.</td><td>The level selection screen shows all elements listed above.</td><td>Pass</td></tr>
<tr><td>US25-V1-AC3.</td><td>The level selection screen shows the correct details of the selected plant species.</td><td>Pass</td></tr>
<tr><td>US25-V1-AC4</td><td>The only unlocked level when the player first enters the level screen would only be level 1.</td><td>Pass</td></tr>
<tr><td>Requirement Use Case</td><td colspan="3">Use Case: Loading level selection screen and its elements.<br>Scope: Quiz feature.<br>Level: Core game feature.<br>Context:<br>(1) When the user arrives at the level selection screen, they should be able to see a variety of level selection screen elements from which to perform actions within the level selection screen.<br>(2) These elements are back button and all available level buttons (unlocked/ locked).<br>Frequency of occurrence: Every time player is at the level selection screen, accessing from the quiz button in the plant menu.<br>Open issues: The set of objects rendered on the level selection screen may be subject to changes.<br>The style of how to show the elements may change.</td></tr>
<tr><td>Design Use Case</td><td colspan="3">Scope: Standard gameplay.<br>Level: Core game feature.<br>Primary actors: Player<br>Description: Player can see a number of game elements on the level selection screen.<br>Dependencies:<br>(1) The user has successfully installed the game, and able to open up the game to the main game screen.<br>(2) The user could add a plant and could access both the plant menu and the quiz button.<br>(3) Game assets (images and icons) are added appropriately to code file.<br>Assumptions: N/A<br>Preconditions:<br>(1) The user has added at least one plant to their main menu.<br>Main Flow:<br>After the user has added a plant, the user should be able to click on the plant and open the plant menu. They have the option to choose to enter the level selection screen by clicking the quiz button within the plant menu.</td></tr>
</table>

| | |
|---|---|
| | **Post conditions:** All elements listed above are visible on the level selection screen.<br>**Alternative Flows:**<br>**Invalid:**<br>(1) One or more of the components are not visible or unaligned with the UI design.<br>(2) The user could not enter the level selection screen.<br>**Frequency of occurrence:** Every time the user enters the level selection screen.<br>**Open issues:** The set of objects rendered on the link screen may be subject to changes. |
| UI Design | **Key UI components:**<br>Level selection screen<br>All available levels for the selected plant<br>Back button<br>**Wireframe:**<br>  <br><br>**Interaction Flow:**<br>User enters the level selection screen.<br>User sees their current level and all levels for the selected plant.<br>**Data Entry and validation:** N/A<br>**Error Handling:** N/A<br>**UI Behaviour:** The back button should by default appeared on the level selection screen.<br>**Visual Design Guidelines:**<br>Styling should align with the game's defined typography and colours.<br>Locked levels should have a shade when comparing to the unlocked levels. |

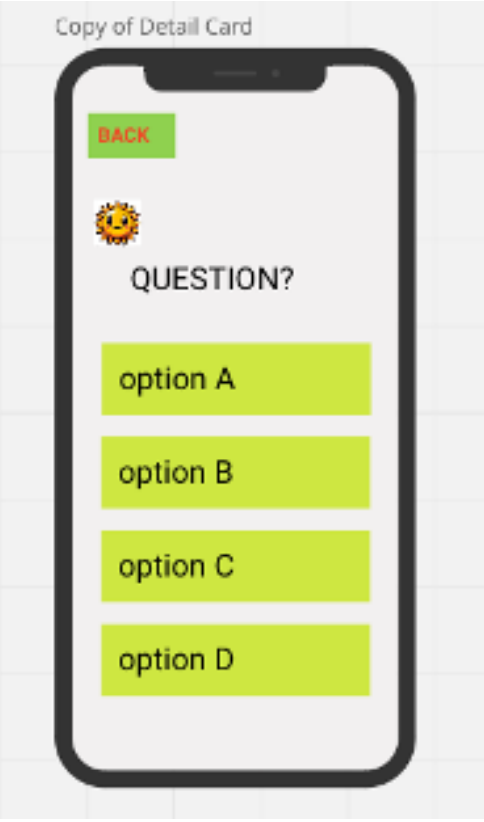
| US26-V1 | Proceed to next question if a question is answered correctly. |
|---|---|
| Description | As a player, I want to a proceed to the next question if I answered the correct answer via one of the answer buttons in the quiz screen. |

<table>
<tr><td rowspan="3">Acceptance Criteria</td><td>Criteria ID</td><td>Description</td><td>Test Status</td></tr>
<tr><td>US26-V1-AC1.</td><td>After pressing the correct answer button, the quiz screen will navigate to the next question</td><td>Pass</td></tr>
<tr><td>US26-V1-AC2.</td><td>The questions will be rendered from all the questions available for that level and questions for the previous levels.</td><td>Pass</td></tr>
<tr><td>Requirement Use Case</td><td colspan="3">Use Case: Answering the question with a correct answer and navigate to the next question.<br>Scope: Quiz<br>Level: Core game feature.<br>Context:<br>(1) When the user presses the correct answer button, the user could enter the next question of a specific level.<br>(2) The questions will be randomly picked from both the questions specific levels and the previous levels.<br>(3) The player can proceed to the next question of the selected level. To perform this action, the player needs to press the answer button which contains the correct answer.<br>Frequency of occurrence: Every time player is at the quiz screen, answering the question with a correct answer.<br>Open issues: N/A</td></tr>
<tr><td>Design Use Case</td><td colspan="3">Scope: Quiz<br>Level: Core game feature.<br>Primary actors: Player<br>Description: Player can enter the next question of the selected level by clicking on one of the answer buttons that contains the correct answer.<br>Dependencies:<br>(1) The user has successfully installed the game, and able to open the game to the main game screen.<br>(2) The user could add a plant and has accessed the following: plant menu, quiz button, level selection screen and selected a level.<br>Assumptions: N/A<br>Preconditions: The user has added at least one plant to their main menu.<br>Main Flow: After the user arrive at the quiz screen (by successfully selecting a level of a selected plant or answering the question in a correct answer), the player could choose one answer button that contains the correct answer.<br>Post conditions: The game show the next question of the plant in through the quiz screen.<br>Alternative Flows:<br>(1) Valid:<br>Player clicks on back button and navigate to the main menu instead of taking the quiz.<br>(2) Invalid:<br>Player clicks on the correct answer, but the quiz screen does not show the next question.<br>Player clicks on the correct answer but the quiz screen shows duplicated questions.</td></tr>
</table>

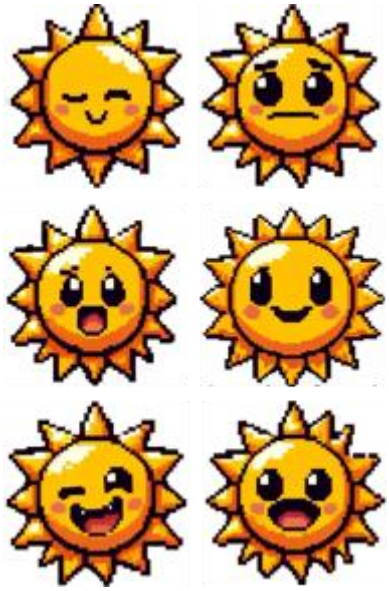
| | |
|---|---|
| | **Frequency of occurrence:** Every time the user answers the question correctly when it is not the last question.<br>**Open issues:** N/A. |
| UI Design | **Key UI components:**<br>Quiz screen<br>Answer button<br>Field for current question<br>**Wireframe:**<br>  <br><br>**Interaction Flow:**<br>User press on the correct answer.<br>Quiz screen navigates the user to the next question.<br>**Data Entry and validation: N/A**<br>**Error Handling: N/A**<br>**UI Behaviour:** The answer button should have a smooth animated effect when pressed.<br>**Visual Design Guidelines:**<br>Styling should align with the game's defined typography and colours. |

| US98-V1 | Receiving oracle guidance throughout the quiz experience. | | |
|---|---|---|---|
| Description | I want the Oracle to offer fun, thematic quizzes related to plant care to add an entertaining and engaging element to my gameplay. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US46-V1-AC1.** | Quiz content should be relevant to the current level or situation in the game, enhancing the thematic experience. | **Pass** |
| | **US46-V1-AC2.** | Quizzes should be easily accessible through a clear and intuitive interface. | **Pass** |

| | | | |
|---|---|---|---|
| | US46-V1-AC3. | All quiz questions must be directly related to the theme of plant care and maintenance. | Pass |
| | US46-V1-AC4. | The difficulty of the quizzes should be appropriate for the game's target audience, offering a fun challenge without being overly complex. | Pass |
| | US46-V1-AC5. | Players should receive immediate feedback after answering each question, indicating if their answer was correct or incorrect. | Pass |
| | US46-V1-AC6. | Correct answers in quizzes can contribute to rewards in the game, such as coins, experience points (XP), or unlocking new content. | Pending |
| | US46-V1-AC7. | The quiz interface should be visually appealing and consistent with the game's overall design. | Pending |
| | US46-V1-AC8. | Text and buttons should be clearly readable and accessible for players of all ages. | Pending |
| | US46-V1-AC9. | Players should have the option to exit the quiz at any point. | Pass |
| Requirement Use Case | **Use Case:** Accessing the quiz with the presence of oracle guidance.<br>**Scope:** Quiz.<br>**Level:** Core game feature.<br>**Context:**<br>(1) The player encounters the Oracle in the game, who offers a thematic quiz related to plant care.<br>(2) The Oracle presents a series of questions, one at a time, related to plant care, appropriate to the player's current level.<br>(3) The player selects an answer for each question.<br>(4) After each answer, the Oracle provides immediate feedback, indicating if the answer was correct or incorrect.<br>(5) If the answer is correct, the player receives rewards like coins or XP.<br>(6) Once all questions are answered, the player is given the option to replay the quiz or return to the main game.<br>**Frequency of occurrence:** Depending on the player's gameplay style and progression, quizzes can occur at regular intervals or at specific milestones within the game. | | |
| Design Use Case | **Scope:** Quiz.<br>**Level:** Core game feature.<br>**Primary actors:** Player<br>**Description:** Player can access the quiz under with the presence of oracle guidance.<br>**Preconditions:**<br>(1) The player has access to the game and has reached a level where quizzes are available.<br>(2) The Oracle is integrated into the game with functional capabilities to present quizzes.<br>**Main Flow:**<br>(1) The Player starts the Quiz. | | |

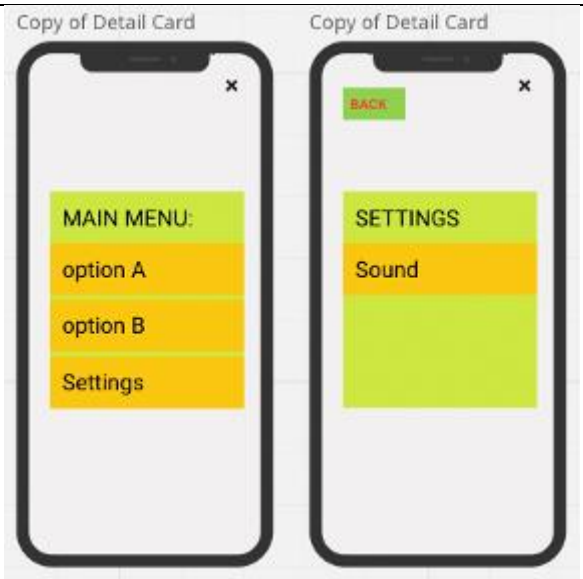
| | |
|---|---|
| | (2) The Oracle presents a serious a question in a dialog box one at a time.<br>(3) After each answer the Oracle provides feedback in a dialog box.<br>(4) If the answer is correct the dialog box displays the reward information.<br>(5) If all questions are answered, the quiz can be replayed, or the player can return to the main screen.<br>**Post conditions:** The quiz flow is presented as specified above.<br>**Alternative Flows:**<br>**(1) Valid:**<br>At any point, the player can choose to skip or exit the quiz.<br>**(2) Invalid:**<br>The questions can relate to another plant.<br>The dialog box displays information in an improper way.<br>The player clicks the "back" to main screen button but stays at the quiz level. |
| UI Design | **Key UI components:**<br>Oracle.<br>Quiz dialog box and candidate answers.<br>Back button.<br>Background image.<br>**Wireframe:**<br>  |

<table>
<tr><td>US45-V1</td><td colspan="3">Oracle displaying facial expressions</td></tr>
<tr><td>Description</td><td colspan="3">I want the Oracle to have a range of facial expressions that change randomly to make the game feel livelier and more interactive.</td></tr>
<tr><td rowspan="7">Acceptance Criteria</td><td>Criteria ID</td><td>Description</td><td>Test Status</td></tr>
<tr><td>US45-V1-AC1.</td><td>The Oracle should display a wide range of facial expressions, reflecting different emotions and reactions.</td><td>Pass</td></tr>
<tr><td>US45-V1-AC2.</td><td>Facial expressions should change randomly, adding a dynamic and unpredictable element to the gameplay.</td><td>Pass</td></tr>
<tr><td>US45-V1-AC3.</td><td>Changes in expressions should be smooth and natural, avoiding abrupt or disjointed transitions.</td><td>Pass</td></tr>
<tr><td>US45-V1-AC4.</td><td>Facial expressions should be consistent with the overall art style and character design of the game.</td><td>Pass</td></tr>
<tr><td>US45-V1-AC5.</td><td>Players should receive immediate feedback after answering each question, indicating if their answer was correct or incorrect.</td><td>Pass</td></tr>
<tr><td>US45-V1-AC6.</td><td>All expressions should be appropriate and easily understandable for the game's target audience.</td><td>Pass</td></tr>
<tr><td>Requirement Use Case</td><td colspan="3">Use Case: Seeing dynamic facial expressions of the Oracle.<br>Scope: Game styling and design.<br>Level: Core game feature.<br>Context:<br>(1) The Oracle changes facial expressions randomly or in response to game events.<br>(2) The player observes these changes during gameplay, enhancing the interactive experience.<br>(3) The expressions are relevant to the player's actions and the game's storyline.<br>Frequency of occurrence: Continuously during gameplay, with variations in expressions occurring at unpredictable intervals.</td></tr>
<tr><td>Design Use Case</td><td colspan="3">Scope: Game styling and design.<br>Level: Core game feature.<br>Primary actors: Player, Oracle<br>Description: Player can see dynamic facial expressions of the Oracle throughout various stages of the game.<br>Preconditions:<br>(1) The Oracle character is designed with the capability to display a range of facial expressions.<br>(2) The Oracle is functionally integrated into the game, capable of providing educational content on plant care.<br>Main Flow:<br>(1) The player can observe a set of facial expressions for the Oracle character.<br>(2) The algorithm changes the Oracle's expressions randomly or based on specific triggers.<br>(3) The animation ensures smooth and natural transitions between different expressions.<br>Post conditions: The quiz flow is presented as specified above.</td></tr>
</table>

| | |
|---|---|
| | **Alternative Flows:**<br>**Invalid:** The animations do not work smoothly and naturally. |
| UI Design | **Key UI components:**<br>Oracle.<br>**Wireframe: (Oracle expressions)**<br>  |

| US41-V1 | Game settings screen | | |
|---|---|---|---|
| Description | As a player, I want to enter a settings screen after selecting the "Settings" option in the main menu, so that I can customize my game experience according to my preferences. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US41-V1-AC1.** | The "Settings" option should be clearly visible and accessible in the main menu. | **Pass** |
| | **US41-V1-AC2.** | Upon selecting this option, the player should be taken to a settings screen. | **Pass** |
| | **US41-V1-AC3.** | The settings screen should offer options for customization, such as sound control or display settings. | **Pass** |
| | **US41-V1-AC4.** | Navigating through the settings should be intuitive and user-friendly. | **Pass** |
| | **US41-V1-AC5.** | Changes made in the settings should be applied immediately and saved for future game sessions. | **Pass** |
| | **US41-V1-AC6.** | There should be an easy and straightforward way to return to the main menu from the settings screen. | **Pass** |
| Requirement Use Case | **Use Case:** Accessing game settings.<br>**Scope:** Game set-up.<br>**Level:** User functionality. | | |

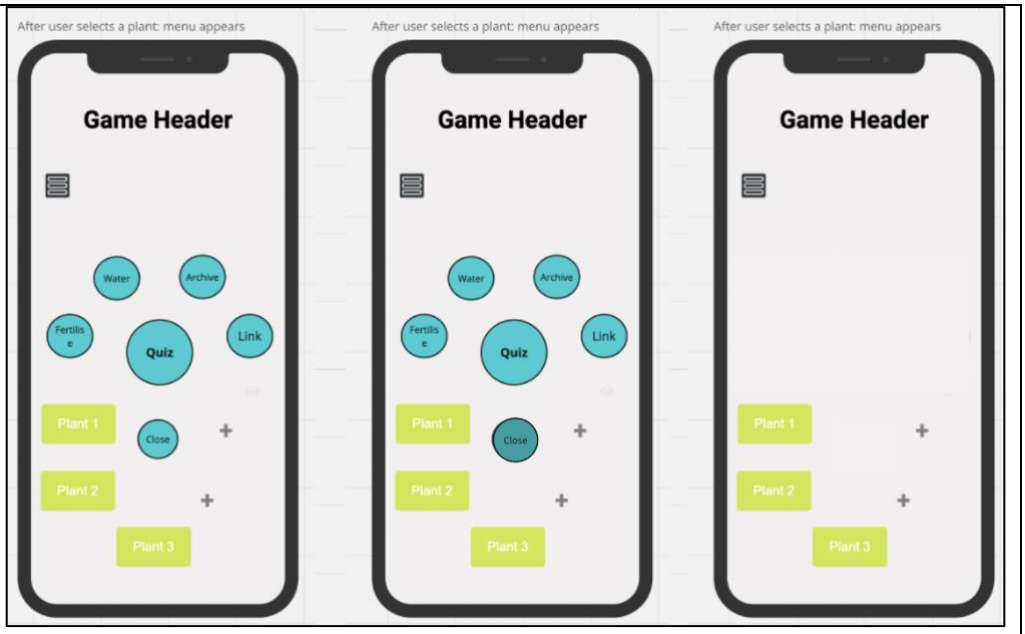
| | |
|---|---|
| | **Context:**<br>(1) The player selects the "Settings" option from the main menu.<br>(2) The settings screen is displayed, showing various customizable options.<br>(3) The player makes changes to the settings and those changes are immediately applied.<br>**Frequency of occurrence:** As needed by the player each time they wish to customise game settings, not bound to any specific gameplay intervals. |
| Design Use Case | **Scope:** Game set-up.<br>**Level:** User functionality.<br>**Primary actors:** Player<br>**Description:** Player can access settings screen to customise game settings with options provided.<br>**Preconditions:**<br>(1) The game includes a functional settings section.<br>(2) The game supports various user preferences and settings.<br>**Main Flow:**<br>(1) The player can access a "Settings" button in the main menu.<br>(2) After clicking the button, player can see a settings screen with various options.<br>(3) All changes made by the Player are automatically saved and applied.<br>**Alternative Flows:**<br>**Valid:** At any time the player can exit and return to main screen.<br>**Invalid:**<br>(1) Player selects “Settings” button but cannot navigate to settings screen.<br>(2) Setting screen is not appropriately displayed.<br>(2) The changes made in the settings are not automatically saved and the game is not modified. |
| UI Design | **Key UI components:**<br>Main menu<br>Settings button<br>Settings options<br>Back to the main menu button<br>**Wireframe:** |

| | |
|---|---|
| |  |

| US42-V1 | Clear game data | | |
|---|---|---|---|
| Description | As a Player, I want the ability to clear all game data after selecting the "Clear data" option in the main menu, so that I can restart the game from scratch whenever I choose. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US42-V1-AC1.** | A "Clear data" option should be visibly accessible in the main menu. | **Pass** |
| | **US42-V1-AC2.** | Upon selecting this option, a confirmation prompt should appear to prevent accidental data clearance. | **Pass** |
| | **US42-V1-AC3.** | All game data, including progress, scores, and settings, should be completely erased upon confirmation. | **Pass** |
| | **US42-V1-AC4.** | After clearing the data, the game should be able to restart from the beginning as if freshly installed. | **Pass** |
| | **US42-V1-AC5.** | The process should be straightforward and user-friendly, with clear instructions and confirmations. | **Pass** |
| | **US42-V1-AC6.** | The feature should be designed to ensure it does not affect the game's core files or stability. | **Pass** |
| Requirement Use Case | **Use Case:** Resetting game data.<br>**Scope:** Game set-up.<br>**Level:** User functionality.<br>**Context:**<br>(1) The player navigates to the main menu and selects the "Clear data" option.<br>(2) A confirmation dialog appears, asking the player to confirm their choice. | | |

| | |
|---|---|
| | (3) Upon confirmation, the game clears all stored data and resets to its initial state.<br>**Frequency of occurrence:** As needed by the player each time they wish rest the game, not bound to any specific gameplay intervals. |
| Design Use Case | **Scope:** Game set-up.<br>**Level:** User functionality.<br>**Primary actors:** Player<br>**Description:** Player can access settings screen to customise game settings with options provided.<br>**Preconditions:**<br>(1) The game has a built-in feature to clear all user data.<br>(2) The game's architecture allows for secure and complete data clearance.<br>**Main Flow:**<br>(1) The player can access a "Clear data" button in the main menu.<br>(2) The player can see a confirmation dialog to appear upon the selection of the "Clear data" option.<br>(3) Upon confirmation the Player can delete all game data securely and reset the game.<br>(4) The player receives feedback that the data has been successfully cleared.<br>**Alternative Flows:**<br>**Invalid:** Data was not deleted, and the game was not reset. |
| UI Design | **Key UI components:**<br>Main menu<br>Settings button<br>Settings options<br>Back to the main menu button<br>**Wireframe:**<br>  |

<table>
<tr><td>US37-V1</td><td colspan="3">Close plant action menu.</td></tr>
<tr><td>Description</td><td colspan="3">As a player, I want to water the plant by pressing the water button in plant action menu.</td></tr>
<tr><td rowspan="2">Acceptance Criteria</td><td>Criteria ID</td><td>Description</td><td>Test Status</td></tr>
<tr><td>US37-V1-AC1.</td><td>After pressing the close button, the plant action menu will collapse.</td><td>Pass</td></tr>
<tr><td>Requirement Use Case</td><td colspan="3">Use Case: Closing the plant menu using the close option.<br>Scope: Game set-up.<br>Level: Core game features.<br>Context:<br>(1) The close button is shown as a part of the menu by default.<br>(2) When the plant action menu is expanded, the user can click the close button to collapse the menu.<br>Frequency of occurrence: Every time the user had previously expanded the plant menu and wishes to close it.</td></tr>
<tr><td>Design Use Case</td><td colspan="3">Scope: Game set-up.<br>Level: Core game features.<br>Primary actors: Player<br>Description: Player can close the plant menu of the selected plant by clicking on the close button.<br>Dependencies:<br>(1) The user has successfully installed and launched the game.<br>(2) User has previously expanded the plant action menu.<br>Assumptions:<br>(1) Player has at least one plant in their virtual room or collection.<br>Preconditions: Player has expanded the plant action menu for a plant.<br>Main Flow: After the user arrive at a virtual room screen or collection screen and expanded the plant action menu by clicking a plant, the user should then be able to click on the close option to close the menu.<br>Postconditions:<br>The plant menu will close.<br>Alternative Flows:<br>Valid: Player clicks on the plant and closes the plant menu.<br>Invalid: Player clicks on the close button, but the plant menu didn't collapse.<br>Frequency of occurrence: Every time the user opened the plant menu and wishes to close it.</td></tr>
<tr><td>UI Design</td><td colspan="3">Key UI components:<br>Virtual room screen (by default the main game screen) or collection screen.<br>Plants (already added to the player's main screen)<br>Expanded plant action menu.<br>Close button in the plant action menu.<br>Wireframe:</td></tr>
</table>




**Interaction Flow:**

User clicks on the close button of the plant menu.

The plant menu collapses (disappears).

**UI Behaviour:**

The close button should have a smooth animated effect when pressed.

The close button should by default be shown on the plant menu as the bottom most option.

**Visual Design Guidelines:**

Consistent Styling: Maintain a consistent design language with the rest of the game, using the established typography, colour palette, and overall theme. This consistency contributes to a cohesive and polished visual identity.

### 3.5.6. CRC Cards

Note that all CRC cards are reverse engineered in Sprint 7 based on a completed code base. This is as an unfortunate outcome of the team's failure to sequentially implement the software process model. Therefore, the CRC cards were not developed in iterative versions. All cards below are in built in accordance with the final version of the completed game.

**The following CRC cards were developed for the functionalities built in Sprint 5.**

| Class: BackgroundImageComponent | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Render an image background for a specific room or environment.<br>Display plants in predefined positions within the environment.<br>Filter and manage plants specific to the current background.<br>Dynamically style plant components based on their positions. | PlantDataContext |

Relevant user stories: US4-V1, US10-V1

| Class: CircularMenu (Component) | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Display a floating menu with animated icons in a modal.<br>Handle animations for the menu items' appearance and disappearance.<br>Dynamically position menu items based on given angles and a fixed radius.<br>Render different types of menu items (either images or icons) based on the menuItems prop.<br>Trigger a callback function (onPress) when a menu item is selected.<br>Scale menu items differently based on whether they are the central icon or not. | Icons and Buttons |

Relevant user stories: US17-V1, US18-V1, US19-V1, US23-V1, US24-V2, US25-V1

| Class: Header Component | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Render a header section for the application, typically at the top of the screen.<br>Display images specifically designed for the header, adjusting to the device's width.<br>Manage the layout and styling of the header images. | GameScreen |

Relevant user stories: US4-V1

| Class: MenuComponent | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Display a modal menu with various options such.<br>Handle navigation to different screens within the app.<br>Implement the functionality to clear data and reset the game state.<br>Provide touchable options for user interaction. | HomeStackNavigator<br>playerConfigContext<br>GameText<br>GameScreen |

Relevant user stories: US46-V2, US41-V1

| Class: Oracle Component | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Display an animated character (Oracle) with various facial expressions.<br>Change the Oracle's expression based on specific triggers or randomly.<br>Manage timed animations such as blinking and facial changes.<br>Handle forced facial expressions for a set duration if provided.<br>Dynamically adjust styling and positioning based on the style prop. | GameScreen |

Relevant user stories: US45-V1, US98-V1, US44-V1

| Class: Plant Component | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Display an interactive plant object within the game environment.<br>Allow users to interact with the plant through various gestures, like tapping or holding.<br>Show different plant stages based on the growth progress.<br>Present a menu with options such as watering, learning, archiving, and linking to real-life plant information.<br>Handle archiving and restoring plants from the archive. | HomeStackNavigator<br>plantsDataContext<br>speciesProgressContext |

Relevant user stories: US5-V1, US17-V1, US37-V1

| Class: PlantHitbox | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Determine the hitbox size of a plant based on its growth progress.<br>Calculate the hitbox dimensions dynamically to ensure accurate interaction zones for plants at different growth stages. | |

Relevant user stories: US4-V1, US5-V1, US15-V1, US27-V2

| Class: QuizQuitButton | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Provide a button for users to initiate quitting from a quiz or game session.<br>Display a confirmation dialog when the button is pressed to ensure user intent. | HomeStackNavigator |

Relevant user stories: US1-V1

| Class: HomeStackNavigator | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Manage the navigation stack for the home screen and related screens in the app.<br>Define navigation routes and screen components for each route.<br>Provide custom navigation options for different screens. | Screens |

Relevant user stories: US1-V1, US15-V1, US41-V1

| Class: GameScreen | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Serve as the main game interface for the user.<br>Display various UI components<br>Manage and display the game's menu<br>Implement navigation to different sections of the game | HomeStackNavigator<br>playerConfigContext<br>BackgroundImageComponent |

| | |
|---|---|
| Maintain and update the game's state, including menu visibility and player configuration.<br>Handle the game's audio setup and background music | |

Relevant user stories: US4-V1, US5-V1, US15-V1, US17-V1, US41-V1, US46-V1

| Class: QuizScreen | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Present a quiz interface to the user, allowing them to answer questions based on plant trivia.<br>Keep track of the current question, correct and incorrect answers count, and handle navigation between questions.<br>Handle user interactions such as answering questions, navigating through the quiz, and closing modals.<br>Display instructions modal, feedback messages, and game-over modal as necessary. | plantsTriviaConfig, levelsConfig<br>plantsDataContext, PlayerConfigContext<br>HeartsDisplay, CoinDisplay, Oracle |

Relevant user stories: US25-V2, US26-V1, US98-V1

| Class: SettingsScreen | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Provide a settings interface where users can toggle the background music on or off.<br>Manage the state of the music setting | backgroundMusic<br><br>GameText |

Relevant user stories: US41-V1

| Class: plantsTriviaConfig | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Store trivia questions and answers for different levels and plant types.<br>Provide instructions and coin rewards for each level.<br>Include a variety of trivia topics including care instructions, propagation methods, common diseases, environmental needs, and other unique characteristics of each plant type. | plantsConfig<br><br>levelsConfig |

Relevant user stories: US26-V1, US27-V2, US98-V1

| Class: GameScreenStyles | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Define and manage the layout and positioning of game screen elements such as icons and bars.<br>Ensure that interactive elements like settings, hearts, coins, collection, and game stats icons are properly placed and accessible.<br>Maintain visual consistency and alignment of elements on the game screen.<br>Adapt the positioning and sizing of elements to different device screens for a consistent user experience. | Game screens<br><br>Assets and Images |

Relevant user stories: N/A – Applied throughout the game flow

| Class: backgroundMusic | |
|---|---|
| Responsibilities: | Collaborators |
| Set up the audio mode settings to ensure compatibility with various device settings.<br><br>Select a random song from the predefined list and play it, ensuring it's different from the last played song.<br><br>Handle the end of a song and trigger the next song. | |

Relevant user stories: US41-V1

### 3.5.7. Team Stand-up

#### 3.5.7.1. Team Stand-up 1

| | |
|---|---|
| Date and Time | 18th Nov 2023, 12:00 noon |
| Location | Microsoft Teams |
| Attendees | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |
| Meeting | **Platform change:**<br><br>**Favour** has experimented with JavaScript, successfully building a game skeleton with React Native. The programme now has a main screen with features yet to be built. After briefly going through the programme code structure, the team agreed to move to JavaScript with React Native.<br><br>As React Native supports Android game development, the game can later be hosted on the Android app store.<br><br>The code file has been pushed to GitHub and everyone has been added as collaborators.<br><br>**Marat** went through some app logo assets to be added to main screen header.<br><br>**Marat** went through a PowerPoint deck collating general information on house plants, plant care, and a selection of plant-specific knowledge for the trivia.<br><br>**Ivan** shared an online course to get up to speed with React web development.<br><br>Everyone except for Favour and Ivan are still completing coursework for CM50260.<br><br>Everyone has committed to downloading the file and understanding the code as soon as the coursework is submitted over the weekend.<br><br>**New process change:**<br><br>Favour will now move to lead the code development stream. |
| Task allocations | **Everyone** should download the code file and run it locally with Expo for phone testing.<br><br>**Everyone** should try to get familiar with the React framework.<br><br>**Ivan** and **Favour** will work on completing a functional main screen and some basic core features before dividing up coding tasks for everyone to complete. |
| Meeting reflections and analysis | **Analysis / reflections of current standpoint:**<br><br>This meeting marks a significant change in the project trajectory. It was evident from the discussions that everyone is on board with the benefits of using a standard front-end suite rather than the Unity game engine. The intuitive features of React gave everyone reassurance that from this moment on the team would be able to make steady progress with code development over the upcoming weeks, particularly with the unforeseen mid- |

| | |
|---|---|
| | term crisis of overlapping assignments due.<br>The fact that Favour was already a highly proficient user in JavaScript and React played a huge part in pushing the decision through. Whilst the initial decision to use Unity was that it has a variety of ready-to-use functionalities for game development, the team failed to consider two drawbacks. First, although several people are proficient with C#, it only a fraction of what the team needs to know to master Unity as a game development engine, the intricacies of which no one has solid industrial experience with. This means the absence of someone who can confidently oversee development and give reassurance pertaining to programme functionalities and code quality. Second, although Unity is a professional engine for game development with a plethora of advanced features, most of them are not required for the purpose of building the plant learning game. Hence the initially proposed benefits of Unity may not be fit for context in this circumstance.<br>**Priorities until next customer meeting:**<br>Everyone except Favour and Ivan to continue with CM50260 coursework and submit as soon as possible.<br>Favour and Ivan to build a skeleton prototype and a code framework for task division. |

### 3.5.7.2. Team Stand-up 2

| | |
|---|---|
| Date and Time | 21st Nov 2023, 12 noon |
| Location | Microsoft Teams |
| Attendees | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |
| Meeting | **Updates on task-wise progress:**<br>**Favour** and **Ivan** shared with the team new features built over the past two days, completing most of what is asked of a basic game prototype.<br>GitHub merge conflict was flagged as a major reason that had slowed down progress.<br>**Ivan** is now working on the "Save" function, which saves game status every time user closes the app.<br>**Everyone** was able to download the file and run on their phone with Expo. |
| Task allocations | Those who had just completed CM50260 to get some sleep and prepare for an intense week ahead.<br>Everyone to continue learning React before Wednesday where Favour will go through the code files in detail. |
| Meeting reflections and analysis | **Current standpoint:**<br>As of this meeting, all midterm distractions from other modules have been completed, and the team expects to enter two weeks of high-intensity game development. It is imperative that over the next two sprints, coding tasks are appropriately allocated such that all core functionalities are complete, leaving only minor alterations (e.g., styling and animations) to finish off before submission.<br>Meanwhile, the restructuring of the team is also having a risk of bringing chaos to future progress. From next week onwards the plan is for most of the team exclusively focusing on code development while Claire catches up with documentation tasks that have fallen behind. The danger of this is that progress on product documentation will be paused as those working on code have no other choice but to dive into it in absence of user stories |

| | |
|---|---|
| | and use cases as guidance. Therefore, the uncertainty persists as to how the team will be delivering the product documentation components in later weeks.<br>**Priorities until next customer meeting:**<br>Learn JavaScript and React Native in preparation for collaborative code development. |

### 3.5.8. Exception Handling

| Exception | Actions | Outcome |
|---|---|---|
| Carried over from Sprint 4 | | |
| No available ready-to-use Unity templates for the game design. | **Sprint 5 actions:**<br>Switched to JavaScript and React Native.<br>**Sprint 4 actions:**<br>The main reason for choosing Unity was to take advantage of available templates. This implies that the entire game needs to be coded from scratch, which the team is not yet prepared for. | Success |
| CM50258 and CM50259 coursework due dates overlapping with Sprint 4. Many members are struggling with overlapping assignments. | **Sprint 5 actions:**<br>All mid-term coursework complete.<br>**Sprint 4 actions:**<br>As most members come from non-computer science backgrounds, the issue seems to be affecting most if not all members of the team. Everyone is trying their best to make piecemeal time for this project, and it is evident that everyone is still able to deliver on a good proportion of their allocated tasks. However, this still inevitably means that Sprint 4 has put the team behind on progress. All peers have committed to spending significant blocks of time to the project as soon as other deadlines are cleared. | Success |
| Unity proving to be much more difficult than everyone had previously imagined. | **Sprint 5 actions:**<br>Switched to JavaScript and React Native.<br>**Sprint 4 actions:**<br>All members have downloaded Unity by end of Sprint 4, yet no one was yet able to become a self-taught Unity user. This is a combined result of overlapping deadlines as well as the difficulty of Unity itself for anyone with no prior experience in game development engines. The immediate plan is for Alec to run through basic functionalities of Unity on Wednesday and to have the whole team dedicate a week to Unity training. | Success |
| As a result of the three exceptions above, the team has failed to deliver a working game demo as per customer request. | **Sprint 5 actions:**<br>Switched to JavaScript and React Native.<br>**Sprint 4 actions:**<br>The team recognises the urgency but there was not much anyone can do since no one would be willing compromise on CM50258 or CM50259, both of which are core units in the programme. It was agreed that as soon as competing deadlines are finished, everyone must commit to spending longer periods on this project | Success |

| | | |
|---|---|---|
| | to compensate for Sprint 4. | |
| **New exceptions raised** | | |
| Falling behind with code development | The team reflected on the reasons behind slow progress and discovered that the lack of experienced Unity users was a significant contributor. The team proposed JavaScript and the React Native framework as an alternative toolkit. This proposal acknowledges that both Favour and Udit were already proficient in them with industry experience, and that JavaScript and React were much simpler and more intuitive to pick up in a short period of time as beginners.<br>Favour played a key role in overseeing the transition by building skeleton code while everyone was still struggling with on other course assignments. Between himself and Ivan, code development progress caught up to the expectations of the customer. | Success |
| Change of team structure and roles | Previously Favour and Claire were co-leading the documentation workstream. Since Favour has moved to be the oversight of coding, Claire is now solely in charge of keeping documentation. In upcoming weeks, especially by the time people wrap up code development work, the team has to split up documentation tasks again in some way given the heaviness of the workload. | Pending |
| Falling behind on documentation. | As Claire was amongst those struggling with CM50260, documentation was put on hold throughout Sprint 5. The process documentation so far contains only meeting minutes and backlog items, with substantial content still awaiting completion. On the product documentation front, user story development is also at an elementary stage, with only a handful of stories elaborated with use cases and test requirements. UI designs have been completely revamped after moving to React Native, hence needs recreation. As the team has officially kicked-off game development, it is crucial that the relevant documentation is in pace with feature delivery. This must be resolved in Sprint 6. | Pending |
| iOS app store charges licence fees and shortage of Android phones amongst the team to test the app | As the project has changed to JavaScript with React Native, the Expo app allows iOS users to test the effects of an Android app on iPhones. Hence the team is now building an Android app which can be hosted on the Android app store without financial implications. | Success |
| Continued commitment to other coursework assignments | Before start of Sprint 5, everyone presumed that the CM50260 coursework could be completed before the weekend, giving everyone sufficient time to start on coding. Hence the reality that four out of seven peers were struggling all the way to evening of the 20th (Monday) was certainly unforeseen. However, it was extremely fortunate that Favour (who is most experienced in JavaScript and React) was able to submit the assignment early and held the team together by building up skeleton code and game prototype (with the help of Ivan in the second half of the sprint).<br>From now until the project due date, there is one more assignment due on 14th December from course CM50258. With | Success |

| | | |
|---|---|---|
| | the lessons learnt, the team unanimously agreed that everyone should aim to start on the assignment as early as possible to avoid last-minute disruptions. | |

### 3.5.9. Sprint 5 Backlog and Completion Status

| TASK | USER STORY | DUE | PRIORITY | ASSIGNED TO | STATUS |
|---|---|---|---|---|---|
| Investigate JavaScript and React as alternative development suite. | | 18 Nov | High | All | Complete |
| Stripped down coded demo of game interface and core functionalities | See Sprint 5 User Story List | 22 Nov | High | Favour, Ivan | Complete |
| Research and develop trivia questions for the various plant difficulties, with more questions depending on the difficulty. | | 22 Nov | High | Marat | Complete |
| Find templates/tutorials for working with C# and Unity for mobile game development. | | 22 Nov | High | Ivan, Alec | Deleted |
| Add user stories/use cases template to Sprint Documentation. | | 22 Nov | High | Claire | Pending |
| Update user story list with latest game design and add acceptance criteria & story points | | 22 Nov | High | Claire | Pending |
| Photoshop background from assets and clean up assets. | | 22 Nov | High | Claire | Complete |
| Research unity and learn how to work with its interface | | 22 Nov | High | All | Deleted |
| Divide up coding assignments to other members of the group. | | 22 Nov | High | Marat | Pending |

Notes to completion status:

- Items relating to Unity have been deleted.
-

### 3.5.10. Individual Retrospectives

Each member reflected on their experience during Sprint 5 by completing the individual weekly reflection template, key points are summarised as follows.

On positive notes, everyone first and foremost has highlighted the ground-breaking progress in code development. This has been attributed to the timely decision to move from Unity to JavaScript, Favour's dedication to coding up the groundwork while everyone else is still working on the CM50260 assignment and the teamwork between himself and Ivan. Second, as the team became more involved in using Git, everyone commented on the ease of real-time collaboration and record keeping the tool has brought, particularly with the functionalities of posting commit messages. Both Favour and Ivan have stressed on the importance of Git in helping the two of them collectively work on different features at the same time during the second half of the sprint.

Nonetheless, several factors have curbed what may have been even better progress that what the team has achieved. First, everyone else besides Favour and Ivan has mentioned the fact that the CM50270 coursework to have occupied significantly more time than they had initially estimated due to the difficulty of materials. The team has been very understanding to the circumstances, given that mastering pure mathematical concepts can vary in difficulty for different people. Despite so, almost all

reflections were sincerely apologetic of their limited time commitment to the project this week. Second, several peers voiced concerns of their unfamiliarity with React and/or JavaScript but recognises that it is a much more intuitive toolkit to learn compared to Unity.

Third, in Favour and Ivan's reflections both mentioned merging conflicts on GitHub to be one of the biggest hurdles that wasted a lot of time, stressing the importance of frequent commits and checking that nothing is broken before making a "push".

Overall, contribution was highly skewed for Sprint 5 measured in terms of hours spent on the project, ranging from single digits to up to 40 hours. Recognising that this was the combined result of CM50260 assignments and different levels of proficiency in JavaScript and React development. Those who has not had many opportunities to contribute to Sprint 5 has committed to learning the tools and jump into coding in Sprint 6. The table below summarises how everyone is feeling out of a scale of 1 to 5 (1=worst, 5=best) about this Sprint as well as the project overall. It seems that most members hold positive thoughts toward the progress so far, and lower ratings in both cases are due to the worry of being potentially incapable of contributing enough to the code due to no prior experience in JavaScript.

| Band | Sprint 5 | Project |
|---|---|---|
| 5 (Best) | 4 | 1 |
| 4 | 2 | 5 |
| 3 | - | 1 |
| 2 | 1 | - |
| 1 (Worst) | - | - |

## 3.6. Sprint 6: 22nd November – 29th November

### 3.6.1. Overview

In Sprint 6, the team achieved a significant milestone by delivering a substantial portion of programmes and features. As Sprint 5 laid the foundations for component-wise code building, the target for Sprint 6 was to efficiently distribute development tasks for members to first conquer and then integrate. The team anticipated having all key game features in place by conclusion of Sprint 6, allowing the remaining weeks before December 15th to be dedicated to finalising the project in preparation for submission. Meanwhile, since progress on documentation had fallen behind in Sprint 5, efforts were refocused towards writing product documentation and completing process documentation for at least a full sprint from start to end (to be reviewed in next week's TA meeting). Specifically, the team acknowledges the urgency to structurally develop user stories in detail. So far only a small number of user stories relating to the main screen have been substantiated, with the rest still pending acceptance criteria, use cases, UI design and test requirements. The challenge at hand lies in contextualising the initial list of user stories Alec compiled in Sprint 1 in the real game prototype built in the last sprint. As Favour and Ivan had started code development in isolation when the rest of the team had other commitments, several features were built upon intuition rather than systematic guidance of acceptance criteria and use cases, meaning that some of these elements required in the product documentation needs to be reverse engineered. Overall, the team expected Sprint 6 to be one of rapid development and delivery, emerging as perhaps the most engaging and intensive sprint to date in terms of both individual and collective efforts.

### 3.6.2. Review

At the planning meeting, aligning with the React Native structure Favour had developed in Sprint 5, a list of programming tasks was allocated to each member. Tasks of varying difficulty was assigned mindful of everyone's levels of experience with JavaScript and React. Progress was steady and linear across the week. A few challenges were identified in the first stand-up after everyone has spent some time on their tasks. This positively triggered closer collaboration between members of the team where some of the more complex pieces were successfully tackled in a collective manner. Towards the end of the sprint, all the backlog items were complete, yielding some time for additional features for enhanced user-friendliness.

On the documentation side, a complete sprint process documentation was written timely and aligned with the structure proposed in Sprint 4, ready for review at the TA documentation clinic. User story development, on the other hand, appear to be a greater challenge. The short period remaining until submission meant that the development team does not have time to write use cases before writing code. Instead, everyone dived straight into coding, and made decisions on the spot based on immediate trial and error. Only informal records were made of these on-the-spot decisions, leaving a void in what should have been a structured user story file with clear version controls. Moreover, in the attempt to rush progress on all fronts, in Sprint 6 product documentation (Claire) and product development (everyone else) was separated into two un-overlapping streams, hindering communication between the two. The outcome was that product documentation was having severe difficulties keeping up with progress in product development as new codes and functionalities are constantly being built and revised throughout the week. In response, the team agreed in the second stand-up that from Sprint 7 onwards, documentation should be escalated to a top priority since game development is already approaching the closing phase.

### 3.6.3. Summary of Key Events

- Completion of majority of core game features.
- Close collaborative programming on complex functionalities.
- Completion of a full Sprint (Sprint 3) process documentation ready for TA review.

- Product documentation falling behind due to rushed code development and isolated workstreams.
- Unanimous agreement of shifting focus to documentation from the next sprint.

### 3.6.4. Sprint Preparation: Customer Meeting and Planning Discussion

| | |
|---|---|
| **Date** | 22nd Nov 2023 |
| **Location** | Campus – Chancellor Building 4.7 |
| **Attendees** | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |
| **Customer Meeting** | **Presentations:**<br>*"*" indicating that customer has made a comment, recorded in the "Customer concerns" section below.*<br>• **Completed features:**<br>  • **Home screen page:** Oracle the sun on main screen gives instructions throughout the game and asks questions in the quiz / trivia*.<br>  • **Place a plant:** Can place a plant from inventory of plants. Currently the plant grows as the player moves up mastery levels. Planning to implement a timed growth for all plants of the same species once the player completes all mastery of a specific plant.<br>  • **Plant option menu:** Visible after a plant (added to main screen) is selected.<br>  • **Demo of Trivia:** Player selects level, enters the quiz screen. Every time a wrong answer is selected a prompt is given informing player of wrong selection, until a correct answer is selected from which the quiz moves onto the next question.<br>  • **Game Stats:** Shows how much of the mastery the player has completed in the form of cards, e.g., passing 2 levels out of 3 would show 66.7% completion. Once the player clicks on a plant card on this screen, a "textbook" containing key knowledge of that plant is shown as an important information source.<br>  • **Watering and Fertilising:** Once the player has reached mastery, a timer will start based on which notifications are given when the plant needs to be watered or fertilised.<br>  • **Settings button:** so far containing option for music on/off.<br>• **Incomplete features:**<br>  • **Shop functionalities (yet to be implemented)**<br>  • **Skins:** Replacing creative skins (e.g., pink) with scientific variations of the same plant as icons, available to purchase at the shop.<br>  • **Background:** Option to purchase different backgrounds and rooms with accumulated coins (currently at single-background mode the plant can be archived if requiring more space.<br><br>**Customer concerns and team responses:**<br>• Customer asks to clarify on role of the oracle in the trivia.<br>  Response: currently the oracle asks questions, returns "Incorrect" when selection is wrong. |

| | |
|---|---|
| | • Customer asks for explanation on how mastery level works.<br>Response: every plant has a certain number of mastery levels. Once all levels are completed (quizzes passed) the player reaches mastery. Every time a quiz is complete the plant in main screen "grows".<br>• Customer asks whether any prompts will be given when on how much water is needed for a real-life plant when an in-game plant is fertilised.<br>Response: After player links to a real-life plant and set its level, the plant will stay at that level (visually the image won't "grow"). From then on, every time a notification is sent to water/fertilise the plant, it is accompanied with instructions on how much water/fertiliser is needed, based on the plant level.<br>• Customer asks if information is available to players to help determine what level / growth stage a real-life plant is at. **[A1]**<br>Response: Something the team will investigate in the next sprint. |
| Planning Discussion | **Additions/changes to product design based on customer meeting:**<br>**[A1]** Add information prompt at "link" screen to help user determine the level of their real-life plant.<br>**Analysis / reflections of current standpoint:**<br>For the past two sprints the team encountered various difficulties with product delivery, such that the lack of progress had escalated to be a major risk to meeting deadlines. Therefore, with all obstacles cleared, Sprint 6 is anticipated to have everyone engaged at full capacity to offset previous delays and bring progress on track. As most of the team immerse themselves in game development, there remains uncertainty around how documentation efforts will be effectively divided, especially with everyone developing game features in absence structured user story guidance. Acknowledging that user stories, use cases and UI designs are typically foundational elements required before starting code development, their absence adds complexity to writing structured product documentation moving forward. Nonetheless, with the urgency of delivering a product in just two weeks' time, the agreement was to shift focus to documentation only after we have something to deliver. Hence for the time being the priority is to have as many people onboard as possible with coding.<br>Moreover, as not everyone is familiar with JavaScript and React Native, the team should be prepared for closer collaboration when building code. This is exacerbated by the fact that many functionalities and components assigned to different members are linked to each other in the overall game flow. Everyone should thus keep up to date with latest messages in the group chat. "Pushes" should not be made unless additions have been thoroughly tested. More importantly, peer review should be incorporated into the coding process such that code quality is ensured to at least a baseline level.<br>**Determining Sprint priorities:**<br>With the high-level code structure in place, the team discussed how to best segment development tasks between members. In line with customer requirements, the team divided coding tasks into two broad categories: new functionalities and existing functionalities (which require enhancements).<br>**Developing New Functionalities**<br>These are functionalities that are not yet set up as part of the game prototype. Including but not limited to: Adding separate rooms to hold more plant space, link to real plant, progress timers starting when player achieves mastery, collection page, etc.<br>**Enhancing Existing Functionalities**<br>These are enhancements that are yet to be made to existing functionalities already incorporated in the stripped-down prototype presented in the customer meeting. Including but not limited to: Adding a "Hearts" system for the quiz recording the number |

| | |
|---|---|
| | of incorrect selections allowed, developing game tutorial to be given by the Oracle first time a player installs the game, saving function for state variables to user device for game continuity, adding more plant variety and quiz question to the game database.<br><br>The team made a full list of tasks which were allocated on a voluntary basis, where everyone picked those which, they deemed manageable for their programming experience. As the coding stream oversight, Favour also voluntarily took on several additional responsibilities that helps ensure code efficiency and functionality at the product level. On documentation, Claire will prioritise user story development and produce a complete example of Sprint process documentation. The team agreed that the week ahead will be demanding, especially with parts of the team having limited experience with React Native and JavaScript. Thus, it's important that if anyone is struggling to raise it as soon as possible so that others can step in to assist.<br><br>The Sprint 6 backlog can be found in Section 3.6.9. |

### 3.6.5. User Stories, Acceptance Criteria (and Tests), Use Cases, and UI Design

In Sprint 6, the following user stories have been developed and built into the game prototype.

Note that parts of the stories below are reverse engineered in Sprint 7 (after the features are already developed into the game) as an unfortunate outcome of the team's failure to sequentially implement the software process model. Therefore, some UI designs were developed without any wireframes as guidance, in which cases a snapshot of the UI in the end-product is presented for illustration.

Full list of user stories relevant to Sprint 6.

*Overwritten versions are greyed out

| User Story ID | Description |
|---|---|
| US4-V2 | Displaying main screen elements |
| US46-V2 | Expand main game menu. |
| US25-V2 | Level selection screen. |
| US51-V1 | Select a level. |
| US2-V1 | Save progress automatically |
| US3-V1 | Showing loading screen before game starts to ensure all saved game data is loaded. |
| US6-V1 | Moving plants in collection (archive) back to virtual room |
| US7-V1 | Displaying number of coins with the coin icon. |
| US11-V1 | Displaying number of hearts with the heart icon. |
| US12-V1 | I want to enter and view the collection (archive) by pressing a button in main screen. |
| US13-V1 | I want to view all archived plants after entering the plant collection. |
| US14-V1 | Perform plant actions to plants in collection. |
| US16-V1 | See plant information for a plant by tapping a plant card in mastery level screen. |
| US18-V1 | Archive plant with "archive" option in plant action menu. |

| | |
|---|---|
| US19-V1 | Enter link screen to link an in-game plant to a real plant with the link button in plant action menu. |
| US20-V1 | Add photo for linked plant. |
| US21-V1 | Edit details of linked plant. |
| US50-V1 | Receive instructions on editing details of linked plants. |
| US23-V1 | Water the plant by tapping on the “water” button in plant action menu. |
| US24-V1 | Fertilise the plant by tapping on the “fertilise” button in plant action menu. |
| US24-V2 | Fertilise the plant by tapping on the “fertilise” button in plant action menu. |
| US57-V1 | Lose “hearts” when selecting incorrect answer in the quiz. |
| US35-V1 | Game over after all “hearts” are lost. |
| US27-V1 | Finish level. |
| US27-V2 | Finish level. |

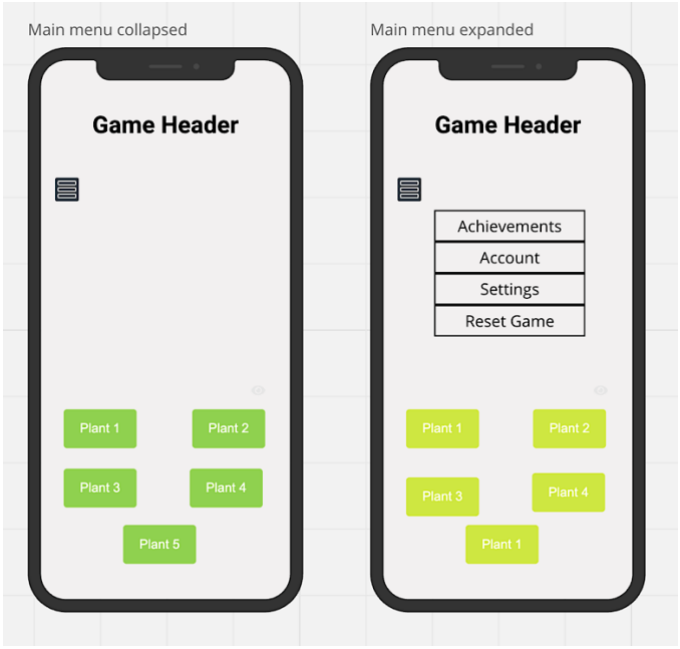
<table>
<tr><td>US4-V2</td><td colspan="3">Displaying main screen elements</td></tr>
<tr><td>Description</td><td colspan="3">As a player, I want to open the game app and see the main screen. The main screen should have the following elements: game header, a sun-shaped oracle, collapsed main menu icon, player's game status (number of "hearts" and "coins"), plant mastery icon, collection icon, a background room, in-game plants the player currently has in room, XP bar, and any empty placeholder spots for new plants.</td></tr>
<tr><td rowspan="3">Acceptance Criteria</td><td>Criteria ID</td><td>Description</td><td>Test Status</td></tr>
<tr><td>US4-V2-AC1.</td><td>I can see the main game page when I enter the game app.</td><td>Pass</td></tr>
<tr><td>US4-V2-AC2.</td><td>The main page shows all listed elements above.</td><td>Pass</td></tr>
<tr><td>Requirement Use Case</td><td colspan="3">Use Case: Loading main game screen and its elements.<br>Scope: Main screen.<br>Level: Basic game set-up.<br>Context:<br>(1) When the user arrives at the main screen, they should be able to see a variety of game elements, some of which can navigate to other game features screens.<br>(2) These elements are game header, a sun-shaped oracle, collapsed main menu icon, player's game status (number of "hearts" and "coins"), plant mastery icon, collection icon, a background room, in-game plants the player currently has in room, XP bar, and any empty placeholder spots for new plants.<br>Frequency of occurrence: Every time player is at the main screen, either just after they open the game app, or after then return to main screen from other screens.<br>Open issues: The set of objects rendered on the main screen may be subject to changes.</td></tr>
<tr><td>Design Use Case</td><td colspan="3">Scope: Main screen.<br>Level: Basic game set-up.</td></tr>
</table>

| | |
|---|---|
| | **Primary actors:** Player<br>**Description:** Player can see a set of game elements on the main screen.<br>**Dependencies:**<br>(1) The user has successfully installed the game, and able to open the game to the main game screen.<br>(2) Game assets (images and icons) are added appropriately to code file.<br>**Assumptions:** N/A<br>**Preconditions:** N/A<br>**Main Flow:** After the user arrive at main screen (either by opening the app, or returning to main screen from other game screens), the user should be able to see a number of game components, including: game header, a sun-shaped oracle, collapsed main menu icon, player's game status (number of "hearts" and "coins"), plant mastery icon, collection icon, a background room, in-game plants the player currently has in room, XP bar, and any empty placeholder spots for new plants.<br>**Post conditions:** All elements listed above are visible on the main screen and positioned according to UI design.<br>**Alternative Flows:** One or more of the components are not visible or unaligned with the UI design (invalid)<br>**Frequency of occurrence:** Every time the player is at the main screen.<br>**Open issues:** The set of objects rendered on the main screen may be subject to changes. |
| UI Design | **Key UI components:**<br>Main game screen,<br>Game header,<br>Sun-shaped oracle<br>Collapsed main menu,<br>Player's game status ("hearts" and "coins"),<br>In-game plants and placeholders,<br>XP bar,<br>Collection icon,<br>Plant mastery icon<br>**Wireframe:** |

| | |
|---|---|
| |  <br>**Interaction Flow:**<br>User arrives at the main screen with all elements displayed as expected.<br>**Data Entry and validation:** N/A<br>**Error Handling:** N/A<br>**UI Behaviour:** Components should take roughly similar time to load with no significant lagging.<br>**Visual Design Guidelines:**<br>Styling should align with the game's defined typography and colours.<br>Elements should be able to stand out from the background image colour / patterns.<br>All elements should be clearly visible without overlap. |

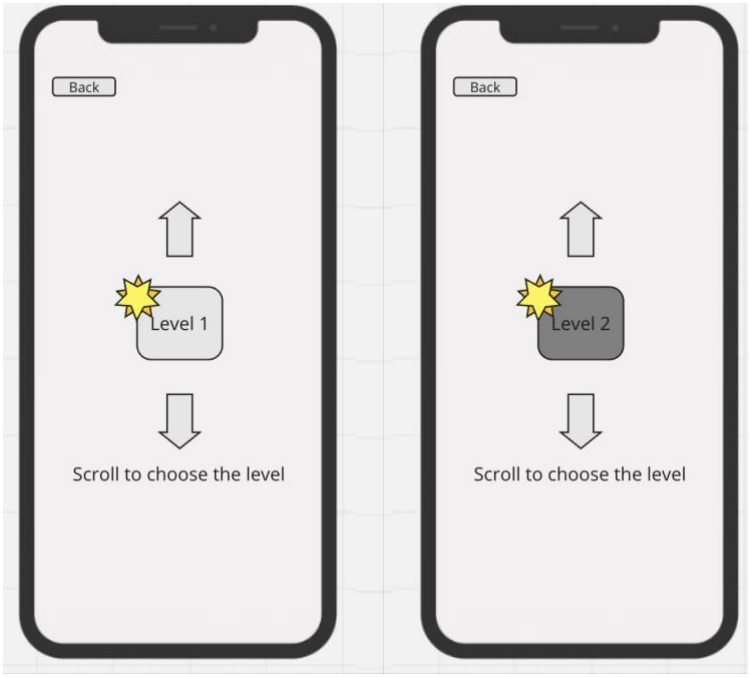
| US46-V2 | Expand main game menu. | | |
|---|---|---|---|
| Description | As a player, I want to expand the main game menu when I select the menu button on the main page. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US46-V1-AC1.** | After selecting the collapsed menu widget, the main game menu should expand and show on the main page. | **Pass** |
| | **US46-V1-AC2.** | The main menu has the following options: Achievements, ~~game stats, mastery level, shop,~~ account, settings, and clear data. | **Pass** |
| | **US46-V1-AC3.** | The menu should go back into archive if I click anywhere else on the screen. | **Pass** |

| | |
|---|---|
| Requirement Use Case | **Use Case:** Accessing main game menu on main screen.<br>**Scope:** Standard gameplay.<br>**Level:** Core game feature.<br>**Context:**<br>(1) When the user arrives at the main screen, they can enter other game screens through the main menu.<br>(2) The menu is represented by a menu widget in the shape of three horizontal lines on the main screen. The default menu state is collapsed, meaning that the user cannot see the options by default when entering main screen.<br>(3) The menu is un-collapsed when user clicks the menu widget, from which the user can select from a series of options.<br>**Frequency of occurrence:** Every time the user is at the main screen and wishes to access main menu options.<br>**Open issues:** N/A |
| Design Use Case | **Scope:** Standard gameplay<br>**Level:** Core game feature.<br>**Primary actors:** Player<br>**Description:** Player can view and choose from the main game menu on the main game screen.<br>**Dependencies:**<br>(1) The user has successfully installed the game, and able to open the game to the main game screen.<br>(2) The user has successfully returned to the main screen from other screens in the game.<br>**Assumptions:** N/A<br>**Preconditions:** Player has been able to successfully install and run game on a mobile phone.<br>**Main Flow:** After the user arrive at the main screen (either by opening the app, or returning to main screen from other game screens), the main menu widget should be visible on the upper left side of the screen. The user should be able to un-collapse the menu by clicking on the widget icon.<br>**Post conditions:** The user should be able to see the expanded main menu show up in the centre of the screen. The menu should include the following options: Achievements, ~~game stats, mastery level, shop,~~ account, settings, and clear data. Player can either select an option from the expanded menu to go to another game screen or collapse the menu by clicking anywhere else on the screen.<br>**Alternative Flows:**<br>**(1) Valid:**<br>Player arrives at main screen but does not wish to view main game menu. Instead, player proceeds to other functionalities (e.g., shop, view mastery, plant actions).<br>**(2) Invalid:**<br>Player arrives at the main screen but cannot see the menu widget.<br>Player clicks on the menu widget, but menu does not expand.<br>Player clicks on the menu widget; menu expands but are not properly displayed.<br>**Frequency of occurrence:** Every time the user is at the main screen and wishes to access main menu options. |

| | |
|---|---|
| | **Open issues:** N/A |
| UI Design | **Key UI components:**<br>Main game screen<br>Main menu widget icon (collapsed icon in the shape of three horizontal lines).<br>Expanded menu.<br>**Wireframe:**<br>  <br><br>**Interaction Flow:**<br>User clicks on the Main menu icon button on main screen.<br>Expanded menu appears on screen in a list.<br>User can select any item in the list to go to the corresponding game page.<br>User can click anywhere on the main screen to collapse the menu and stay at main page.<br>**Data Entry and validation: N/A**<br>**Error Handling: N/A**<br>**UI Behaviour:**<br>The main menu should have a smooth animated effect when expanded and collapsed.<br>The menu should by default be collapsed on the main screen,<br>The collapsed menu icon should always be placed along the upper left edge.<br>**Visual Design Guidelines:**<br>Styling should align with the game's defined typography and colours.<br>The collapsed menu icon should be visible, not obstruct any widgets on the main screen.<br>The expanded menu should be displayed at the centre of the main screen.<br>The expanded menu option list should not fill up the entire main screen, leaving space for the user to click if they want to collapse the menu and return to main game.<br>When the expanded menu is displayed, the main game screen background colours should be dimmed, enabling the menu to stand out to the player. |

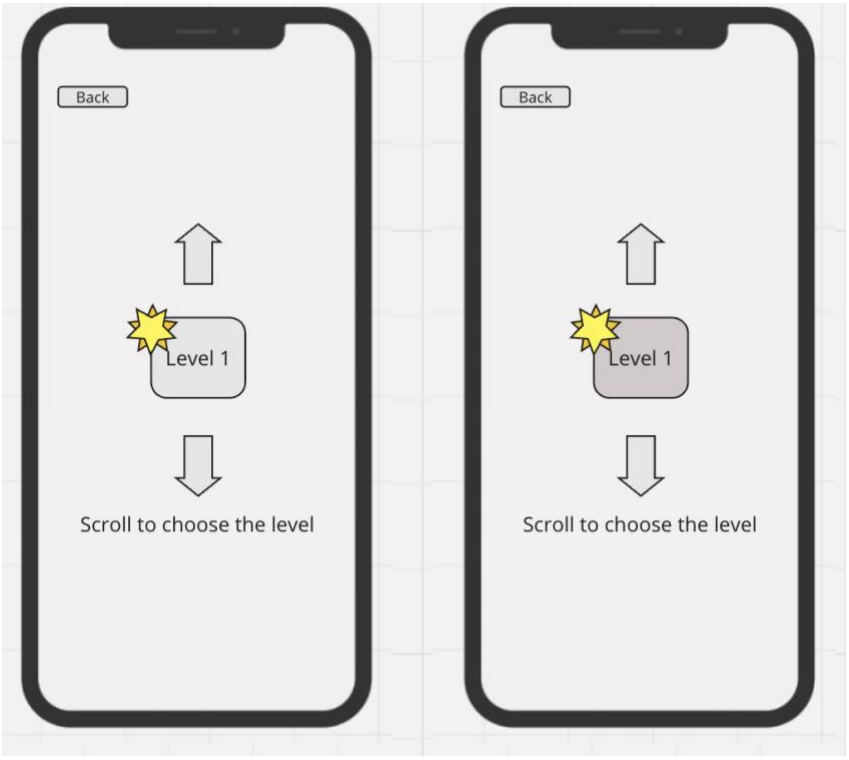
| | |
|---|---|
| | There should be enough spacing between each option for the player to choose and make selection. |

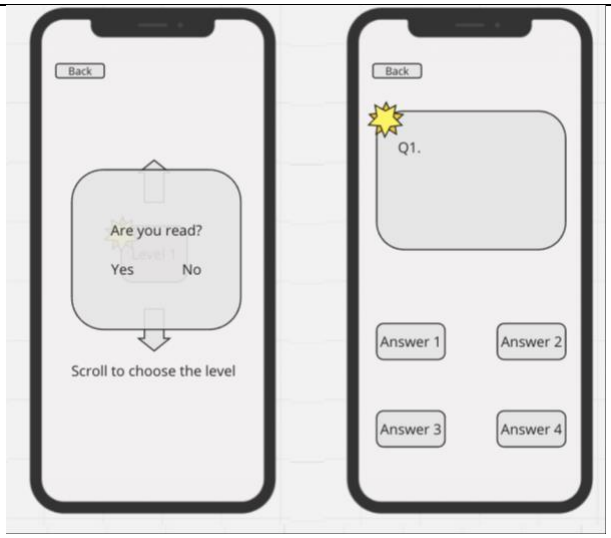
| US25-V2 | Level selection screen. | | |
|---|---|---|---|
| Description | I want to enter the level selection screen via the quiz button and see the level selection screen. The level selection screen should have the following elements: header back button and all available level buttons (unlocked/locked). | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US25-V2-AC1.** | I can see the level selection screen when I enter the level selection screen via the quiz button. | **Pass** |
| | **US25-V2-AC2.** | The level selection screen shows the correct levels of the selected plant species. | **Pass** |
| | **US25-V2-AC3.** | The only available level when the player first enter the level screen would only be level 1. | **Pass** |
| | **US25-V2-AC4** | The level selection screen should show the latest available level to the user. | **Pass** |
| Requirement Use Case | **Use Case:** Loading level selection screen and its elements.<br>**Scope:** Quiz feature.<br>**Level:** Core game feature.<br>**Context:**<br>(1) When the user arrives at the level selection screen, they should be able to see a variety of level selection screen elements from which to perform actions within the level selection screen.<br>(2) These elements are back button and all available level buttons (unlocked/ locked).<br>**Frequency of occurrence:** Every time player is at the level selection screen, accessing from the quiz button in the plant menu.<br>**Open issues:** The set of objects rendered on the level selection screen may be subject to changes.<br>The style of how to show the elements may change. | | |
| Design Use Case | **Scope:** Quiz feature.<br>**Level:** Core game feature.<br>**Primary actors:** Player<br>**Description:** Player can see a number of game elements on the level selection screen.<br>**Dependencies:**<br>(1) The user has successfully installed the game, and able to open up the game to the main game screen.<br>(2) The user could add a plant and could access both the plant menu and the quiz button.<br>(3) Game assets (images and icons) are added appropriately to code file.<br>**Assumptions:** N/A<br>**Preconditions:**<br>(1) The user has added at least one plant to their main menu. | | |

| | |
|---|---|
| | **Main Flow:**<br>After the user has added a plant, the user should be able to click on the plant and open the plant menu. They have the option to choose to enter the level selection screen by clicking the quiz button within the plant menu.<br>**Post conditions:** All elements listed above are visible on the level selection screen.<br>**Alternative Flows:**<br>**Invalid:**<br>(1) One or more of the components are not visible or unaligned with the UI design.<br>(2) The user could not enter the level selection screen.<br>**Frequency of occurrence:** Every time the user enters the level selection screen.<br>**Open issues:** The set of objects rendered on the link screen may be subject to changes. |
| UI Design | **Key UI components:**<br>Level selection screen<br>All available levels for the selected plant<br>Back button<br>**Wireframe:**<br>  <br><br>**Interaction Flow:**<br>User enters the level selection screen.<br>User sees their current level and all levels for the selected plant.<br>**Data Entry and validation:** N/A<br>**Error Handling:** N/A<br>**UI Behaviour:** The back button should by default appeared on the level selection screen.<br>**Visual Design Guidelines:**<br>Styling should align with the game's defined typography and colours.<br>Locked levels should have a shade when comparing to the unlocked levels.<br>The oracle should be at the top left corner of the level button. |

| | |
|---|---|
| | The latest level unlocked should by default appear on the level selection screen.<br>The scroll mechanics should be smooth. |

| US51-V1 | Select a level. | | |
|---|---|---|---|
| Description | As a player I want to select an available level before taking a plant quiz. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US51-V1-AC1.** | After pressing the level button, a pop-up menu will be shown. | **Pass** |
| | **US51-V1-AC2.** | The pop-up menu contains a message to confirm the selected stage, 2 buttons "yes" or "no". | **Pass** |
| | **US51-V1-AC3.** | The user will enter the level if "yes", pop up menu will collapse if "no". | **Pass** |
| | **US51-V1-AC4** | If user selected “yes” they will proceed into the quiz. | **Pass** |
| | **US51-V1-AC5** | User can scroll vertically to choose from all available levels of the quiz. | **Pass** |
| | **US51-V1-AC6** | The highest level available is one above user’s current level for that plant. | **Pass** |
| Requirement Use Case | **Use Case:** Loading level selection screen and its elements.<br>**Scope:** Quiz feature.<br>**Level:** Core game feature.<br>**Context:**<br>(1) When the user presses the level button, the user could enter the level of the selected plant.<br>(2) The level button is shown as a part of the link screen by default.<br>(3) It should show the latest available level when the user arrives to the level selection screen.<br>(4) The player can enter a level of the selected plant. To perform this action, the player needs to first scroll through the levels and choose the desired level and confirm by clicking yes from the pop-up menu.<br>**Frequency of occurrence:** Every time the user opened the level selection screen and wishes to enter a level of quiz of the plant to take. | | |
| Design Use Case | **Scope:** Quiz feature.<br>**Level:** Core game feature.<br>**Primary actors:** Player<br>**Description:** Player can enter a desired level of quiz of the selected plant by clicking on the level button.<br>**Assumptions:** There is at least one level of quiz developed for that plant.<br>**Dependencies:**<br>(1) The user has successfully installed the game, and able to open the game to the main game screen.<br>(2) The user could add a plant and could access both the plant menu and the quiz button. | | |

| | |
|---|---|
| | **Preconditions:** The user has added at least one plant to their main menu.<br>**Main Flow:**<br>After the user arrive at the level selection screen (by clicking the quiz button of a selected plant), the user should be able to scroll to select a level and click on the level button, confirm whether he/she wants to enter the level or not. Alternatively, the user can also scroll to past levels and retake quizzes.<br>**Post conditions:** User is taken to the quiz of the level selected, and the first quiz question is rendered on screen.<br>**Alternative Flows:**<br>**Valid:**<br>(1) Player cancels the action by clicking no on the level confirmation menu.<br>**Invalid:**<br>(1) Player clicks on the level button, but the pop-up confirmation box does not show up.<br>(2) Player clicks on the level button, confirmed the level but cannot enter the quiz screen.<br>**Frequency of occurrence:** Every time the user opened the level selection screen and wishes to enter a level of quiz of the plant to take. |
| UI Design | **Key UI components:**<br>Level selection screen<br>Level button<br>Scroller to see all available levels for the selected plant<br>Back button<br>**Wireframe:**<br>  |




**Interaction Flow:**

User chooses a desired level.

User clicks on the level button of the desired level.

A pop-up menu is shown.

Users confirm whether this is the level of the quiz they want to do.

The game navigates to the quiz screen displaying the first question.

**UI Behaviour:**

The level selection scroller should have a smooth animated effect.

The pop-up should be shown with a smooth animation.

**Visual Design Guidelines:**

Styling should align with the game's defined typography and colours.

Guidance should be given to users on what they need to do (e.g., scroll, confirm yes/no).

| US2-V1 | Save progress automatically | | |
|---|---|---|---|
| Description | As a player, I want my progress to be automatically saved even when I exit the game. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US2-V1-AC1.** | Each time I exit the game, progress is automatically saved and reloaded when I reopen the game. | **Pass** |
| Requirement Use Case | **Use Case:** Returning player's latest game progress every time the game opens.<br>**Scope:** Game set-up.<br>**Level:** Single player and one local device.<br>**Context:**<br>(1) Exist of the progress saving system saves the game progress manually or automatically.<br>(2) The progress of the game automatically loaded each time the game is opened.<br>(3) The progress should be the latest.<br>**Frequency of occurrence:** Every time player quits and reopens the game. | | |

| | |
|---|---|
| | **Open issues:** N/A |
| Design Use Case | **Scope:** Standard gameplay<br>**Level:** Single player and one local device.<br>**Primary actors: In-built game configurations.**<br>**Description:** Each time player exits the game on a device, progress is automatically saved and reloaded when I reopen the game.<br>**Dependencies:**<br>(1) The user has successfully installed the game and can open up the game to the main game screen.<br>(2) The player did not reset the game progress.<br>**Assumptions:** The player made new progress in the game (e.g. added a new plant, levelled up a plant, established new link, etc.).<br>**Preconditions:** The player is using the same device.<br>**Main Flow:**<br>Every progressive change made by the player in the game is automatically stored in the AsyncStorage, which is the database storing any progress-related variables. The screen resets with the variables in the AsyncStorage when any changes are detected.<br>**Post conditions: T**he game is reopened after the quitting in player's device and the background storage of the game in the device was removed, the game screen load with the variables in the AsyncStorage.<br>**Alternative Flows:**<br>**Invalid:** Progress was not saved and reloaded, or partially saved and reloaded.<br>**Frequency of occurrence:** Every time the user quits and reopens the game.<br>**Open issues:** As the account feature is not yet implemented, **i**f player switches mobile device, progress will be lost. |

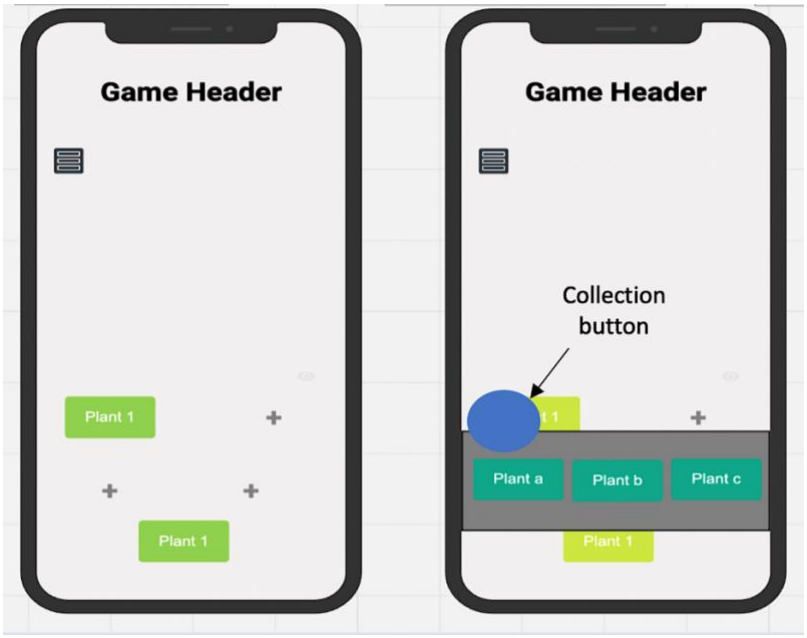
| US3-V1 | Showing loading screen before game starts to ensure all saved game data is loaded. | | |
|---|---|---|---|
| Description | As a Player, I want to see a loading screen before the actual game starts, so that I am assured that all necessary game data is being properly loaded and prepared for my gameplay experience. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US3-V1-AC1.** | A loading screen should appear immediately after the game is launched and before the main screen is displayed. | **Pass** |
| | **US3-V1-AC2.** | The loading screen should provide visual feedback, such as a progress bar or animation, indicating that the game is loading. | **Pass** |
| | **US3-V1-AC3.** | The loading screen should remain visible until all necessary game data, including assets and settings, is fully loaded. | **Pass** |
| | **US3-V1-AC4.** | The transition from the loading screen to the main game or menu should be smooth and without abrupt changes. | **Pass** |

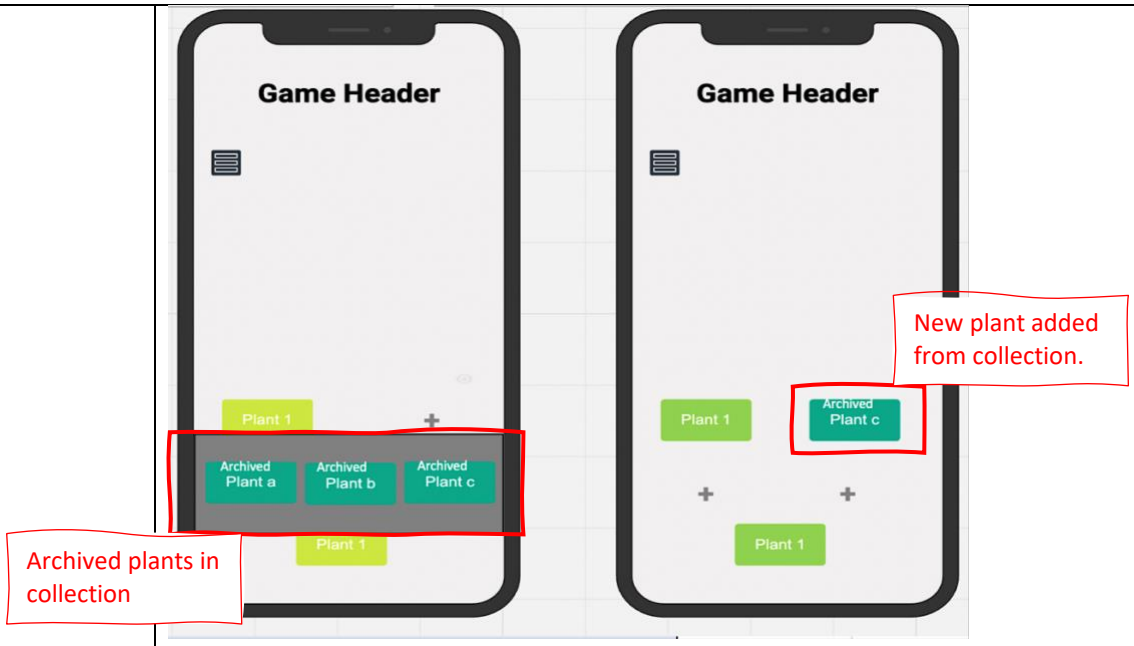
| | | | |
|---|---|---|---|
| | US3-V1-AC5. | The design of the loading screen should be consistent with the overall aesthetic of the game. | Pass |
| | US3-V1-AC6. | The loading screen should not stay on for an excessively long time, ensuring a quick transition to gameplay. | Pass |
| Requirement Use Case | **Use Case:** Displaying loading screen when opening the game.<br>**Scope:** Standard gameplay<br>**Level:** Initialization process<br>**Context:**<br>(1) The player starts the game application.<br>(2) The loading screen is displayed, showing progress in loading game data.<br>(3) The game completes the loading process and transitions to the main screen.<br>**Frequency of occurrence:** Each time the game application is launched or restarted. | | |
| Design Use Case | **Scope:** Game set-up<br>**Level:** Initialization process<br>**Primary actors:** Player<br>**Description:** Player can see a loading screen immediately when they open the game as saved game data is being loaded.<br>**Preconditions:**<br>(1) The game includes a functional loading mechanism.<br>(2) The game has assets and data that need to be loaded before starting.<br>**Main Flow:**<br>(1) The game displays a loading screen upon launch.<br>(2) The loading screen shows a visual indicator of progress.<br>(3) The game transitions from the loading screen to the game environment after loading is complete.<br>**Alternative Flows:**<br>**Valid:** The loading time adjusts based on the amount of data being loaded.<br>**Invalid:**<br>(1) Loading screen fails to display or shows incorrect information.<br>(2) The game starts before the loading process is complete. | | |
| UI Design | **Key UI components:**<br>Loading screen background or animation.<br>Progress indicator (e.g., progress bar or spinner).<br>Brief tips or messages related to gameplay (optional).<br>**Wireframe:** | | |

| | |
|---|---|
| |  |

| US6-V1 | Moving plants in collection (archive) back to virtual room | | |
|---|---|---|---|
| Description | As a player, I want to put my plants that were archived in the collection to the planting spot on the main screen. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US6-V1-AC1.** | After I tap an available location in the room, a plant inventory list with the choice of entering archive should appear on the screen. | **Pass** |
| | **US6-V1-AC2.** | I can scroll left and right to see all the plants available in the collection. | **Pass** |
| | **US6-V1-AC3.** | If I tap on a selected archived plant, it should be added to the selected placeholder spot in the room. | **Pass** |
| | **US6-V1-AC4.** | The added plant is accordingly removed from collection. | **Pass** |
| | **US6-V1-AC5.** | The collection and inventory list should then disappear as soon as I have chosen a plant. | **Pass** |
| | **US6-V1-AC6.** | The plant should remain in the spot for the rest of the game until I clear data or move it to the archive. | **Pass** |
| Requirement Use Case | **Use Case:** Moving plants in collection back to in-game room.<br>**Scope:** Standard gameplay<br>**Level:** Core game features.<br>**Context:**<br>(1) When the user arrives at a room, they can choose to add plants from their collection (archive) back to display on the placeholder spots. | | |

| | |
|---|---|
| | (2) The collection is accessible through the inventory, which is by default archived, meaning that the user cannot see the available plants in collection each time they arrive at or return to a virtual room.<br>(3) On the main screen, there are five available plant spots with a "+" icon, each representing a placeholder for an in-game plant. To add a plant from collection to any one of these spots, the player should first click on the "+" sign, which will trigger the plant inventory to show up on screen.<br>(4) On the inventory scrollbar, there is a “collection” button. Once clicked, the in-game collection will open, showing all archived plants in a horizontal scrollbar format.<br>(5) The player can select any one of the archived plants in collection by tapping on the plant. The plant will be added to their virtual room in the placeholder spot selected.<br>(6) The inventory option should also go back into the archive each time a plant is selected.<br>(7) The selected plant should no longer be in the archive.<br>**Frequency of occurrence:** Each time the user wishes to move a plant in archive back to virtual room display. |
| Design Use Case | **Scope:** Standard gameplay<br>**Level:** Core game features.<br>**Primary actors:** Player<br>**Description:** layer can view and choose from the collection and add selected plants to placeholder spots in a virtual room.<br>**Dependencies:**<br>(1) The user has successfully installed the game and is able to open up the game to the main game screen.<br>(2) The plant collection data and function has been developed and rendered.<br>**Preconditions:**<br>(1) There is at least one placeholder spot available in virtual room screen.<br>(2) There is at least one place currently archived in collection.<br>**Main Flow:** After the user arrive at the main screen that has a background of a virtual room (either by opening the app or returning to main screen from other game screens), the user should be able to see placeholder spots for in-game plants, to which the user can add plants from their collection of archived plants.<br>**Postconditions:** The player should be able to see the complete inventory of plants appear as a horizontal scrollbar with a button of the collection of the left top of the scrollbar on the main screen once they select a placeholder. Player can tap on collection button and pick a plant to add it to the spot. After selection the inventory scrollbar would disappear (go back into archive until player decides to add another plant). The selected plant should be removed from collection.<br>**Alternative Flows:**<br>**(1) Valid:**<br>Player arrives at virtual room screen, but no empty placeholders are available to hold any more plants.<br>Player arrives at virtual room screen but do not wish to add a new plant.<br>**(2) Invalid:**<br>Player arrives at the virtual room screen but does not see any "+" icons.<br>Player arrives at the virtual room screen screen, selects a placeholder, but collection button does not appear. |

| | |
|---|---|
| | Player arrives at the main screen, selects a placeholder, selects the collection button, the inventory list appears but is not correctly displayed - scroll function not working, layout inappropriate, not all plants are listed, etc.<br>Player selects a plant, but it is not added to the placeholder.<br>Player selects a plant, the plant is added to the placeholder, but collection does not go back into archive.<br>Player selects a plant, the plant is added to the placeholder, but the plant is not removed from collection.<br>**Frequency of occurrence:** Each time the user wishes to move a plant in archive back to virtual room display. |
| UI Design | **Key UI components:**<br>Main game screen<br>Plant placeholder icon<br>Plant collection icon<br>Plants (in collection)<br>Collection list as a scroll bar<br>**Wireframe:**<br>  |




**Interaction Flow:**

User clicks on a placeholder icon "+".

Plant inventory appears on screen as a horizontal scrollbar on the bottom half of the screen with a collection button on the top left.

Collection inventory appears, replacing the original inventory.

User can select any plant from collection in inventory.

Selected plant will be added to the placeholder.

Once a plant is selected, the inventory goes back into archive.

**UI Behaviour:**

The inventory list and plant collection should have a smooth animated effect when appearing on screen.

The inventory should by default be archived on the main screen, only to show when the player clicks on a placeholder icon.

The player should be able to view all inventory plants by scrolling left and right in the form of a scrollbar.

The player should be able to view all plants in collection by scrolling left and right in the form of a scrollbar.

**Visual Design Guidelines:**

Styling should align with the game's defined typography and colours.

The expanded inventory and collection should be displayed centre in the bottom half of the screen.

All plants should show as an image icon with its name labelled next to the icon.

The font size and style of plant name labels should ensure it is eligible but not consuming too much space.

The size of each plant icon should be enough for the player to tell what it is.

Each plant icon should be of similar size.

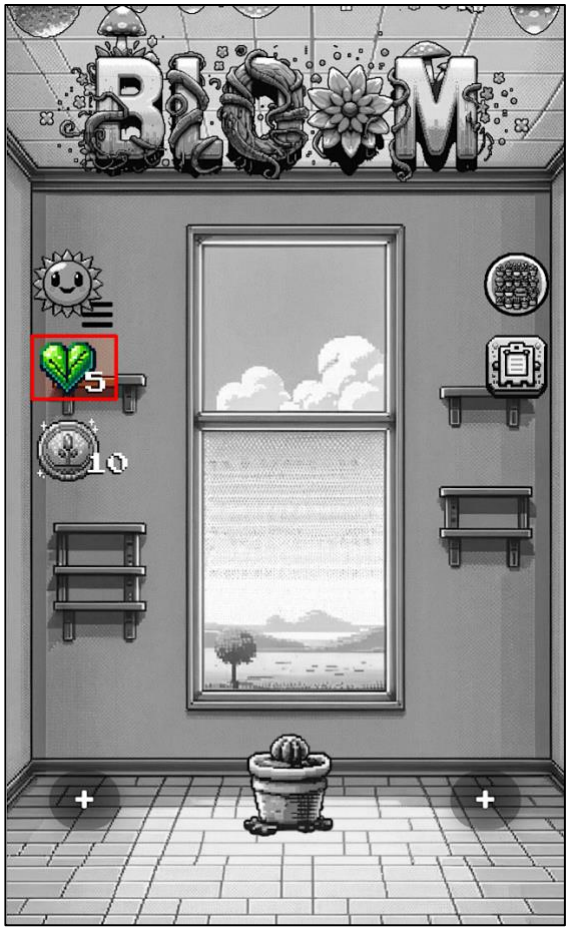
The size of each image icon should be sufficient for the user to effortlessly select.

| | |
|---|---|
| | Each option should have an icon with an appropriate image that matches the style of the game.<br>There should be enough spacing between each option for the player to choose and make selection. |

| | | | |
|---|---|---|---|
| US7-V1 | Displaying number of coins with the coin icon. | | |
| Description | As a player, I want to view the number of in-game coins I own on the main screen. | | |
| Acceptance Criteria | Criteria ID | Description | Test Status |
| | US7-V1-AC1. | An indicator in the main menu indicating the number of game coin user currently own. | Pass |
| | US7-V1-AC2. | When user earn more coins by completing a quiz, the amount earned is added to the number displayed on main screen. | Pass |
| Requirement Use Case | **Use Case:** Displaying the most up to date in-game coins amount on the main menu.<br>**Scope:** Standard gameplay<br>**Level:** Core game features.<br>**Context:**<br>(1) When the player arrives at the main screen, the number of coins shows.<br>(2) The number of the coin should always be up to date.<br>**Frequency of occurrence:** very time player in the game's main screen. | | |
| Design Use Case | **Scope:** Standard gameplay<br>**Level:** Core game features.<br>**Primary actors:** Player<br>**Description:** Displaying the most up to date in-game coins amount on the main menu.<br><br>**Dependencies:** The user has successfully installed the game and can open the game to the main game screen.<br>**Preconditions:** Save progress function working as expected.<br>**Main Flow:** Player arrives at main screen having just launched the game or returned from other screens.<br>**Postconditions:** The coins icon is displayed along the left edge of the main screen, labelled with the number of coins owned by the player.<br>**Alternative Flows:**<br>**Invalid:**<br>Player opens the game and arrives game main screen, coin indicator indicates wrong amount of game coins.<br>Player returns to main screen from another screens, coin indicator indicates wrong amount of game coins.<br>Player opens the game and arrives game main screen, no coin indicator appears.<br>Player returns to main screen from another screens, no coin indicator appears. | | |

| | |
|---|---|
| | Player opens the game and arrives game main screen, coin indicator appears in wrong position.<br>Player returns to main screen from another screens, coin indicator appears in wrong position.<br>**Frequency of occurrence:** Every time player arrives at the main screen. |
| UI Design | **Key UI components:**<br>Main game screen<br>Coins icon with labelled number<br>**Wireframe:**<br>  <br>**Interaction Flow:**<br>Player arrives at the main screen.<br>The "coin" icon and numeric label is displayed as a main screen component with the number of coins currently owned by the player.<br>If player earns more coins, or makes a purchase with coins in the shop, the corresponding additions / deductions will immediately show on the numeric label.<br>**UI Behaviour:**<br>The number of coins will automatically update each time player has earned or spent coins.<br>**Visual Design Guidelines:**<br>The "coin" icon and label should be at the side of the screen.<br>The button should not cover other functional items except the background.<br>The icon should resemble a coin.<br>The button should contain a number indicating the amount of coins player owning. |

<table>
<tr><td>US11-V1</td><td colspan="3">Displaying number of hearts with the heart icon.</td></tr>
<tr><td>Description</td><td colspan="3">As a player, I want to view the number of in-game hearts I own on the main screen.</td></tr>
<tr><td rowspan="5">Acceptance Criteria</td><td>Criteria ID</td><td>Description</td><td>Test Status</td></tr>
<tr><td>US11-V1-AC1.</td><td>An indicator in the main menu indicating the number of in-game hearts player currently own.</td><td>Pass</td></tr>
<tr><td>US11-V1-AC2.</td><td>When player purchases more hearts in the shop, the number is immediately updated on main screen.</td><td>Pending</td></tr>
<tr><td>US11-V1-AC3.</td><td>When player loses hearts from incorrect answers in a quiz, the number is immediately updated on main screen.</td><td>Pass</td></tr>
<tr><td>US11-V1-AC4.</td><td>The number of hearts replenished at regular intervals, until the user is back with 5 hearts.</td><td>Pending</td></tr>
<tr><td>Requirement Use Case</td><td colspan="3">Use Case: Displaying the most up to date in-game heart amount on the main menu.<br>Scope: Standard gameplay<br>Level: Core game features.<br>Context:<br>(1) When the player arrives at the main screen, the number of hearts is shown for the user’s information about their status in the game.<br>(2) The number of the hearts should always be up to date.<br>Frequency of occurrence: Every time player in the game's main screen.</td></tr>
<tr><td>Design Use Case</td><td colspan="3">Scope: Standard gameplay<br>Level: Core game features.<br>Primary actors: Player<br>Description: Displaying the most up to date in-game hearts amount on the main menu.<br><br>Dependencies:<br>(1) The user has successfully installed the game and can open the game to the main game screen.<br>(2) Heart revival functionality complete.<br>(3) Shop function complete.<br>Preconditions: The relevant save progress function working as expected.<br>Main Flow: Player arrives at main screen having just launched the game or returned from other screens.<br>Postconditions: The hearts icon is displayed along the left edge of the main screen, labelled with the number of coins owned by the player.<br>Alternative Flows:<br>Invalid:<br>Player opens the game and arrives game main screen, heart indicator indicates wrong number of in-game hearts.<br>Player returns to main screen from another screens, heart indicator indicates wrong number of in-game hearts.</td></tr>
</table>

| | |
|---|---|
| | Player opens the game and arrives game main screen, no heart indicator appears.<br>Player returns to main screen from another screens, no heart indicator appears.<br>Player opens the game and arrives game main screen, heart indicator appears in wrong position.<br>Player returns to main screen from another screens, heart indicator appears in wrong position.<br>**Frequency of occurrence:** Every time player arrives at the main screen. |
| UI Design | **Key UI components:**<br>Main game screen<br>Hearts icon with labelled number<br>**Wireframe:**<br>  <br>**Interaction Flow:**<br>Player arrives at the main screen.<br>The "heart" icon and numeric label are displayed as a main screen component with the number of hearts currently owned by the player.<br>If player purchases more hearts, or loses hearts from incorrect answers in the quiz, the corresponding additions / deductions will immediately show on the numeric label.<br>**UI Behaviour:**<br>If player presses on the icon, an animated effect is shown.<br>The number of hearts will automatically update each time player purchased or lost hearts. |

| | |
|---|---|
| | If less than 5 hearts, the number of hearts will increase by one at a time interval until it reaches 5.<br>**Visual Design Guidelines:**<br>The "hearts" icon should be at the side of the screen.<br>The icon should not cover other functional items except the background.<br>The icon should be in the shape of a heart, coloured in green to match the game theme.<br>The numeric label should contain a number indicating the number of hearts owned by player currently owns. |

| US12-V1 | I want to enter and view the collection (archive) by pressing a button in main screen. | | |
|---|---|---|---|
| Description | As a player, I want to view the number of in-game hearts I own on the main screen. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US12-V1-AC1.** | A button is visible in main screen with a collection icon. | **Pass** |
| | **US12-V1-AC2.** | When player clicks on the icon, they will leave the main screen and enter the plant collection. | **Pass** |
| Requirement Use Case | **Use Case:** Accessing to the collection screen with a button on the main screen.<br>**Scope:** Archive and collection.<br>**Level:** Core game features.<br>**Context:**<br>(1) A button with a "collection" icon is displayed on the main menu.<br>(2) The use can enter the collection via the icon if they wish to view or perform actions on the archived plants.<br>**Frequency of occurrence:** Every time player is on the main screen and wish to access the collection screen. | | |
| Design Use Case | **Scope:** Archive and collection.<br>**Level:** Core game features.<br>**Primary actors:** Player<br>**Description:** User can enter and view plants in collection via the collection button on main screen.<br>**Dependencies:** The user has successfully installed the game and can open the game to the main game screen.<br>**Assumptions:** The use has at least one plant currently in collection.<br>**Main Flow:** Player arrives at main screen having just launched the game or returned from other screens. Player clicks the collection button and enters the collection screen.<br>**Postconditions:** Player successfully enters the collection screen where archived plants are displayed.<br>**Alternative Flows:**<br>**Invalid:**<br>Player opens the game or switches to the main screen, "collection" button does not appear<br>Player presses the "collection" button, the game fails to switch to the collection screen | | |

| | |
|---|---|
| | Player presses the "collection" button, switches to the collection screen, archived plants are not displayed on the screen.<br>Player presses the "collection" button, switches to the collection screen, incorrect type or number of plants are show on the screen.<br>**Frequency of occurrence:** Every time player wishes to enter plant collection. |
| UI Design | **Key UI components:**<br>Main game screen<br>"Collection" icon<br>Plants (in collection)<br>**Wireframe:**<br>  <br><br>**Interaction Flow:**<br>Player arrives the main screen.<br>Player press on the "collection" icon.<br>Player leaves the main screen and enters the collection screen.<br>**UI Behaviour:**<br>The "collection" button should have a hollow animation when player pressed it.<br>**Visual Design Guidelines:**<br>The "collection" button should be in the corner of the screen.<br>The button should not cover other functional items except the background. |

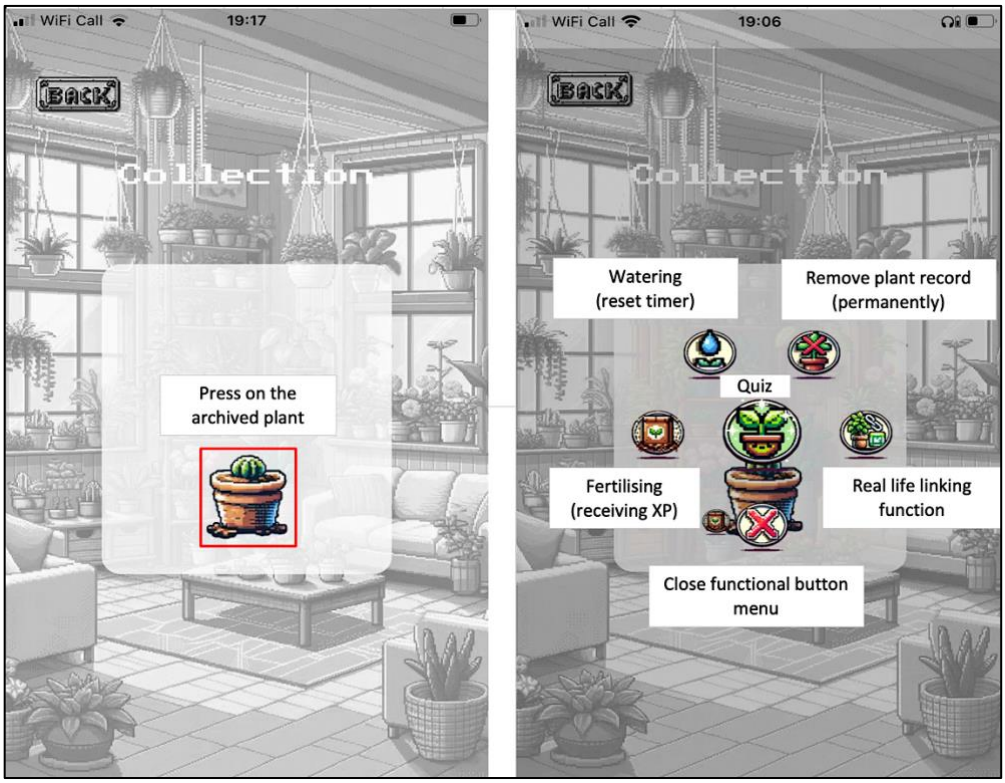
| US13-V1 | I want to view all archived plants after entering the plant collection. | | |
|---|---|---|---|
| Description | As a player, I want to view all my archived plants in the collection screen. | | |
| | Criteria ID | Description | Test Status |

<table>
<tr><td rowspan="2">Acceptance Criteria</td><td>US13-V1-AC1.</td><td>Once in the collection, plants are displayed in the form of cards in centre of screen.</td><td>Pass</td></tr>
<tr><td>US13-V1-AC2.</td><td>Player can scroll left and right to see all plants in collection.</td><td>Pass</td></tr>
<tr><td>Requirement Use Case</td><td colspan="3">Use Case: Viewing all archived plants in collection screen.<br>Scope: Archive and collection.<br>Level: Core game features.<br>Context:<br>(1) Once in the collection, user wants to see all archived plants listed in the collection screen with unique information (e.g. plant type, stage...etc).<br>(2) User can scroll horizontally to see all plants in collection<br>Frequency of occurrence: Every time player is in the collection screen.</td></tr>
<tr><td>Design Use Case</td><td colspan="3">Scope: Archive and collection.<br>Level: Core game features.<br>Primary actors: Player<br>Description: User can enter and view plants in collection via the collection button on main screen.<br>Dependencies:<br>(1) The user has successfully installed the game and can open the game to the main game screen.<br>(2) The archive function has been built and successfully archived plants to collection.<br>Assumptions:<br>(1) Player's game progress has been saved<br>(2) Player has entered collection via collection button on main screen.<br>Main Flow: An archived plant appears at the middle of the collection screen. User scrolls left or right to view other plants if there are two or more plants archived.<br>Postconditions: Player successfully enters the collection screen where archived plants are displayed.<br>Alternative Flows:<br>Valid: No plants are displayed in collection because no plants were previously archived by the user.<br>Invalid:<br>No plants or wrong plants shown on the screen.<br>Plants are improperly displayed (incorrect alignment, missing information, etc.)<br>Frequency of occurrence: Every time player enters plant collection.</td></tr>
<tr><td>UI Design</td><td colspan="3">Key UI components:<br>"Collection" screen<br>"Collection" icon<br>Plants (in collection)<br>Scroll bar.<br>Wireframe:</td></tr>
</table>

| | |
|---|---|
| |  <br>**Interaction Flow:**<br>Player arrives the main screen.<br>Player press on the "collection".<br>Game screen returns the collection screen with one archived plant.<br>Player scrolls left or right (if more than one plant archived)<br>Game displays other archived following the scroll action.<br>**UI Behaviour:**<br>The plants in collection should be scrollable horizontally.<br>**Visual Design Guidelines:**<br>The plants should appear in the form of cards that are scrollable from left to right.<br>Each plant should be clear and visible, displayed at the centre of screen. |

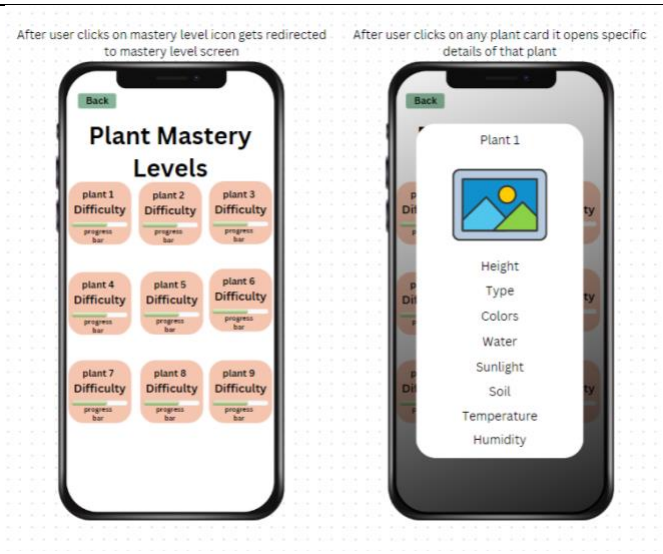
| US14-V1 | Perform plant actions to plants in collection. | | |
|---|---|---|---|
| Description | As a player, I want to view the number of in-game hearts I own on the main screen. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US14-V1-AC1.** | A button is visible in main screen with a collection icon. | **Pass** |
| | **US14-V1-AC2.** | When player clicks on the icon, they will leave the main screen and enter the plant collection. | **Pass** |
| Requirement Use Case | **Use Case:** Interacting with plants whilst in collection.<br>**Scope:** Game set-up.<br>**Level:** Core game features.<br>**Context:**<br>(1) Due to limited placeholder space in virtual room, the use is allowed to interact with plants that are currently in the collection. | | |

| | |
|---|---|
| | (2) The use can expand and implement functions the plant action menu by tapping on a plant in the collection.<br>**Frequency of occurrence:** Every time player is in the collection and wish to interact with an archived plant. |
| Design Use Case | **Scope:** Archive and collection.<br>**Level:** Core game features.<br>**Primary actors:** Player<br>**Description:** Player interacts with archived plants through functional buttons.<br>**Dependencies:** The user has successfully installed the game and run the game on device.<br>**Assumptions:**<br>(1) User has entered the collection and is able to view plants.<br>(2) The use has at least one plant currently in collection.<br>**Main Flow:** Player scrolls and presses on the target plant for interaction, then press on the functional button for the specific interactional function.<br>**Postconditions:** Corresponding functions performed according to the choice of the button player selected in action menu.<br>**Alternative Flows:**<br>**Valid:** No plants are displayed in collection because no plants were previously archived by the user.<br>**Invalid:**<br>Player opens the game and arrives collection screen(at least one plant being archived), no/missing/false archived plants show on screen.<br>Player presses on the plant in the collection screen, no/missing/false interactive functional buttons pop out.<br>Player presses the "Fertilizing" button, player does not/does not gain the right amount of game XP.<br>Player presses the "Watering" button, the timer for interacting plant does not reset/reset with false amount of time.<br>Player presses the "Quiz" button, game screen fails to return quiz screen.<br>Player presses the "Link to real life" button, game screen fails to return the link to real life screen.<br>Player presses the "Delete plant" button, the plant record does not being removed.<br>**Frequency of occurrence:** Every time player want to have interactions with plants in collection. |
| UI Design | **Key UI components:**<br>Collection screen.<br>Plants in collection.<br>"Watering" icon.<br>"Remove plant record" icon.<br>"Real life linking" icon.<br>"Close menu" icon.<br>"Fertilising" icon.<br>“Quiz” icon. |

| | |
|---|---|
| | Plants (in collection screen)<br>**Wireframe:**<br>  <br>**Interaction Flow:**<br>Player arrives collection screen.<br>Player scrolls left or right (if more than one plant archived)<br>Game screens change other archived following the scroll action.<br>Player press on the plant on collection screen<br>Functional buttons pop surrounding the plant.<br>Player press on the functional button.<br>Corresponding functional reactions according to the button selected<br>Returns functional screen if specific function was selected.<br>**UI Behaviour:**<br>The plant should be in the middle of the collection screen.<br>The functional buttons should appear in similar fashion to plant action menu displayed in virtual room.<br>**Visual Design Guidelines:**<br>The functional buttons should appear in a pentagon-shape and surround the plant. |

| US16-V1 | See plant information for a plant by tapping a plant card in mastery level screen. | | |
|---|---|---|---|
| Description | As a player, I want to see the plant information when i click on any plant card in the matery level screen. | | |
| | **Criteria ID** | **Description** | **Test Status** |

| Acceptance Criteria | US16-V1-AC1. | I should be able to see the plants information like height type, etc., when I click a plant and move back to level mastery screen wherever I tap on the screen | Pass |
|---|---|---|---|
| Requirement Use Case | **Use Case:** Display plant information whenever a card for that plant in the mastery level screen is clicked.<br>**Scope:** Game set-up.<br>**Level:** Core game features.<br>**Context:**<br>(1) After the player is already in the mastery level screen, they may wish to learn general information about each plant type.<br>(2) They can do so by tapping the card for that plant, the general information will then show up on screen.<br>**Frequency of occurrence:** Every time player is in mastery level screen and wishes to see general plant information for a type of plant. | | |
| Design Use Case | **Scope:** Game set-up.<br>**Level:** Core game features.<br>**Primary actors:** Player<br>**Description:** Player can tap on any plant card and see the specific information of that card and close the card to return to plant mastery screen.<br>**Dependencies:**<br>(1) The user has successfully installed the game and can open the game to the main game screen.<br>(2) Plants and Plant mastery levels are rendered in the game.<br>**Preconditions:** Player has entered the plant mastery screen via the plant mastery button on main screen.<br>**Main Flow:** Player taps on any plant card in mastery screen. General information about the plant is rendered on screen.<br>**Postconditions:** Player once satisfied with the plant info can click anywhere on the screen to navigate back to plant mastery screen.<br>**Alternative Flows:**<br>**Valid:** Player navigates back to main screen from mastery screen without viewing any plant information.<br>**Invalid:**<br>General information is not displayed or wrongly displayed after player taps on a plant mastery card.<br>The general information does not disappear after user clicks anywhere on the screen.<br>**Frequency of occurrence:** Every time player is in plant mastery screen and wishes to see general plant information for a type of plant. | | |
| UI Design | **Key UI components:**<br>Plant mastery screen<br>Plant mastery cards<br>General plant information pop-up<br>**Wireframe:** | | |




**Interaction Flow:**

Player taps on a plant mastery card.

General information about the plant appears on screen as a pop-up card.

Player clicks anywhere on screen to collapse the general information and return to plant mastery screen where all plant mastery cards are displayed.

**UI Behaviour:**

(1) Intuitive Navigation: The user should be able to navigate through the plant mastery cards effortlessly. Tapping on a specific card should provide additional details, and a clear back button should facilitate a seamless return to the main screen.

(2) Specific plant information details is displayed in pop-up format when a card is clicked overlapping with the plant mastery screen.

(3) The plant mastery screen is dimmed so that focus is on plant details diaplayed.

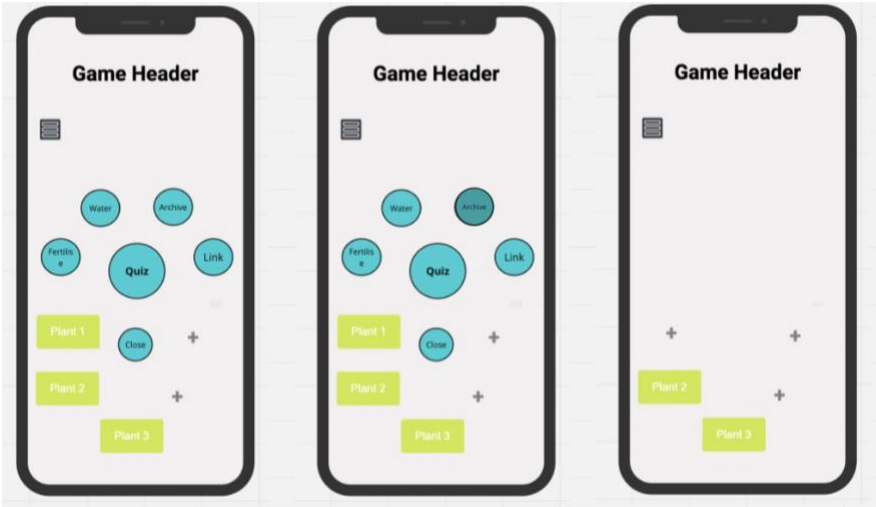
**Visual Design Guidelines:**

Consistent Styling: Maintain a consistent design language with the rest of the game, using the established typography, colour palette, and overall theme. This consistency contributes to a cohesive and polished visual identity.

Plant information displayed should be clearly eligible.

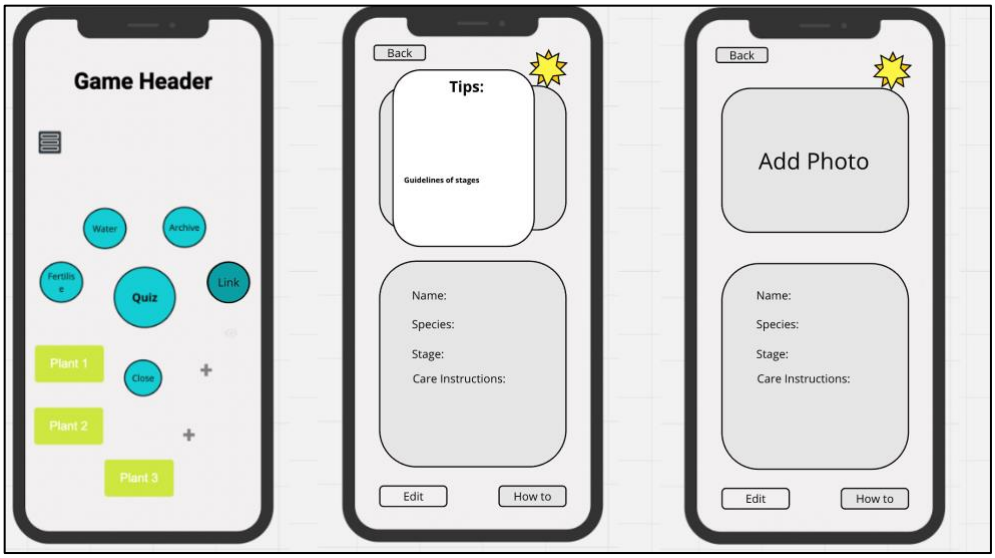
| US18-V1 | Archive plant with “archive” option in plant action menu. | | |
|---|---|---|---|
| Description | As a player, want to move the selected plant to the collection by pressing the archive button in plant action menu. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US18-V1-AC1.** | After pressing the archive button, the plant is going to be removed from the screen. | **Pass** |
| | **US18-V1-AC2.** | The plant menu will collapse. | **Pass** |
| | **US18-V1-AC3.** | The plant is added to the plant collection, and can be viewed together with other archived plants if player enters collection screen. | **Pass** |

| | |
|---|---|
| Requirement Use Case | **Use Case:** Archiving a plant by pressing the archive button in plant action menu for plants currently planted in a virtual room.<br>**Scope:** Archive and collection.<br>**Level:** Core game features.<br>**Context:**<br>(1) As each virtual room offers limited placeholder space for plants, the archive option is offered to users who wish to grow more plants. After player expands the plant action menu for a plant in virtual room, the player can press the archive button, to move their plant to the collection and make space for other plants they wish to display.<br>(2) The archive button is shown as a part of the menu by default.<br>(3) The plant will then stay in the collection until player decides to move it back to a room spot of delete the plant permanently from game.<br>**Frequency of occurrence:** Every time the user wishes to archive a plant currently taking spce in the virtual room. |
| Design Use Case | **Scope:** Archive and collection.<br>**Level:** Core game features.<br>**Primary actors:** Player<br>**Description:** Player can move the selected plant (currently in virtual room display) to the collection screen by clicking on the archive button in the expanded plant action menu.<br>**Dependencies:**<br>(1) The user has successfully installed the game, able to open the game to a virtual room screen and could access the plant menu.<br>(2) The collection function has been built.<br>**Assumptions:** Player currently as at least one plant in their virtual room(s).<br>**Preconditions:** Player has expanded the plant action menu for a plant currently displayed in virtual room.<br>**Main Flow:** After the user arrive at a virtual room screen (by default the main screen) that contains at least one plant, the user should be able to click on a plant to expand the plant action menu and select the archive button, to move the selected plant to archived collection.<br>**Postconditions:** The selected plant should now be removed from the room screen, e replaced by the original "+" button in its original position. The selected plant will be moved to the collection screen.<br>**Alternative Flows:**<br>**Valid:** Player selects other options in the plant action menu.<br>**Invalid:**<br>Player clicks on the archive button, but the plant is not removed.<br>Player clicks on the archive button, but the plant menu does not collapse<br>Player clicks on the archive button and the plant is removed but the plus button does not appear.<br>**Frequency of occurrence:** Every time player is in plant mastery screen and wishes to see general plant information for a type of plant. |
| UI Design | **Key UI components:**<br>Main game screen.<br>Plants (already added to the player's main screen). |

| | |
|---|---|
| | Expanded plant option menu.<br>Archive button.<br>**Wireframe:**<br>  <br><br>**Interaction Flow:**<br>User clicks on the archive button of the plant action menu.<br>The plant menu will collapse.<br>The plant will be removed from the main menu.<br>The place where the plant is originally at will be replaced by the "+" button indicating the placeholder spot is now empty.<br>The archived plant is moved to plant collection.<br>**UI Behaviour:**<br>(1) The archive button should have a smooth animated effect when pressed.<br>(2) The archived plant should immediately disappear after player taps archive.<br>**Visual Design Guidelines:**<br>Consistent Styling: Maintain a consistent design language with the rest of the game, using the established typography, colour palette, and overall theme. This consistency contributes to a cohesive and polished visual identity. |

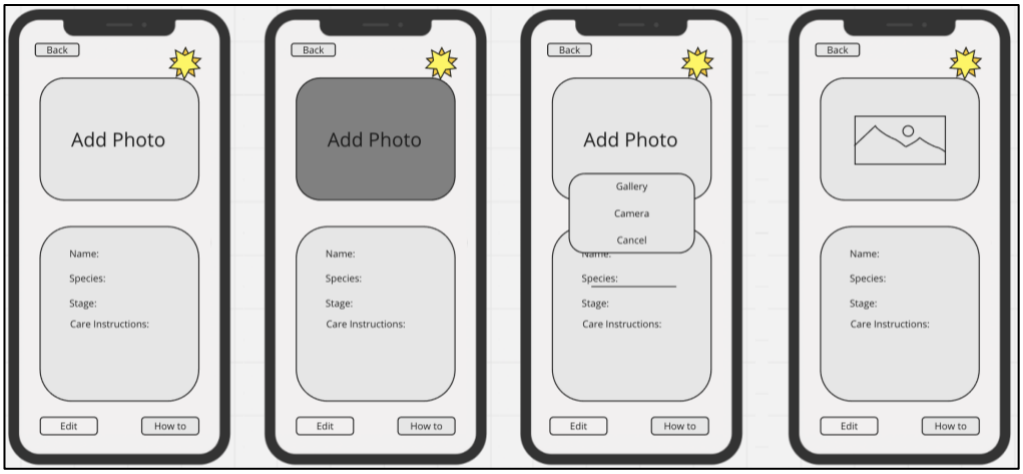
| | | | |
|---|---|---|---|
| US19-V1 | **Enter link screen to link an in-game plant to a real plant with the link button in plant action menu.** | | |
| Description | As a player, want to link an in-game plant to a real plant with the link button in plant action menu (either from main screen or collection screen). | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US19-V1-AC1.** | After pressing the link button, player arrives at the link screen. | **Pass** |
| | **US19-V1-AC2.** | The plant action menu collapses after pressing the link button. | **Pass** |
| | **US19-V1-AC3** | The link screen should have the following elements: header back button, a sun-shaped oracle, add photo button, edit button, how to | **Pass** |

| | | | |
|---|---|---|---|
| | | button, fields to display the information of the linked plant (name, stage, species, care instructions) and a water timer. | |
| Requirement Use Case | **Use Case:** Enter link screen to link an in-game plant to a real plant with the link button in plant action menu.<br>**Scope:** Link to real life.<br>**Level:** Core game features.<br>**Context:**<br>(1) The game's functionalities help users track the care for their real-life plants. The use may wish to link an in-game plant to a real-life plant to enable tracking features.<br>(2) When the user presses the link button, the user could link their plant.<br>(3) The link button is shown as a part of the menu by default.<br>(3) The player can link the selected plant by entering the link screen. To perform this action, the player needs to first select the targeted plant, click on it, and click on the link button.<br>(4) The link screen should have the following elements for its functionality: back button, a sun-shaped oracle, add photo button, edit button, how to button, fields to display the information of the linked plant (name, stage, species, care instructions) and a water timer.<br>**Frequency of occurrence:** Every time the user opened the plant menu and clicks the link button to enter the link screen. | | |
| Design Use Case | **Scope:** Link to real life.<br>**Level:** Core game features.<br>**Primary actors:** Player<br>**Description:** Enter link screen to link an in-game plant to a real plant with the link button in plant action menu.<br>**Dependencies:**<br>(1) The user has successfully installed the game, able to open the game to a virtual room screen or collection screen.<br>(2) Player can access the plant menu by tapping on a plant in virtual room or collection.<br>(3) Game assets (images and icons) are added appropriately to code file.<br>**Assumptions:** Player currently as at least one plant in their virtual room(s) and/or collection.<br>**Preconditions:** Player has expanded the plant action menu for a plant currently displayed in virtual room or collection.<br>**Main Flow:** After the user arrive at a virtual room screen (by default the main screen) that contains at least one plant, the user should be able to click on a plant to expand the plant action menu and select the link button to launch the link screen. The link screen should contain all components as specified in acceptance criteria.<br>**Postconditions:** The plant menu will collapse, and the linking screen will be shown with all the listed components.<br>**Alternative Flows:**<br>**Valid:** Player selects other options in the plant action menu.<br>**Invalid:**<br>Player clicks on the link button, but the plant menu didn't collapse.<br>Player clicks on the link button and the linking screen does not show up.<br>Elements in the link screen are not displayed as expected. | | |

| | |
|---|---|
| | **Frequency of occurrence:** Every time player wishes to enter the link screen to link an in-game plant to a real-life plant. |
| UI Design | **Key UI components:**<br>Plant option menu expanded for plant (either in room display or in collection).<br>Link button.<br>Link screen.<br>Back button.<br>Link screen components: header back button, a sun-shaped oracle, add photo button, edit button, how to button, fields to display the information of the linked plant (name, stage, species, care instructions) and a water timer.<br>**Wireframe:**<br>  <br>**Interaction Flow:**<br>User clicks on the link button of the plant menu.<br>The plant menu will collapse.<br>The link screen is rendered with all specified elements.<br>**UI Behaviour:**<br>The link button should have a smooth animated effect when pressed.<br>The link button should by default be shown on the plant menu.<br>The link screen should be shown with a smooth animation.<br>**Visual Design Guidelines:**<br>Consistent Styling: Maintain a consistent design language with the rest of the game, using the established typography, colour palette, and overall theme. This consistency contributes to a cohesive and polished visual identity.<br>All elements on the link screen should be clearly eligible. |

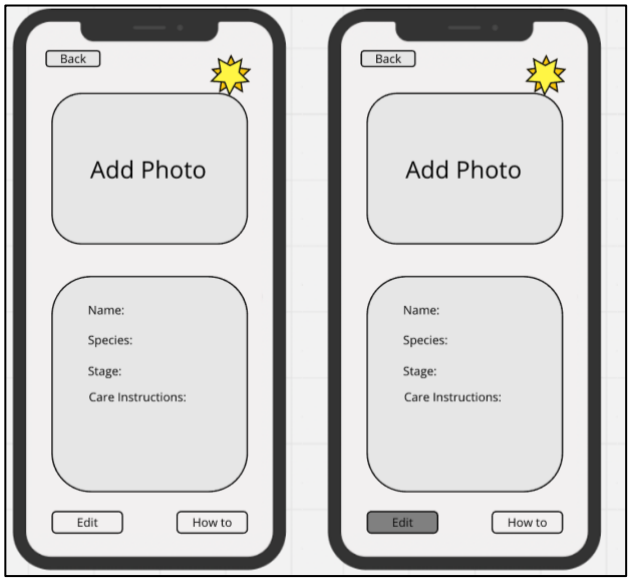
| US20-V1 | Add photo for linked plant. |
|---|---|
| Description | As a player, I want to add a photo to the plant that I want to link by pressing the add photo button on link screen. |

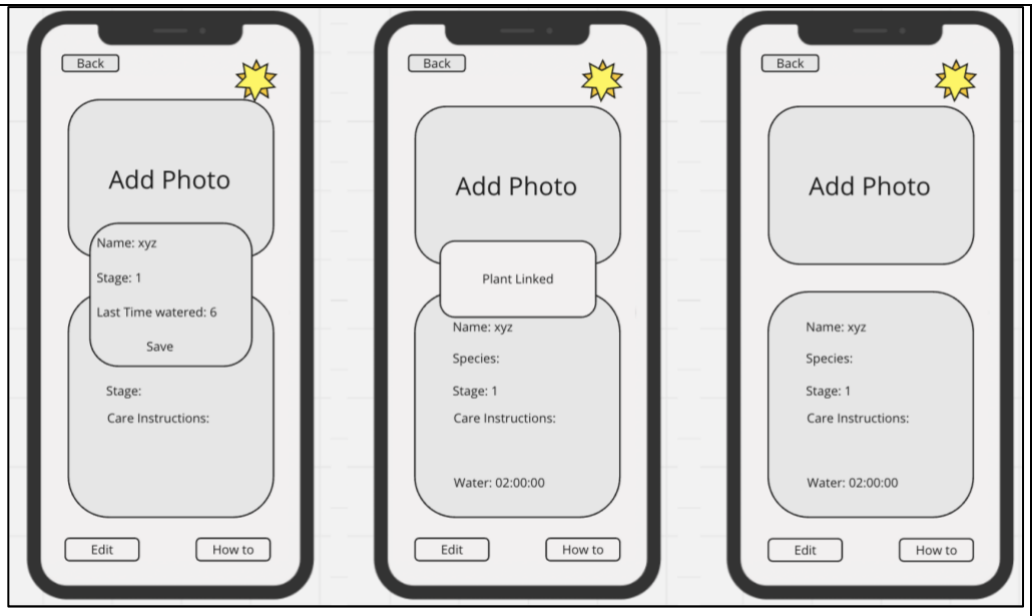
<table>
<tr><td rowspan="5">Acceptance Criteria</td><td>Criteria ID</td><td>Description</td><td>Test Status</td></tr>
<tr><td>US20-V1-AC1.</td><td>After pressing the add “photo button” on link screen, a pop-up menu will be shown.</td><td>Pass</td></tr>
<tr><td>US20-V1-AC2.</td><td>The pop-up menu contains 3 buttons, gallery, camera and cancel.</td><td>Pass</td></tr>
<tr><td>US20-V1-AC3</td><td>The user could add the photo using either method.</td><td>Pass</td></tr>
<tr><td>US20-V1-AC4</td><td>The photo would replace the “add photo” button if it is successfully added.</td><td>Pass</td></tr>
<tr><td>Requirement Use Case</td><td colspan="3">Use Case: Adding a photo to a plant by pressing the add photo button in link screen.<br>Scope: Link to real life.<br>Level: Core game features.<br>Context:<br>(1) The game’s functionalities help users track the care for their real-life plants. The use may wish to link an in-game plant to a real-life plant to enable tracking features.<br>(2) The user may wish to set a customised photo of their real-life plant.<br>(3) User is already in link screen. When the user press the add photo button in the link screen, the user could add a photo to their selected plant.<br>(4) The add photo button is shown as a part of the link screen by default.<br>(5) The player can add a photo to the selected plant. To perform this action, the player needs to first click on the add button and choose the desired method to choose a photo/ take a photo.<br>Frequency of occurrence: Every time the user opened the link screen and wishes to add a photo to the plant.</td></tr>
<tr><td>Design Use Case</td><td colspan="3">Scope: Link to real life.<br>Level: Core game features.<br>Primary actors: Player<br>Description: Player can add a photo to the selected plant by clicking on the add photo button in link screen.<br>Dependencies:<br>(1) The user has successfully installed the game, able to open the game to a virtual room screen or collection screen.<br>(1) The user has successfully installed the game, and able to open up the game to the main game screen.<br>(2) The user could add a plant and could access both the plant menu and the link button.<br>Assumptions: Player currently as at least one plant in their virtual room(s) and/or collection to link with real life plant, as well as a real-life plant they wish to manage in the game.<br>Preconditions: Player has entered the link screen.<br>Main Flow: After the user arrive at the link screen (by clicking the link button of a selected plant’s action menu), the user should be able to click on the add photo button, choose a method, and add a photo to the selected plant.<br>Postconditions: The selected photo would replace the add photo button.<br>Alternative Flows:</td></tr>
</table>

| | |
|---|---|
| | **Valid:** Player clicks on add photo button and cancel the action by clicking outside of the pop up menu/ clicking cancel.<br>**Invalid:**<br>Player clicks on the add photo button but the pop up menu does not show up.<br>Player clicks on the add photo button, chosen a method but it is not doing the desired action (e.g. click gallery but opened camera)<br>Player has chosen a photo, but the photo does not replace the button<br>**Frequency of occurrence:** Every time the user opened the link screen and wishes to add a photo to the plant. |
| UI Design | **Key UI components:**<br>Link screen.<br>"Add photo" button.<br>Pop up menu (prompt user desired method).<br>**Wireframe:**<br>  <br>**Interaction Flow:**<br>User clicks on the "Add photo" button.<br>A pop-up menu will be shown.<br>Users choose a picture from their phone gallery/ take a photo.<br>The photo replaces the add photo button.<br>**UI Behaviour:**<br>The add photo button should have a smooth animated effect when pressed.<br>The pop-up should be shown with a smooth animation.<br>The photo used to replace the button should be cropped to fit the button.<br>**Visual Design Guidelines:**<br>Consistent Styling: Maintain a consistent design language with the rest of the game, using the established typography, colour palette, and overall theme. This consistency contributes to a cohesive and polished visual identity. |

| | |
|---|---|
| US21-V1 | Edit details of linked plant. |
| Description | As a player, I want to edit the detail of the plant and link the plant via the edit button |

<table>
<tr><td rowspan="5">Acceptance Criteria</td><td>Criteria ID</td><td>Description</td><td>Test Status</td></tr>
<tr><td>US21-V1-AC1.</td><td>After pressing the edit button, a pop-up menu will be shown.</td><td>Pass</td></tr>
<tr><td>US21-V1-AC2.</td><td>The pop-up menu contains 3 input fields, name, stage and last time watered.</td><td>Pass</td></tr>
<tr><td>US21-V1-AC3</td><td>The user fills in the 3 fields using the keyboard on their phone.</td><td>Pass</td></tr>
<tr><td>US21-V1-AC4</td><td>The details entered would be reflected in the fields respectively.</td><td>Pass</td></tr>
<tr><td>Requirement Use Case</td><td colspan="3">Use Case: Adding a photo to a plant by pressing the add photo button in link screen.<br>Scope: Link to real life.<br>Level: Core game features.<br>Context:<br>(1) When the user presses the edit button in link screen, the user could edit the details of their selected plant.<br>(2) The edit button is shown as a part of the link screen by default.<br>(3) The player can add edit the details of the selected plant. To perform this action, the player needs to first click on the edit button and fill in the fields shown.<br>Frequency of occurrence: Every time the user opened the link screen and wishes to edit the details of the plant.</td></tr>
<tr><td>Design Use Case</td><td colspan="3">Scope: Link to real life.<br>Level: Core game features.<br>Primary actors: Player<br>Description: Player can edit the details of the selected plant by clicking on the edit button in link screen.<br>Dependencies:<br>(1) The user has successfully installed the game, and able to open the game to the main game screen.<br>(2) The user could add a plant and could access both the plant menu and the link button.<br>Assumptions:<br>(1) Player has at least one plant in their virtual room or collection.<br>(2) Player’s device has a functional keyboard.<br>Preconditions: Player has entered the link screen.<br>Main Flow: After the user arrive at the link screen (by clicking the link button of a selected plant), the user should be able to click on the edit button, fill in the fields in the popup menu and save the details entered.<br>Postconditions:<br>(1) The details entered by the users would be reflected in the fields respectively.<br>(2) The countdown timer will start automatically.<br>(3) There will be a pop-up message saying the plant is successfully linked.<br>Alternative Flows:</td></tr>
</table>

| | |
|---|---|
| | **Valid:** Player clicks on edit button and cancel the action by clicking outside of the pop-up menu<br>**Invalid:**<br>Player clicks on the edit button but the pop up menu does not show up.<br>Player clicks on the edit button, but could not fill in the fields.<br>Player fills in all the fields but the details are not reflected on screen.<br>The timer did not start.<br>Time is not correctly reflected (i.e. total time needed minus last watered)<br>**Frequency of occurrence:** Every time the user opened the link screen and wishes to add a photo to the plant. |
| UI Design | **Key UI components:**<br>Link screen.<br>Edit button.<br>Pop up menu (Fields needed to input).<br>Pop up message (Successfully linked).<br>**Wireframe:**<br>  |




**Interaction Flow:**

User clicks on the edit button in link screen.

A pop-up menu will be shown.

Users enter the fields inside the pop-up menu.

User click the save button inside the pop-up menu.

A pop-up message will be shown.

User clicks anywhere on the screen to close the message.

The water countdown timer will start.

**Data Entry and validation:**

(1) The details for name should be in English

(2) The details entered for time should be an integer

(3) The details entered for stage should be in integer

**UI Behaviour:**

The edit button should have a smooth animated effect when pressed.

The pop-up menu should be shown with a smooth animation.

The pop-up message should be shown with a smooth animation.

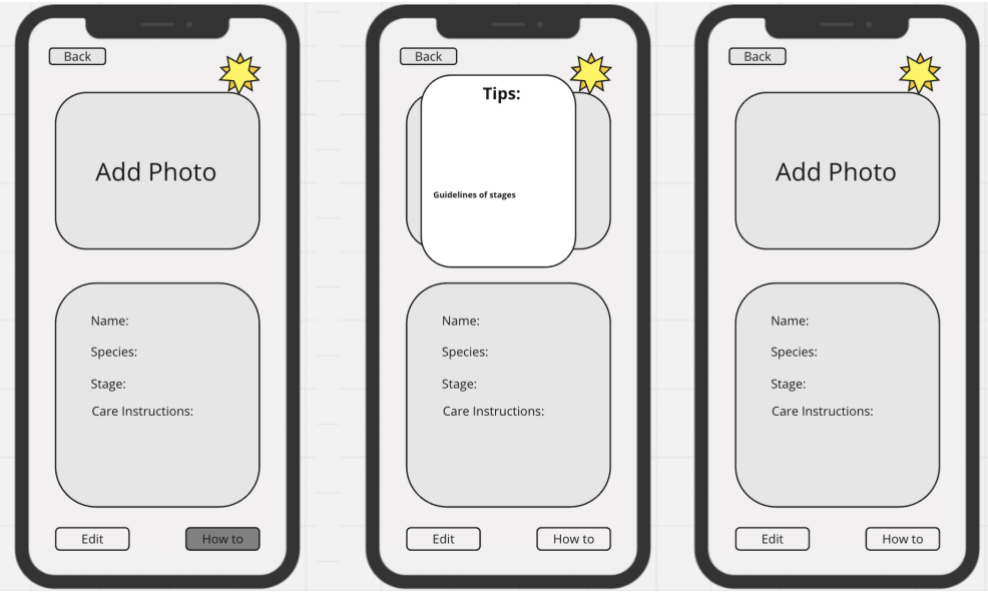
**Visual Design Guidelines:**

Consistent Styling: Maintain a consistent design language with the rest of the game, using the established typography, colour palette, and overall theme. This consistency contributes to a cohesive and polished visual identity.

| US50-V1 | Receive instructions on editing details of linked plants. | | |
|---|---|---|---|
| Description | As a player, I want to see the instructions to fill in the details for a linked plant by pressing the how to button. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US50-V1-AC1.** | After pressing the “how to” button, a pop up menu of advice will be shown next to the oracle. | **Pass** |

| | | | |
|---|---|---|---|
| | US50-V1-AC2. | The advice contains the instructions of how to fill in the page and how to determine the stage of a plant. | Pending |
| | US50-V1-AC3 | The user could close the menu by tapping outside of the popup menu. | Pass |
| Requirement Use Case | **Use Case:** Receive instructions on editing details of linked plants.<br>**Scope:** Link to real life.<br>**Level:** Core game features.<br>**Context:**<br>(1) When the user presses the how-to button, the user could see the tips from the oracle for their selected plant.<br>(2) The how-to button is shown as a part of the link screen by default.<br>(3) The player can close the menu by pressing spaces outside of the menu.<br>**Frequency of occurrence:** Every time the user opened the link screen and wishes to see the tips of how to edit details for linked plant. | | |
| Design Use Case | **Scope:** Link to real life.<br>**Level:** Core game features.<br>**Primary actors:** Player<br>**Description:** Player can see the instructions to fill in the details for a linked plant by pressing the how to button.<br>**Dependencies:**<br>(1) The user has successfully installed the game, and able to open the game to the main game screen.<br>(2) The user could add a plant and could access both the plant menu and the link button.<br>**Assumptions:**<br>(1) Player has at least one plant in their virtual room or collection.<br>**Preconditions:** Player has entered the link screen.<br>**Main Flow:** After the user arrive at the link screen (by clicking the link button of a selected plant), the user should be able to click on the how to button and see the advice for the plant.<br>**Postconditions:**<br>(1) The correct advice and tips are shown to the user in a clear and eligible manner.<br>(2) User can hide the tips by tapping anywhere outside of the pop-up.<br>**Alternative Flows:**<br>**Invalid:** Player clicks on the how to button but the pop up menu does not show up, or shows up but displays incorrect information.<br>**Frequency of occurrence:** Every time the user opened the link screen and wishes to see the tips of how to edit details for linked plant. | | |
| UI Design | **Key UI components:**<br>Link screen.<br>“How to” button.<br>Pop up information (advice and tips). | | |

**Wireframe:**




**Interaction Flow:**

User clicks on the how to button.

A pop-up menu will be shown.

User closes the pop-up menu by clicking anywhere outside of the menu.

**Data Entry and validation:**

(1) The details for name should be in English

(2) The details entered for time should be an integer

(3) The details entered for stage should be in integer

**UI Behaviour:**

The “how to” button should have a smooth animated effect when pressed.

The pop-up menu should be shown with a smooth animation.

The pop-up menu should be archived with a smooth animation.

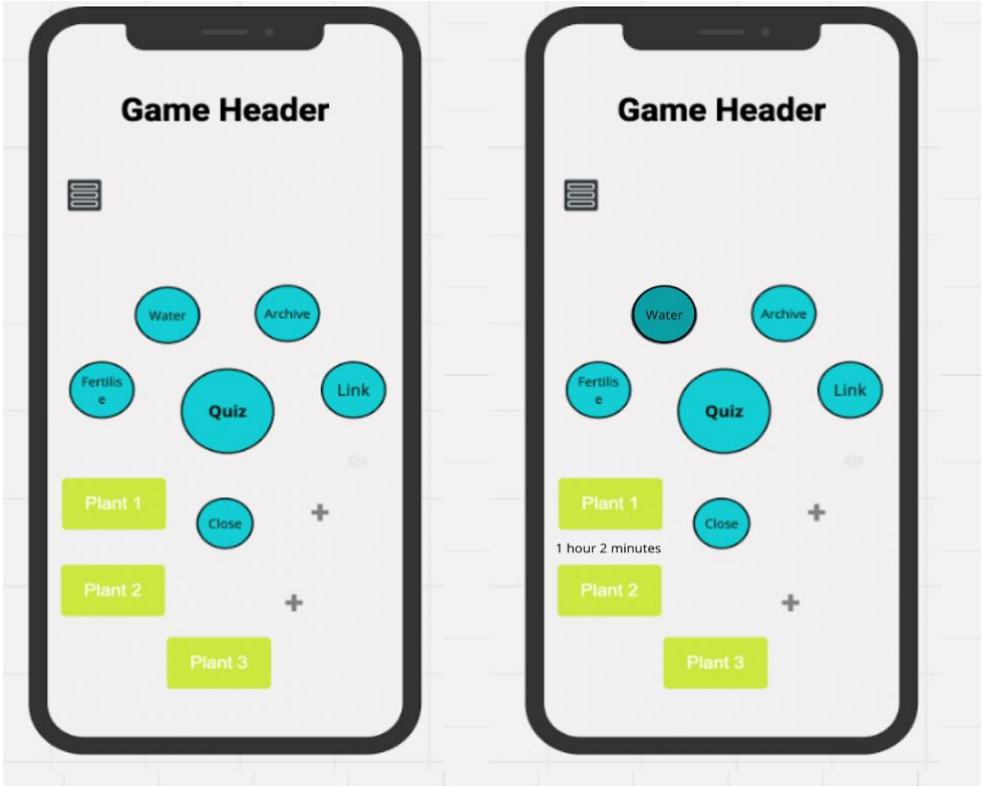
**Visual Design Guidelines:**

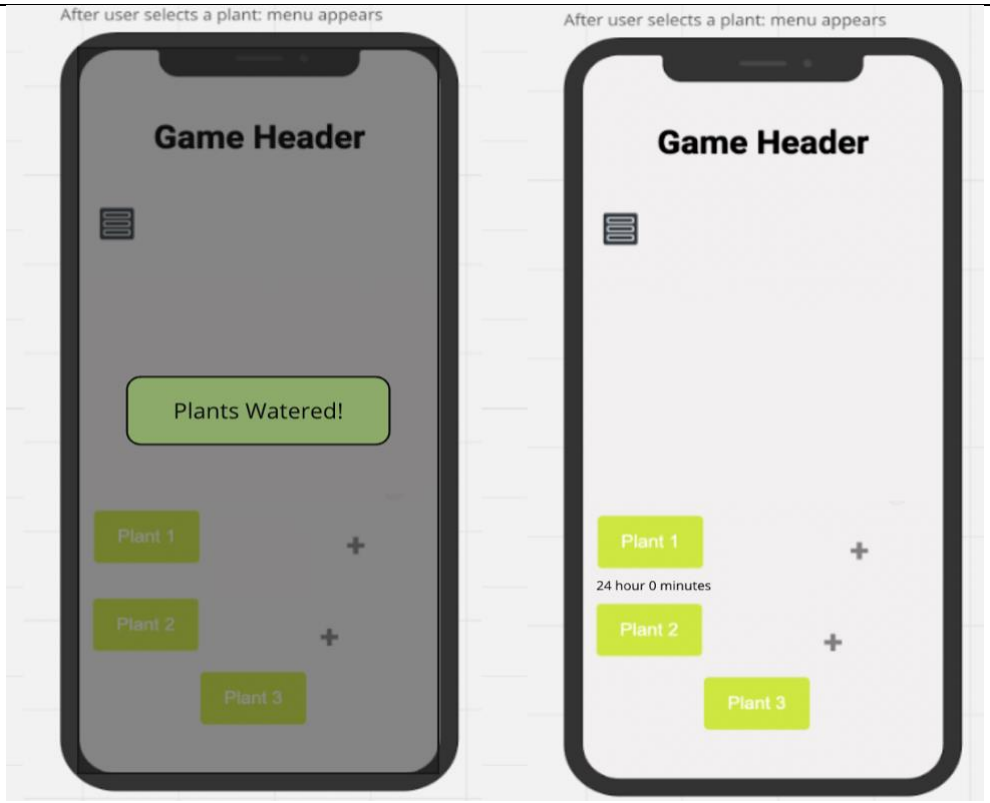
Consistent Styling: Maintain a consistent design language with the rest of the game, using the established typography, colour palette, and overall theme. This consistency contributes to a cohesive and polished visual identity.

The advice pop up menu should be shown beside the oracle

| US23-V1 | Water the plant by tapping on the “water” button in plant action menu. | | |
|---|---|---|---|
| Description | As a player, I want to water the plant by pressing the water button in plant action menu. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US23-V1-AC1.** | After pressing the water button, the plant menu will collapse. | **Pass** |
| | **US23-V1-AC2.** | A text saying "plant watered" is shown. | **Pass** |

| | | | |
|---|---|---|---|
| | US23-V1-AC3. | The text disappears after user taps elsewhere on screen. | Pass |
| | US23-V1-AC4. | The water timer of the plant will be reset. | Pending |
| Requirement Use Case | **Use Case:** Water the plant by tapping on the “water” button in plant action menu.<br>**Scope:** Plant care.<br>**Level:** Core game features.<br>**Context:**<br>(1) When the user presses the water button, the user could water their plant.<br>(2) The water button is shown as a part of the menu by default.<br>(3) The player can water the selected plant. To perform this action, the player needs to first select the targeted plant, click on it, and click on the water button.<br>**Frequency of occurrence:** Every time the user opened the plant menu and wishes to water a plant. | | |
| Design Use Case | **Scope:** Plant care.<br>**Level:** Core game features.<br>**Primary actors:** Player<br>**Description: P**layer can water the selected plant by clicking on the water button in plant action menu.<br>**Dependencies:**<br>(1) The user has successfully installed the game, and able to open the game to the main game screen.<br>(2) The user could add a plant and could access the plant menu.<br>**Assumptions:**<br>(1) Player has at least one plant in their virtual room or collection.<br>**Preconditions:** Player has expanded the plant action menu for a plant.<br>**Main Flow:** After the user arrive at a virtual room screen or collection screen, the user should be able to click on the water button and to water the plant.<br>**Postconditions:**<br>The plant menu will collapse and a text message "plant watered" will be shown, the in-game water timer will be reset.<br>**Alternative Flows:**<br>**Valid:** Player clicks on the plant and closes the plant menu.<br>**Invalid:**<br>(1) Player clicks on the water button, but the plant menu didn't collapse.<br>(2) Player clicks on the water button and the text does not appear.<br>(3) Player clicks on the water button; text appears but do not disappear after user clicks elsewhere.<br>(4) Player clicks on the water button, but the water timer does not reset.<br>**Frequency of occurrence:** Every time the user opened the plant menu and wishes to water a plant. | | |
| UI Design | **Key UI components:** | | |

| | |
|---|---|
| | Virtual room screen or plant collection screen<br>Plants in virtual room screen or plant collection screen<br>Expanded plant option menu.<br>Water button<br><br>**Wireframe:**<br>  |




**Interaction Flow:**

User clicks on the water button of the plant action menu.

The plant menu will collapse.

Text saying "plants is being watered" is shown briefly on screen.

The text will be closed when user press any spaces on the screen.

**UI Behaviour:**

The water button should have a smooth animated effect when pressed.

The water button should by default be shown on the plant menu.

The water text message should be shown with a smooth animation.

The water timer would be reset.

**Visual Design Guidelines:**

Consistent Styling: Maintain a consistent design language with the rest of the game, using the established typography, colour palette, and overall theme. This consistency contributes to a cohesive and polished visual identity.

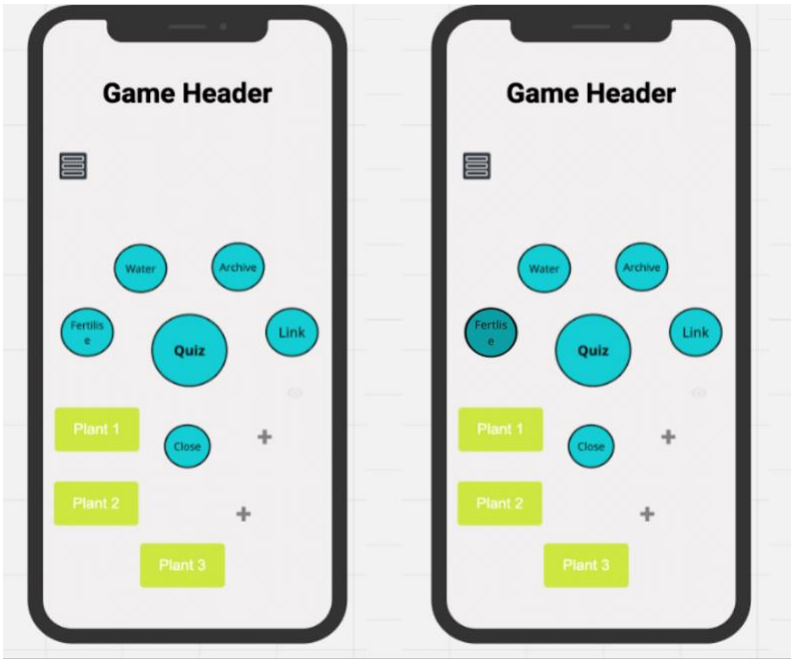
| US24-V1 | Fertilise the plant by tapping on the “fertilise” button in plant action menu. | | |
|---|---|---|---|
| Description | As a player, I want to fertilise the plant by pressing the fertilise button in plant action menu. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US24-V1-AC1.** | After pressing the fertilise button, the plant menu will collapse. | **Pass** |
| | **US24-V1-AC2.** | A text saying "plant fertilised" will be shown. | **Pass** |

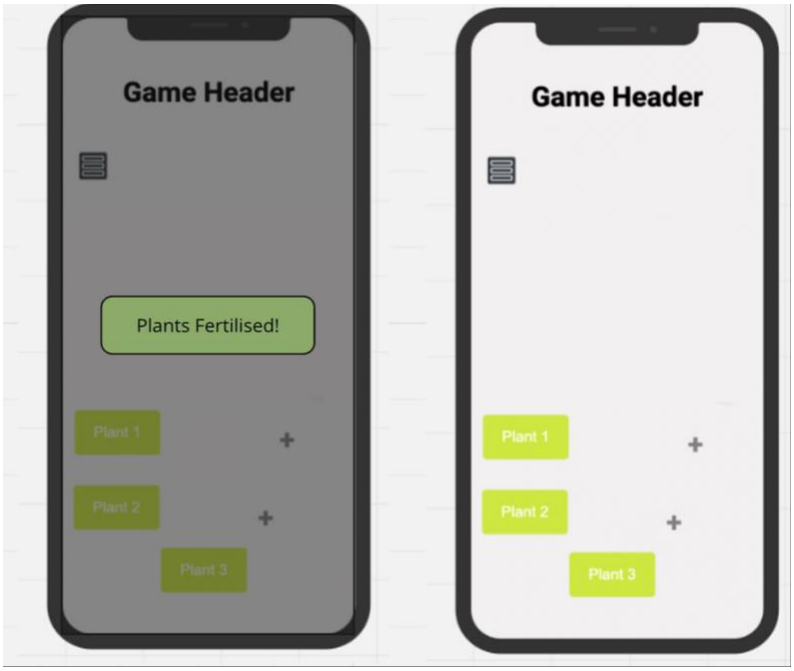
| | | | |
|---|---|---|---|
| | US24-V1-AC3 | The text disappears after user taps elsewhere on screen. | Pass |
| Requirement Use Case | **Use Case:** Fertilise the plant by tapping on the “fertilise” button in plant action menu.<br>**Scope:** Plant care.<br>**Level:** Core game features.<br>**Context:**<br>(1) When the user presses the fertilise button in plant action menu, the user could fertilise their plant.<br>(2) The fertilise button is shown as a part of the plant action menu by default.<br>(3) The player can fertilise the selected plant. To perform this action, the player needs to first select the targeted plant, click on it, and click on the fertilise button.<br>**Frequency of occurrence:** Every time the user opened the plant menu and wishes to fertilise a plant. | | |
| Design Use Case | **Scope:** Plant care.<br>**Level:** Core game features.<br>**Primary actors:** Player<br>**Description: P**layer can fertilise the selected plant by clicking on the fertilise button in plant action menu.<br>**Dependencies:**<br>(1) The user has successfully installed the game, and able to open the game to the main game screen.<br>(2) The user could add a plant and could access the plant menu.<br>**Assumptions:**<br>(1) Player has at least one plant in their virtual room or collection.<br>**Preconditions:** Player has entered the link screen.<br>**Main Flow:** After the user arrive at a virtual room screen or collection screen, the user should be able to click on the fertilise button and to fertilise the plant.<br>**Postconditions:**<br>The plant menu will collapse and a text message "plant fertilised" will be shown.<br>**Alternative Flows:**<br>**Valid:** Player clicks on the plant and closes the plant menu.<br>**Invalid:**<br>(1) Player clicks on the fertilise button, but the plant menu didn't collapse.<br>(2) Player clicks on the fertilise button and the text does not appear.<br>(3) Player clicks on the fertilise button; text appears but do not disappear after user clicks elsewhere.<br>**Frequency of occurrence:** Every time the user opened the plant menu and wishes to fertilise a plant. | | |
| UI Design | **Key UI components:**<br>Virtual room screen or plant collection screen<br>Plants in virtual room screen or plant collection screen. | | |

Expanded plant option menu.

Fertilise button.

**Wireframe:**




**Interaction Flow:**

User clicks on the fertilise button of the plant action menu.

The plant menu will collapse.

Text saying "plants is being fertilised" is shown briefly on screen.

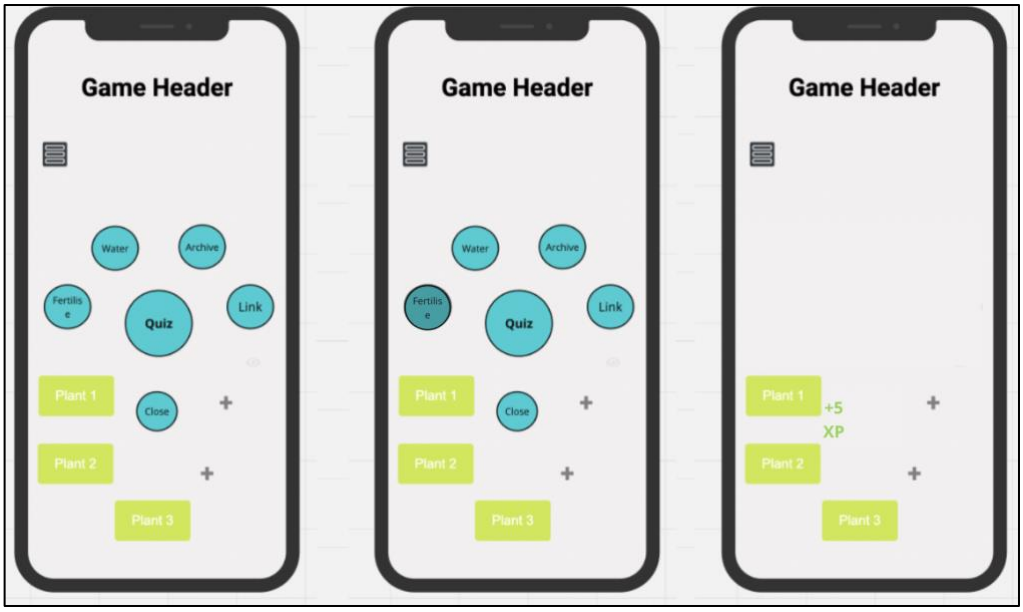
The text will be closed when user press any spaces on the screen.

**UI Behaviour:**

The fertilise button should have a smooth animated effect when pressed.

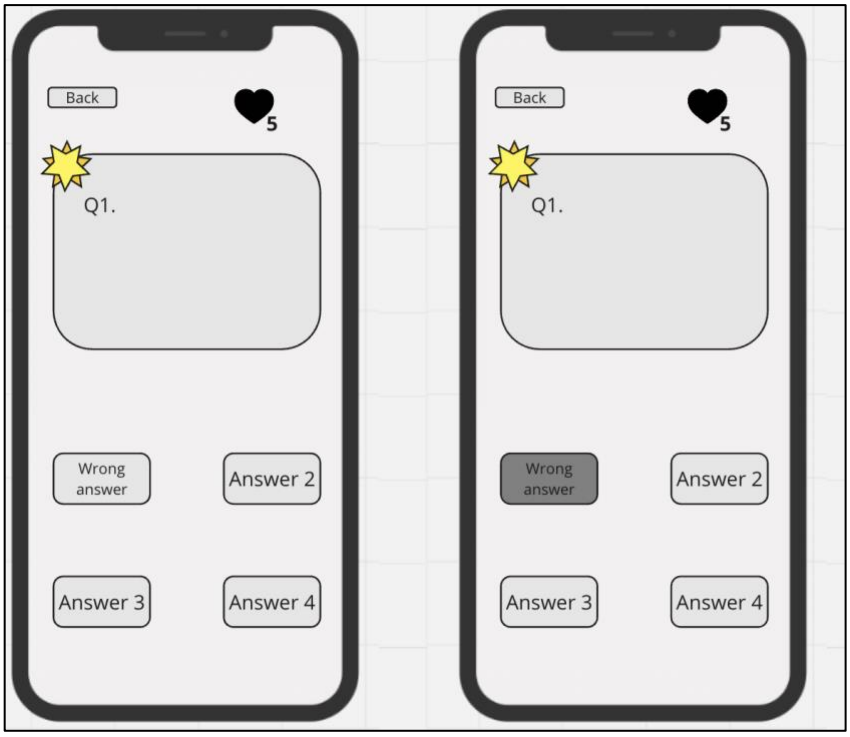
| | |
|---|---|
| | The fertilise button should by default be shown on the plant menu.<br>The fertilise text message should be shown with a smooth animation.<br>**Visual Design Guidelines:**<br>Consistent Styling: Maintain a consistent design language with the rest of the game, using the established typography, colour palette, and overall theme. This consistency contributes to a cohesive and polished visual identity. |

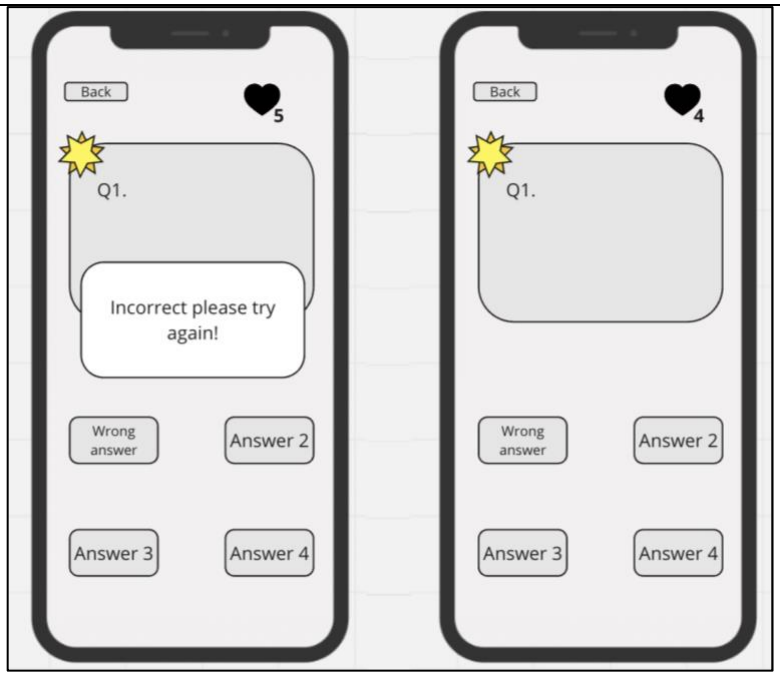
| US24-V2 | Fertilise the plant by tapping on the “fertilise” button in plant action menu. | | |
|---|---|---|---|
| Description | As a player, I want to fertilise the plant by pressing the fertilise button in plant action menu. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US24-V2-AC1.** | After pressing the fertilise button, the plant menu will collapse. | **Pass** |
| | **US24-V2-AC2.** | ~~**A text saying "plant fertilised" will be shown.**~~<br>A animated text saying "+5XP" will be shown. | **Pass** |
| | US24-V2-AC3 | ~~**The text disappears after user taps elsewhere on screen.**~~ | Deleted |
| Requirement Use Case | **Use Case:** Fertilise the plant by tapping on the “fertilise” button in plant action menu.<br>**Scope:** Plant care.<br>**Level:** Core game features.<br>**Context:**<br>(1) When the user presses the fertilise button in plant action menu, the user could fertilise their plant.<br>(2) The fertilise button is shown as a part of the plant action menu by default.<br>(3) The player can fertilise the selected plant. To perform this action, the player needs to first select the targeted plant, click on it, and click on the fertilise button.<br>**Frequency of occurrence:** Every time the user opened the plant menu and wishes to fertilise a plant. | | |
| Design Use Case | **Scope:** Plant care.<br>**Level:** Core game features.<br>**Primary actors:** Player<br>**Description: P**layer can fertilise the selected plant by clicking on the fertilise button in plant action menu.<br>**Dependencies:**<br>(1) The user has successfully installed the game, and able to open the game to the main game screen.<br>(2) The user could add a plant and could access the plant menu.<br>**Assumptions:**<br>(1) Player has at least one plant in their virtual room or collection. | | |

| | |
|---|---|
| | **Preconditions:**<br>(1) Player has expanded the plant action menu for a plant.<br>(2) The plant is available to be fertilised.<br>**Main Flow:** After the user arrive at a virtual room screen or collection screen, the user should be able to click on the fertilise button and to fertilise the plant.<br>**Postconditions:**<br>The plant menu will collapse and ~~a text message "plant fertilised" will be shown.~~ A "+5XP" animation appears on screen. A background timer is then reset to 24 hours.<br>**Alternative Flows:**<br>**Valid:** Player clicks on the plant and closes the plant menu.<br>**Invalid:**<br>(1) Player clicks on the fertilise button, but the plant menu didn't collapse.<br>(2) Player clicks on the fertilise button and the +5XP animation does not show<br>(3) Player clicks on the fertilise button again in less than 24 hours and the +5XP animation show up<br>**Frequency of occurrence:** Every time the user opened the plant menu and wishes to fertilise a plant. |
| UI Design | **Key UI components:**<br>Virtual room screen or plant collection screen<br>Plants in virtual room screen or plant collection screen.<br>Expanded plant option menu.<br>Fertilise button.<br>+5XP animation.<br>**Wireframe:**<br>  <br>**Interaction Flow:**<br>User clicks on the fertilise button of the plant action menu.<br>The plant menu will collapse. |

| | |
|---|---|
| | ~~Text saying "plants is being fertilised" is shown on screen.~~<br>~~The text will be closed when user press any spaces on the screen.~~<br>An animation saying +5XP is shown on screen.<br>**UI Behaviour:**<br>The fertilise button should have a smooth animated effect when pressed.<br>The fertilise button should by default be shown on the plant menu.<br>~~The fertilise text message should be shown with a smooth animation.~~<br>The +5XP animation should be smooth and seamless.<br>**Visual Design Guidelines:**<br>Consistent Styling: Maintain a consistent design language with the rest of the game, using the established typography, colour palette, and overall theme. This consistency contributes to a cohesive and polished visual identity. |

| US57-V1 | Lose "hearts" when selecting incorrect answer in the quiz. | | |
|---|---|---|---|
| Description | As a player, I want to have one "heart" deducted every time I selected an incorrect answer for when completing a quiz. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US27-V1-AC1.** | When a wrong answer is selected, one heart is deducted. | **Pass** |
| | **US27-V1-AC2.** | A message indicating a wrong answer will be shown. | **Pass** |
| Requirement Use Case | **Use Case:** Lose one "heart" deducted every time I selected an incorrect answer for when completing a quiz.<br>**Scope:** Quiz<br>**Level:** Core game features.<br>**Context:** When the user presses the wrong answer button, one heart would be deducted.<br>**Frequency of occurrence:** Every time the user is answering a quiz question and selects a wrong answer. | | |
| Design Use Case | **Scope:** Quiz<br>**Level:** Core game features.<br>**Primary actors:** Player<br>**Description:** Player loses one "heart" deducted every time I selected an incorrect answer for when completing a quiz.<br>**Dependencies:**<br>(1) The user has successfully installed and launched.<br>(2) The user could add a plant and could access the following: plant menu, quiz button, level selection screen.<br>**Assumptions:**<br>(1) Player has at least one plant in their virtual room or collection.<br>**Preconditions:** Player has entered the quiz screen. | | |

| | |
|---|---|
| | **Main Flow:** After the user arrive at the quiz screen (by successfully selecting a level of a selected plant or answering the question in a correct answer), the player has chosen one answer button that contains the wrong answer.<br>**Postconditions:**<br>A message would be shown to the user indicating wrong answer and one heart would be deducted from the original number of lives.<br>**Alternative Flows:**<br>**Valid:** Player clicks on back button and navigate to the main menu.<br>**Invalid:**<br>(1) Player clicks on the wrong answer, but the pop-up message does not show up.<br>(2) Player clicks on the wrong answer but the heart is not being deducted/ deducted multiple times.<br>**Frequency of occurrence:** Every time the user is answering a quiz question and selects a wrong answer. |
| UI Design | **Key UI components:**<br>Quiz screen<br>Answer buttons<br>Pop up message (wrong answer)<br>Remaining lives<br>**Wireframe:**<br>  |




**Interaction Flow:**

User press on the wrong answer

A message pop up indicating a wrong answer.

The user closes the message.

One heart is deducted.

**UI Behaviour:**

The answer button should have a smooth animated effect when pressed.

The pop-up message should appear in centre of screen.

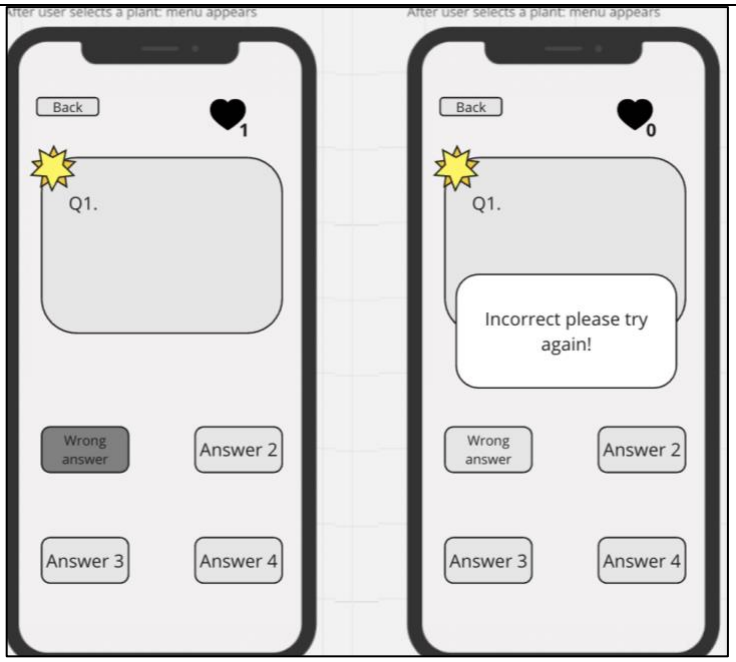
**Visual Design Guidelines:**

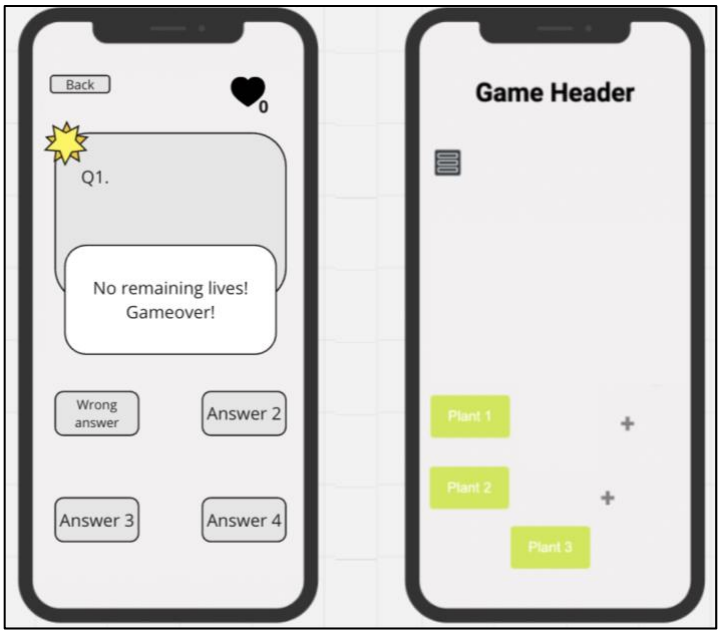
Consistent Styling: Maintain a consistent design language with the rest of the game, using the established typography, colour palette, and overall theme. This consistency contributes to a cohesive and polished visual identity.

The pop-up message should be in a colour that stands out from the context.

| US35-V1 | Game over after all "hearts" are lost. | | |
|---|---|---|---|
| Description | As a player, I want to enter a "game over" state once I have lost all "hearts" in a quiz. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US35-V1-AC1.** | After user selected a wrong answer, if no "hearts" remaining, a pop-up message stating "game over" is shown on screen. | **Pass** |
| | **US35-V1-AC2.** | Closing the game over screen will navigate the user back to main screen. | **Pass** |
| Requirement Use Case | **Use Case:** Game over if user selects a wrong answer but no hearts are remaining.<br>**Scope:** Quiz<br>**Level:** Core game features. | | |

| | |
|---|---|
| | **Context:** When the user presses the wrong answer button, hearts are deducted by one. If this leaves the user with zero hearts left then the game is over, and user is navigated back to main screen.<br>**Frequency of occurrence:** Every time player is at the quiz screen, answering the question with a wrong answer and heart is equal to 0 after deduction. |
| Design Use Case | **Scope:** Quiz<br>**Level:** Core game features.<br>**Primary actors:** Player<br>**Description:** Player would get 1 heart deducted by clicking on one of the answer buttons that contains the wrong answer, and game is over if user has run out of hearts.<br>**Dependencies:**<br>(1) The user has successfully installed and launched.<br>(2) The user could add a plant and could access the following: plant menu, quiz button, level selection screen.<br>**Assumptions:** Player has at least one plant in their virtual room or collection; there is only one heart remaining.<br>**Preconditions:** Player has entered the quiz screen.<br>**Main Flow:** After the user arrive at the quiz screen (by successfully selecting a level of a selected plant or answering the question in a correct answer), the player has chosen one answer button that contains the wrong answer.<br>**Postconditions:**<br>A message would be shown to the user indicating wrong answer and 1 heart would be deducted from the original number of lives. A game over message will be shown and after closing it the user would be navigated back to the main screen.<br>**Alternative Flows:**<br>**Valid:** Player clicks on back button and navigate to the main menu.<br>**Invalid:**<br>(1) Player clicks on the wrong answer, but the pop-up message does not show up.<br>(2) Player clicks on the wrong answer, but the heart is not being deducted/ deducted multiple times.<br>(3) The remaining life is 0 but the game over message does not show up.<br>(4) The user is not navigated to the main screen after closing the game over message.<br>**Frequency of occurrence:** Every time the user is answering a quiz question and selects a wrong answer. |
| UI Design | **Key UI components:**<br>Quiz screen<br>Answer button<br>Pop up message (wrong answer)<br>Remaining lives<br>Pop up message (gameover)<br>Main screen<br>**Wireframe:** |




Interaction Flow:

User press on the wrong answer

A message pop up indicating a wrong answer.

The user closes the message.

The heart is being deducted.

A message stating game over would be shown.

The user closes the message and be navigated back to main screen.

UI Behaviour:

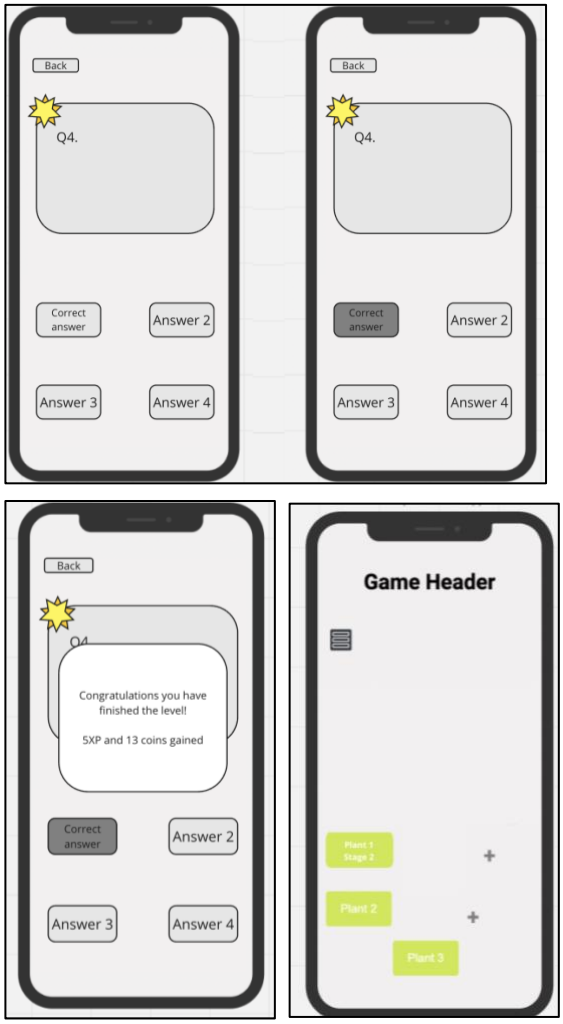
The answer button should have a smooth animated effect when pressed.

The messages should have a smooth animated effect when appearing.

The pop-up message should be rendered at the centre of screen.

| | |
|---|---|
| | **Visual Design Guidelines:**<br>Consistent Styling: Maintain a consistent design language with the rest of the game, using the established typography, colour palette, and overall theme. This consistency contributes to a cohesive and polished visual identity.<br>The pop-up message should be in a colour that stands out from the context. |

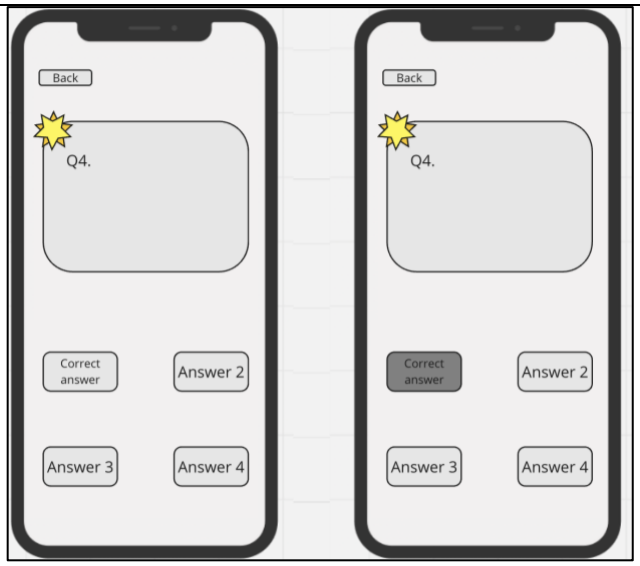
| | | | |
|---|---|---|---|
| US27-V1 | **Finish level.** | | |
| Description | As a player, I want to finish a level for a plant by completing the corresponding quiz. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US27-V1-AC1.** | After completing a quiz (without game over), a message shown on the quiz screen stating the user has completed the level. | **Pass** |
| | **US27-V1-AC2.** | The user will then be navigated back to main screen. | **Pass** |
| | **US27-V1-AC3.** | The plant that is the user has done the quiz for will grow to the next stage. | **Pass** |
| Requirement Use Case | **Use Case:** Game over if user selects a wrong answer but no hearts are remaining.<br>**Scope:** Quiz<br>**Level:** Core game features.<br>**Context:** When the user completes a quiz, a pop-up message will be shown to the user stating the completion of the level.<br>**Frequency of occurrence:** Every time player completes all questions in a quiz without game over. | | |
| Design Use Case | **Scope:** Quiz<br>**Level:** Core game features.<br>**Primary actors:** Player<br>**Description:** Player can finish the level of the selected plant by completing all questions in the quiz without losing all hearts.<br>**Dependencies:**<br>(1) The user has successfully installed and launched.<br>(2) The user could add a plant and could access the following: plant menu, quiz button, level selection screen.<br>**Assumptions:** Player has at least one plant in their virtual room or collection.<br>**Preconditions:** Player has completed the last question in the quiz without losing all hearts.<br>**Main Flow:** Player completes all questions in the quiz without losing all lives.<br>**Postconditions:**<br>(1) A message will pop up indicating the end of the level<br>(2) The selected plant in the main screen will grow to the next stage<br>(3) Player is navigated back to main screen.<br>**Alternative Flows:** | | |

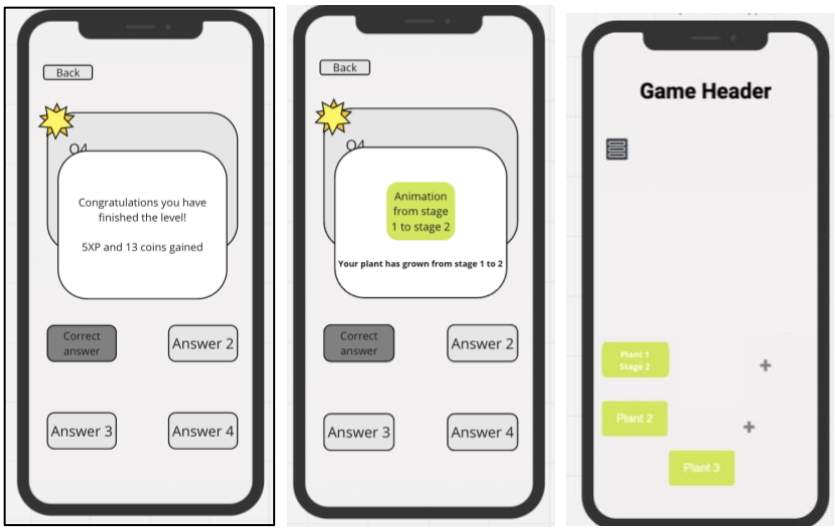
| | |
|---|---|
| | **Valid:** Player clicks on back button and navigate to the main menu.<br>**Invalid:**<br>(1) Player completes the quiz, but no pop-up appears.<br>(2) Pop-up does not navigate user back to main screen.<br>(3) The image of plant in virtual room/collection does not progress to next stage.<br>**Frequency of occurrence:** Every time player completes all questions in a quiz without game over. |
| UI Design | **Key UI components:**<br>Quiz screen<br>Answer button<br>Field for current question<br>Pop up message<br>Growth animation on main screen<br>Main screen<br>Selected plant<br>**Wireframe:**<br>  <br><br><br><br>**Interaction Flow:** |

| | |
|---|---|
| | User completes the quiz.<br>A message is shown indicating the end of level.<br>User closes the message.<br>Navigated back to the main screen.<br>The selected plant grows to the next stage.<br>**UI Behaviour:**<br>The answer button should have a smooth animated effect when pressed.<br>The messages should have a smooth animated effect when appearing.<br>The pop-up message should be rendered at the centre of screen.<br>**Visual Design Guidelines:**<br>Consistent Styling: Maintain a consistent design language with the rest of the game, using the established typography, colour palette, and overall theme. This consistency contributes to a cohesive and polished visual identity.<br>The pop-up message should be in a colour that stands out from the context. |

| US27-V2 | Finish level. | | |
|---|---|---|---|
| Description | As a player, I want to finish a level for a plant by completing the corresponding quiz. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US27-V2-AC1.** | After completing a quiz (without game over), a message shown on the quiz screen stating the user has completed the level. | **Pass** |
| | **US27-V2-AC2.** | The user will then be navigated back to main screen. | **Pass** |
| | **US27-V2-AC3.** | After the close the message the game will show a short animation showing that the plant growing from previous stage to current stage. | **Pass** |
| | **US27-V2-AC4.** | The plant that is the user has done the quiz for will grow to the next stage. | **Pass** |
| Requirement Use Case | **Use Case:** Game over if user selects a wrong answer but no hearts are remaining.<br>**Scope:** Quiz<br>**Level:** Core game features.<br>**Context:** When the user completes a quiz, a pop-up message will be shown to the user stating the completion of the level, plus a plant growth animation.<br>**Frequency of occurrence:** Every time player completes all questions in a quiz without game over. | | |
| Design Use Case | **Scope:** Quiz<br>**Level:** Core game features.<br>**Primary actors:** Player | | |

| | |
|---|---|
| | **Description:** Player can finish the level of the selected plant by completing all questions in the quiz without losing all hearts.<br>**Dependencies:**<br>(1) The user has successfully installed and launched.<br>(2) The user could add a plant and could access the following: plant menu, quiz button, level selection screen.<br>**Assumptions:** Player has at least one plant in their virtual room or collection.<br>**Preconditions:** Player has completed the last question in the quiz without losing all hearts.<br>**Main Flow:** Player completes all questions in the quiz without losing all lives.<br>**Postconditions:**<br>(1) A message will pop up indicating the end of the level.<br>(2) A plant growth animation is shown after user closes the pop-up.<br>(3) Player is navigated back to the main screen, where they can see, the plant has grown to the next stage.<br>**Alternative Flows:**<br>**Valid:** Player clicks on back button and navigate to the main menu.<br>**Invalid:**<br>(1) Player completes the quiz, but the message pop-up and plant animation pop-up do not appears correctly as expected.<br>(2) The plant animation pop-up does not navigate user back to main screen.<br>(3) The image of plant in virtual room/collection does not progress to next stage.<br>**Frequency of occurrence:** Every time player completes all questions in a quiz without game over. |
| UI Design | **Key UI components:**<br>Quiz screen<br>Answer button<br>Field for current question<br>Pop up message<br>Growth animation in second pop-up<br>Growth animation in main screen<br>Main screen<br>Selected plant<br>**Wireframe:** |




**Interaction Flow:**

User completes the quiz.

A message is shown indicating the end of level.

User closes the message.

Growth animation appears.

Navigated back to the main screen.

The selected plant grows to the next stage.

**UI Behaviour:**

The answer button should have a smooth animated effect when pressed.

The messages should have a smooth animated effect when appearing.

The pop-up message should be rendered at the centre of screen.

**Visual Design Guidelines:**

Consistent Styling: Maintain a consistent design language with the rest of the game, using the established typography, colour palette, and overall theme. This consistency contributes to a cohesive and polished visual identity.

The pop-up message should be in a colour that stands out from the context.

Additional tests performed:

| Criteria ID | Description | Test Status |
|---|---|---|
| US46-V1-AC6. | Correct answers in quizzes can contribute to rewards in the game, such as coins, experience points (XP), or unlocking new content. | Complete |
| US46-V1-AC7. | The quiz interface should be visually appealing and consistent with the game's overall design. | Complete |
| US46-V1-AC8. | Text and buttons should be clearly readable and accessible for players of all ages. | Complete |

### 3.6.6. CRC Cards

Note that all CRC cards are reverse engineered in Sprint 7 based on a completed code base. This is as an unfortunate outcome of the team's failure to sequentially implement the software process model. Therefore, the CRC cards were not developed in iterative versions. All cards below are in accordance with the final version of the completed game.

The following CRC cards were developed for the functionalities built in Sprint 6.

| Class: ArchiveModalSelection (Component) | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Display a modal that allows users to select plants from an archive.<br><br>Show archived plants in a horizontally scrollable view.<br><br>Handle user interactions for selecting a plant to remove from the archive. | plantsDataContext<br><br>plantsConfig<br><br>GameText |

Relevant user stories: US6-V1, US13-V1, US14-V1, US18-V1, US19-V1, US23-V1, US24-V2

| Class: CollectionButton Component | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Render a button with an icon to navigate to the Collection Screen. | HomeStackNavigator |

Relevant user stories: US12-V1

| Class: CollectionScreen | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Display the user's collection of archived plants. | plantsDataContext |

| | |
|---|---|
| Present each plant in a visually appealing way with styling and layout.<br><br>Integrate with Plant component to display individual plant details. | PlantComponent |

Relevant user stories: US13-V1, US14-V1

| Class: PlantSelectionModal | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Display a modal for selecting a plant or accessing the plant archive.<br><br>Enable users to select a plant from the collection to add to their game environment.<br><br>Provide a button to access the plant collection/archive. | plantsDataContext |

Relevant user stories: US17-V1, US6-V1, US14-V1, US18-V1

| Class: GameStatsButton Component | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Render a button with a game statistics icon to navigate to the Game Stats Screen | HomeStackNavigator |

Relevant user stories: US14-V1, US16-V1, US50-V1

| Class: GameStatsScreen | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Display the mastery levels of different plants in the game.<br><br>Manage and display a modal to show detailed information about a selected plant.<br><br>Format and present plant details such as name, height, type, colours, care instructions, and more.<br><br>Show progress bars to represent plant mastery levels. | plantsDataContext<br><br>GameText |

Relevant user stories: US12-V1, US18-V1, US19-V1, US23-V1, US24-V2

| Class: ScaleAnimation |
|---|

| Responsibilities: | Collaborators |
|---|---|
| Provide a scaling animation effect for its child components.<br><br>Apply an opacity change corresponding to the scaling effect for visual enhancement. | |

Relevant user stories: US12-V1, US18-V1, US19-V1, US23-V1, US24-V2

| Class: LevelSelectionScreen | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Display the levels available for a selected plant.<br><br>Handle user interactions for selecting levels.<br><br>Access game progress data.<br><br>Implement logic to determine the state of each level (completed, unlocked, or locked). | playerConfigContext<br><br>completedLevelsContext<br><br>HeartsDisplay, CoinDisplay, Oracle |

Relevant user stories: US51-V1, US25-V2, US27-V1

| Class: Loading Screen | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Display a loading screen to the user.<br><br>Manage the loading of various data such as plants, player state, completed levels, species progress, and configuration.<br><br>Provide feedback to the user about the loading status by displaying loading messages. | plantsDataContext<br><br>playerConfigContext<br><br>completedLevelsContext |

Relevant user stories: US3-V1

| Class: CompletedLevelsContext | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Maintain and manage the state of completed levels for different species in the game. | plantsConfig |

| | |
|---|---|
| Update the completed level of a specific species.<br><br>Persist changes in completed levels to local storage. | |

Relevant user stories: US2-V1, US51-V1, US25-V2, US27-V1

| Class: levelsConfig | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Hold configuration for the levels of different species in the game.<br><br>Define the total number of levels available for each species.<br><br>Specify the experience points (XP) reward for each level of every species.<br><br>Maintain a record of completed levels for each species | plantsConfig<br><br>plantsTriviaConfig |

Relevant user stories: US51-V1, US25-V2, US27-V1

| Class: PlantsDataContext | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Maintain the state of plantDataContext and plantsConfig<br><br>Provide functions to update plantDataContext and plantsConfig | plantsConfig |

Relevant user stories: US4-V2, US2-V1, US3-V1, US6-V1, US12-V1, US16-V1, US13-V1, US14-V1, US18-V1, US19-V1

| Class: speciesProgressContext | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Maintain and track the progress of each species defined in the plant's configuration.<br><br>Initialize the default progress state for each species based on the plant's configuration.<br><br>Provide functionality to update the progress of a specific species. | plantsConfig |

| | |
|---|---|
| Supply context and access to species progress data and update functionality throughout the application.<br><br>Ensure that the progress data is consistently updated and saved as changes occur. | |

Relevant user stories: US4-V2, US2-V1, US3-V1, US16-V1

| Class: CollectionStyles | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Define visual appearance of various components in the application.<br><br>Provide a consistent design language across different parts of the app.<br><br>Ensure scalability of UI elements across different device sizes.<br><br>Manage layout, color schemes, fonts, and spacing of UI elements. | Images and Assets |

Relevant user stories: All

| Class: GameText | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Render text content.<br><br>Apply a custom font and additional styles to the text.<br><br>Provide a consistent text appearance across different parts of the game where this component is used.<br><br>Enhance readability and maintain the game's visual theme through text styling. | Menu and Components |

Relevant user stories: All

| Class: PlantStyles | |
|---|---|
| **Responsibilities:** | **Collaborators** |
| Define and manage the visual presentation of various UI components used in modals, including plant selection and menu overlays. | Menu, Icons and Components |

| | |
|---|---|
| Provide consistent styling for buttons, cards, images, and other elements within the modals.<br><br>Control the layout of elements like scroll views, buttons, and image containers.<br><br>Apply shadows, opacity, margins, padding, and other visual properties to enhance the user interface. | |

Relevant user stories: US6-V1, US12-V1, US13-V1, US14-V1, US18-V1, US19-V1, US23-V1, US24-V2

## 3.6.7. Team Stand-up

### 3.6.7.1. Team Stand-up 1

| | |
|---|---|
| Date and Time | 24th Nov 2023, 12:00 noon |
| Location | Microsoft Teams |
| Attendees | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |
| Meeting | **Updates on task-wise progress:**<br><br>**Claire:** Process documentation on track, Sprint 5 complete now moving on to previous sprints. Lost with user story development due to isolation of workstreams and rushed delivery of new features.<br><br>**Udit:** Heart functionality and level lock complete.<br><br>**Alec:** Collection screen (archive) function complete, including "Select from archive" when adding plant.<br><br>**Marat:** Making progress in asset generation for more plants, taking time to photoshop each asset. Completed "Easy" and "Medium" level plants. Added quiz questions for different plant types. Still to update level configuration file.<br><br>**Eddie:** Making a good start to "Link to real plant" function, high-level code structure is in place, work pending on features.<br><br>**Ivan:** Working on save function on new branch. So far able to save plant type and location but saving plant progress has been challenging. Conceptualised logic: Quiz complete à fetch plant ID, level, progress etc. into a list passed onto home screen à pass onto background image component à plant component. The challenge being saved data could not be access in several screens, so the variables need to be passed around and saved in "progress".<br><br>**Favour:** Created contextual variables which can be imported and called from contextual.js, also avoid having dependent components between each other's code. All data (plant data, font data, and anything else that will need to load) is now loaded once at the loading screen (which appears very briefly before the main screen), available throughout the app, replacing dependencies between different components where variables are passed repeatedly between components. Hence now each component runs independently from other components.<br><br>**Proposed future process change:**<br><br>After this sprint, the team aims to only keep 2-3 members on the coding stream while everyone else move to documentation to work on product documentation. |
| Task allocations | **Favour** to continue with styling-related features and code configuration.<br><br>**Eddie** to continue with link function, with help from Favour and Ivan from Sunday onwards. |

| | |
|---|---|
| | **Ivan** to continue with "save" function.<br>**Marat** to continue with plant data and asset development. |
| Meeting reflections and analysis | **Analysis / reflections of current standpoint:**<br>During the Wednesday customer meeting, tasks were allocated by functionalities without any detailed coding guidelines. This meant that since several members do not have JavaScript / React experience, the exact level of difficulty for each individual task will only become apparent after some trial and error. Hence, the fact that everyone has already started on their work over the past 1.5 days had been immensely helpful for mapping out the weight of each task, allowing those without prior experience in React to share their high-level thoughts processes and logical approach to their tasks and receive feedback from more experienced members of the team.<br>Favour's expertise with React continues to be assurance to the team's work. The creation of contextual variables, for instance, is a brilliant addition which the team otherwise would not have been able to incorporate, resolving several issues (pertaining to dependent components) troubling other members. With that said, it is also commendable that members without JavaScript and React experience have been able to pick up the framework very quickly and start building full-fledged functions. This further proves that the bold switch from C# and Unity to JavaScript and React Native is the best-suited decision for team circumstances.<br>To address progress delays over previous weeks, the team adopted the quick-fix strategy of concentrating effort to game development before refocusing on product documentation. Sadly, this partially defeats the purpose having user story design in the development cycle, which is to guide the software development process rather than reverse engineered after the features are built. Further discussions on the reasons and implications of this misstep are included in the project retrospect.<br>**Priorities until next customer meeting:**<br>Claire to continue documenting processes for previous sprints.<br>Everyone to continue with coding tasks. |

### 3.6.7.2. Team Stand-up 2

| | |
|---|---|
| Date and Time | 28th Nov 2023, 12 noon |
| Location | Microsoft Teams |
| Attendees | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |
| Meeting | **Documentation update:**<br>Process document complete for Sprint 3, ready for review on Wednesday TA meeting.<br>**Game features update:**<br>**(1) Completed:**<br>**Archive / collection function:** Complete with remaining bug being that when a plant is added to mains screen from the archive it doesn't get removed from the archive, due to interference from the "Save" function.<br>**Quiz / Trivia:** Questions are randomised, future levels are locked, and the quiz at each level incorporates two random questions from previous levels, information shown for each plant before starting a quiz.<br>**Plant data:** Many new plants are added to the game database.<br>**Link to real plant:** Add and save image function complete. Pending plant stage setting, |

| | |
|---|---|
| | and last time watered/fertilised, and background timer.<br>**Restructuring main menu:** Collection and mastery level functions now moved to main screen as icons.<br>**Styling:** Various additions for visual aesthetics and user friendliness.<br>**(2) Remaining features to be built:**<br>**Oracle:** Oracle to give random advice on the main screen, and in the quiz screen the oracle need to provide feedback.<br>**Link to real plant:** All functionalities aside from adding photo and setting plant nickname.<br>**Background timer:** For both plant that have achieved mastery and linked plant instances.<br>**Shop feature**: To allow users to purchase skins and backgrounds with "coins" earned.<br>**XP and Coins:** Players should earn XP and coins each time they complete a quiz. |
| Task allocations | **Udit** to start building the coin and shop features.<br>**Favour** to start on Oracle advice and continue working on "Save data" function with contextual variables.<br>**Eddie** and **Ivan** to continue looking at Link plant feature.<br>**Marat** to continue adding quiz questions for different plants.<br>**Alec** to continue with removing plant from archive. |
| Meeting reflections and analysis | **Current standpoint:**<br>Following the summary in Stand-up 2, the team is in a good position on the code development front, ready to shift towards documentation. As mentioned, the team has reached a critical point of simultaneously closing out both code development and documentation. Recognising the significance of both streams, its co-existence requires prudent navigation and economical project-management.<br>**Priorities until next customer meeting:**<br>Proceed with allocations made in stand-up 2. |

### 3.6.8. Exception Handling

| Exception | Actions | Outcome |
|---|---|---|
| **Carried over from Sprint 5** | | |
| Change of team structure and roles | **Sprint 6 actions:**<br>Everyone agreed to only keep 2-3 people on the coding stream in Sprint 7 while everyone else move to the documentation stream.<br>**Sprint 5 actions:**<br>Previously Favour and Claire were co-leading the documentation workstream. Since Favour has moved to be the oversight of coding, Claire is now solely in charge of keeping documentation. In upcoming weeks, especially by the time people wrap up code development work, the team has to split up documentation tasks again in some way given the heaviness of the workload. | Success |
| Falling behind on | **Sprint 6 actions:** | Pending |

| | | |
|---|---|---|
| documentation. | Progress still behind for product documentation. More people will move to documentation in Sprint 7 where hopefully the problem could be resolved.<br>**Sprint 5 actions:**<br>As Claire was amongst those struggling with CM50260, documentation was put on hold throughout Sprint 5. The process documentation so far contains only meeting minutes and backlog items, with substantial content still awaiting completion. On the product documentation front, user story development is also at an elementary stage, with only a handful of stories elaborated with use cases and test requirements. UI designs have been completely revamped after moving to React Native, hence needs recreation. As the team has officially kicked-off game development, it is crucial that the relevant documentation is in pace with feature delivery. This must be resolved in Sprint 6. | |
| **New exceptions raised** | | |
| User story development proving to be a difficult task as game feature development is condensed into one week without time intervals to develop use cases. | People from the development stream will move back to documentation in Sprint 7 where they can reverse-engineer what they have coded into user stories and use cases. | Pending |

### 3.6.9. Sprint 6 Backlog and Completion Status

| TASK | Relevant User Story | DUE | PRIORITY | ASSIGNED TO | STATUS |
|---|---|---|---|---|---|
| Add feedback/boink sound to touch feedback. | | 29 Nov | High | Favour | Complete |
| Product Documentation & Process Documentation for at least 1 fully completed Sprint. | | 29 Nov | High | Claire | Complete |
| Implement ability to obtain new rooms/backgrounds based on overall XP level. | US9-V1<br>US10-V1 | 29 Nov | Medium | Favour | Pending |
| Implementation of link to real plant function. | US19-V1<br>US20-V1<br>US21-V1<br>US50-V1 | 29 Nov | High | Eddie | Complete |
| Progress of plant linked to plant instances not specific plant type. | US2-V1<br>US3-V1<br>US23-V1<br>US24-V2 | 29 Nov | High | Favour | Complete |
| Notification to fertilize and water plants as needed. | | 29 Nov | High | Favour, Ivan | Deleted |
| Timers should start once mastery is completed for plant \|\| Player needs to water and fertilize regularly (based on real plant requirements) | US23-V1<br>US24-V2<br>US50-V1 | 29 Nov | High | Ivan | Complete |

| | | | | | |
|---|---|---|---|---|---|
| When entering plant/real life link, oracle should give tips on how to indicate its data. e.g. Height correlating with stage. | US21-V1<br>US50-V1 | 29 Nov | High | Ivan | Pending |
| Oracle should also give regular advice on your plants. | US44-V1 | 29 Nov | Medium | Not assigned | Pending |
| Five hearts system - hearts reset every hour. If you lose all your hearts, game over and you must wait. | US8-V1<br>US11-V1<br>US57-V2<br>US35-V1 | 29 Nov | High | Not assigned | Pending |
| Need to save any state variables to user device. These variables should then be compiled in a JSON file which will be the data uploaded for each Account. | US2-V1 | 29 Nov | High | Favour, Alec | Complete |
| User should not be able to play following levels if they have not completed current level. | US51-V1<br>US25-V2 | 29 Nov | High | Udit | Complete |
| Separate the menu icons into 16bit icons along the side of the screen. | US4-V2<br>US46-V2 | 29 Nov | High | Marat | Complete |
| Implement archive button in plant menu and implement ability to store plants in collection view. | US4-V2<br>US6-V1<br>US12-V1<br>US14-V1<br>US18-V1<br>US46-V2 | 29 Nov | High | Alec | Complete |
| Require consistent styles for Mastery/Game Stats Screen, Level Select Screen, and Quiz Screen. | | 29 Nov | High | Favour | Complete |
| Add user stories/use cases template to Sprint Documentation | | 29 Nov | High | Claire | Pending |
| Update user stories with acceptance criteria & story points | | 29 Nov | High | Claire | Pending |
| Update config files with information for ten plants one tutorial, three easy, three medium, three hard] | US98-V1<br>US16-V1 | 29 Nov | High | Marat | Complete |
| A few speech bubbles from the Oracle about how to operate game + first quiz for cactus (if player has not used it before) | US98-V1 | 29 Nov | High | Favour | Complete |
| Show a little information before the quiz about learning the plant. | US98-V1 | 29 Nov | High | Udit | Complete |
| Code reconfiguration: contextual variables and loading screen. | US3-V1 | 29 Nov | High | Favour | Complete |
| Implement the coins system: user earns coins after completing quiz. | US7-V1 | 29 Nov | High | Udit | Complete |
| Complete plant information for plant mastery cards. | US16-V1 | 29 Nov | High | Marat | Complete |

**Note to completion status:**

1. Notification feature deleted due to unexpected complication/difficulty.

### 3.6.10. Individual Retrospectives

Each member reflected on their experience during Sprint 6, key points are summarised as follows.

Sprint 6 has so far been the most positive amongst all previous sprints. As most of the team (except for Claire staying on documentation) had dived into code development, it was also the first time that the team was able to put into practice some of the Agile collaborative techniques introduced in the course lecture. Ivan and Eddie have highlighted how they were able to adopt Pair Programming when developing the "Link to real plant" function, and how much it contributed their success in building some of the features. Both agreed that having the inputs of two people is tremendously helpful when neither has had significant React development experience before. The most significant benefit of the practice has been the opportunity for each of them to take on distinct roles, where Eddie was primarily the "Navigator" (thinking ahead of the overall strategic design) and Ivan the "Driver" (writing immediate code), thereby enabling a more comprehensive approach to solving the task.

Moreover, frequent communication was shortlisted as the strongest contributor for this week's progress. It was evident that everyone took the extra mile to keep the whole team informed of individual accomplishments and difficulties. Daily communication was carried out on the WhatsApp group, where everyone would post timely updates on their individual undertakings, as well as notifying everyone before making a GitHub "push". Consequently, each peer when developing their own programmes was mindful of the most recent additions to the code, which helped a lot with synchronisation and avoiding any merge conflicts.

Contributions over Sprint 6 is evenly distributed across the team, measured in terms of hours spent on the project, with everyone spending at least 20 hours. This is not surprising as everyone was able to deliver on most, if not all, of their allocated backlog tasks. Meanwhile, there is a drastic increase of hours spent on the project compared to previous sprints, suggesting that everyone has delivered on their promise to make this project their foremost priority as all other mid-semester coursework out of the way.

By end of Sprint 6, it had become apparent that a collaborative spirit and team morale played a massive role in delivering promising outcomes. Everyone, in one way or another, expressed their appreciation to one or more team members in their reflections, and everyone had emphasized how effortless and enjoyable the teamwork experience has been so far. A solid trust seems to have formed between all members of the team knowing that everyone is doing the best they could on their responsibilities. Whilst everyone had varying degrees of expertise, it is obvious that everyone had the best interest of the whole team, and was willing to contribute wherever they can, whether by taking the time to learn new toolkits, or taking on extra work in specialised areas. All members commented on the miraculous progress on code development accomplished in just a week's time, taking the team from a state of lagging behind to already having a presentable game. See below is a summary of how everyone's feeling out of scale of 1 to 5 (1=worst, 5=best) regarding this Sprint as well as the project overall.

| Band | Sprint 6 | Project |
|---|---|---|
| 5 (Best) | 2 | 5 |
| 4 | 5 | 2 |
| 3 | - | - |
| 2 | - | - |
| 1 (Worst) | - | - |

## 3.7. Sprint 7: 29th November – 5th December

### 3.7.1. Overview

In Sprint 7, the team was divided into two parallel working groups. Favour and Ivan and Udit stayed on the coding stream to complete remaining features while everyone else focused exclusively on documentation. On the coding side, the goal is to complete the game by end of Sprint 7, after which everyone will spend the final week preparing for submission. On the documentation side, the goal is to finish process documentation for all Sprints excluding Sprints 7 and 8, as well as completion of all product documentation elements (user stories, use cases, UI design, tests, CRC cards). Meanwhile, the team also aim to make a start on the installation guide and maintenance guide as part of the coursework submission requirements.

### 3.7.2. Review

From Wednesday 29th Nov to Friday 1st December, the team aimed to wrap up any unfinished code carried over from Sprint 6. From Friday onwards, documentation is the primary focus for everyone except Favour, Ivan, and Udit, both of whom stayed to complete game features requested by the customer. Progress stayed steady and linear throughout the week and both workstreams were able to deliver on their assigned responsibilities. Aside from hard work and long hours for all peers, Sprint 7 was rather uneventful as everyone stayed deep in focus on their tasks. By end of week, process documentation was completed for Sprints 1 to 6, with all product document components built by end of Sprint 6. On the coding side all remaining features were completed as requested by the customer, including oracle engagement, player level, shop feature, background timer, earning coins and XP after completing quizzes.

### 3.7.3. Summary of Key Events

- Completion of the game product.
- Completion of process document for Sprints 1-6.
- Completion of product documentation components for features delivered before Sprint 7.

### 3.7.4. Sprint 7 Preparation

#### 3.7.4.1. Customer Meeting and Planning Discussion

| | |
|---|---|
| **Date** | 19th Nov 2023 |
| **Location** | Campus – Chancellor Building 4.7 |
| **Attendees** | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |
| **Customer Meeting** | **Presentations:**<br>*"*" indicating that customer has made a comment, recorded in the "Customer concerns" section below.*<br>• **Player level measurements:** Implementation of the "Hearts" for wrong answers in the quiz, and the "Coins" for purchases in the "shop" (not yet implemented).<br>• **"Collection":** Now able to move plant from a background to archive which goes into your collection. Once in the collection you can still perform actions to archived plants such as water, fertilise, link, quiz. |

| | |
|---|---|
| | • **Level selection:** All levels above the current level are now locked until the player have passed the current level.<br>• **Quiz:** Player is shown an info card about the plant before taking each quiz. The information increases in specificity and detail as level advances. During the quiz the Oracle asks the quiz questions. Upon finishing there is a congratulation prompt after which the player can return to main screen to see that the plant has "grown". In each quiz at level 2 and above, 2 questions from previous levels selected at random are incorporated into the current quiz for knowledge retention.<br>• **Link to real life plant:** Opportunity to add photo of your real-life plant and to set what level it is currently at. Instructions are given to help player determine current levels (e.g., plant heights of certain range). Timer feature still in development. *<br>• **General saving function:** Saving progress of the game every time player exists app.<br>• **Incomplete features:**<br>  • **Shop functionalities (yet to be implemented)**<br>  • **Earning XP and coins as player levels up*.**<br>  • **Skins:** Replacing creative skins (e.g., pink) with scientific variations of the same plant as icons, available to purchase at the shop.<br>  • **Background:** Option to unlock different rooms with game progression (currently at single-background mode the plant can be archived if requiring more space).<br>**Customer comments:**<br>• Customer asks whether the team is on track to completing game development next week to leave a week of thorough testing.<br>Response: Three main features remaining before development completion: Shop, player level (XP), Plant link.<br>• Customer suggests the team focus on "Player Level" and "Plant Link" features as they are crucial to a complete product, whilst the Shop can be viewed as an add-on.<br>• Customer stresses that all development work should stop by end of next Sprint as last-minute development can lead to bugs and unforeseen challenges potentially hindering delivery.<br>• Customer happy to accept anything of an executable file for installation. A single entity preferred over command line prompts.<br>Customer happy with the project progress and team dynamics. |
| Planning Discussion | **Additions/changes to product design based on customer meeting:**<br>No additions arise from the customer meeting.<br>**Analysis / reflections of current standpoint:**<br>With two weeks until submission, the team recognises that the state of documentation is in grave danger despite good progress on game development with only few (but crucial) features remaining. The problem at hand is thus to allocate team resources to deliver on both workstreams.<br>Considering many people may still have tasks carried over from Sprint 6, two more days (until Friday) are given for everyone to wrap up any incomplete code, from Friday onwards everyone except for Favour and Ivan will move to user story development and other elements of product documentation.<br>**Determining Sprint priorities:**<br>User story development and CRC cards for product documentation is currently the biggest bulk of work pending, followed by completion of remaining game features. |

| | |
|---|---|
| | The Sprint 7 backlog can be found in Section 3.7.9. |

### 3.7.5. User Stories, Acceptance Criteria (and Test Outcome), Use Cases, and UI Design

In Sprint 7, the following user stories have been developed and built into the game prototype.

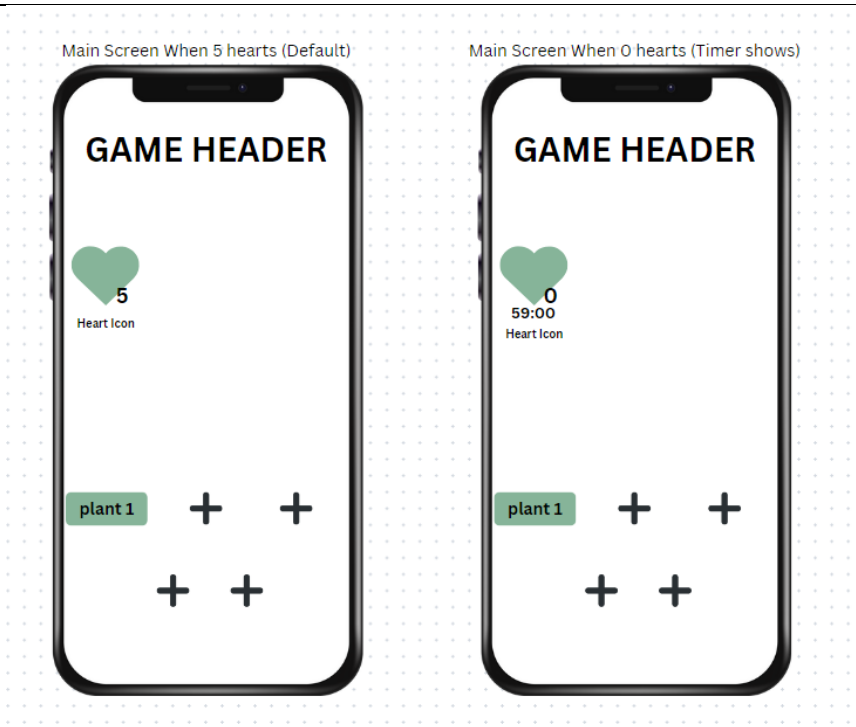
Note that majority the stories below are reverse engineered in Sprint 8 (after the features are already developed into the game) as an unfortunate outcome of the team's failure to sequentially implement the software process model. Therefore, some UI designs were developed without any wireframes as guidance, in which cases a snapshot of the UI in the end-product is presented for illustration.

Full list of user stories relevant to Sprint 7.

| User Story ID | Description |
|---|---|
| US8-V1 | Hearts should automatically replenish once every hour, capped at 5. |
| US40-V1 | Players can trade coins for hearts at the in-game shop. |
| US39-V1 | Players can trade coins for plant skins at the in-game shop. |
| US44-V1 | Receive plant care tips from the Oracle. |
| US9-V1 | Gaining XPs as player progress through the game. |
| US10-V1 | Unlock new rooms to increase available place space. |

| US8-V1 | Hearts should automatically replenish once every hour, capped at 5. | | |
|---|---|---|---|
| Description | As a player, I want the hearts in the game to automatically increase once every hour. This ensures that I receive additional hearts over time without requiring manual intervention. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US8-V1-AC1.** | If number of hearts is below 5, the hearts should automatically increase by one every hour. | **Pass** |
| | **US8-V1-AC2.** | The increase in hearts should happen in the background without disrupting the player's experience. | **Pass** |
| | **US8-V1-AC3.** | The player should be able to see a timer next to the heart icon on main screen counting down to the next gain. | **Pass** |
| | **US8-V1-AC4.** | Hearts should stop increasing after reaching 5. | **Pass** |
| Requirement Use Case | **Use Case:** Hearts increasing by one every hour if below 5.<br>**Scope:** Game set-up<br>**Level:** Core game features.<br>**Context:** | | |

| | |
|---|---|
| | (1) As the player engages with the game, the hearts that were lost in quizzes should be gradually replenished, providing a steady flow of this in-game resource.<br>(2) This increase in hearts occurs in the background even when the player is not actively playing the game, encouraging consistent engagement.<br>**Frequency of occurrence:** Every hour, in the background. |
| Design Use Case | **Scope:** Game set-up<br>**Level:** Core game features.<br>**Primary actors:** Player<br>**Description:** Player should see an automatic increase in hearts with a timer under the heart icon counting down to the next increase. The increase should happen once every hour without any manual action on their part. This contributes to a sense of progression and allows players to accumulate resources over time. The increase stops if the player has reached 5 hearts.<br>**Dependencies:**<br>(1) The user has successfully installed and launched the game.<br>(2) The game assets related to hearts are correctly integrated into the code base.<br>**Assumptions:** Player has at least one plant in their virtual room or collection.<br>**Preconditions:** Player has completed the last question in the quiz without losing all hearts.<br>**Main Flow:** After every hour, the player should see the number of hearts increase by one on the main screen until it reaches 5 without any additional input.<br>**Postconditions:** The updated number of hearts is visible on the main screen.<br>**Alternative Flows:**<br>**Valid:** Hearts stop increasing as player already has 5 or more hearts.<br>**Invalid:**<br>(1) Hearts increasing at longer or shorter intervals than on hour.<br>(2) Hearts not increasing even when the number is below 5.<br>(3) Hearts still increasing even when the number is 5+.<br>(4) Timer not showing next to the hearts.<br>**Frequency of occurrence:** Every hour, contributing to the player's sense of progression. |
| UI Design | **Key UI components:**<br>Main game screen<br>Hearts Icon<br>Timer<br>**Wireframe:** |




**Interaction Flow:**

The hearts increase automatically in the background every hour without any user inputs.

While hearts are replenishing, a timer is visible next to the heart icon showing time remaining before next increase.

The updated number of hearts is reflected in real-time on the main screen.

**UI Behaviour:**

The hearts display should smoothly update without causing disruptions to the player's interaction with the game.

The timer should show the ongoing countdown.

The updated number should be visually indicated on the main screen.

**Visual Design Guidelines:**

Styling should align with the game's defined typography and colours.

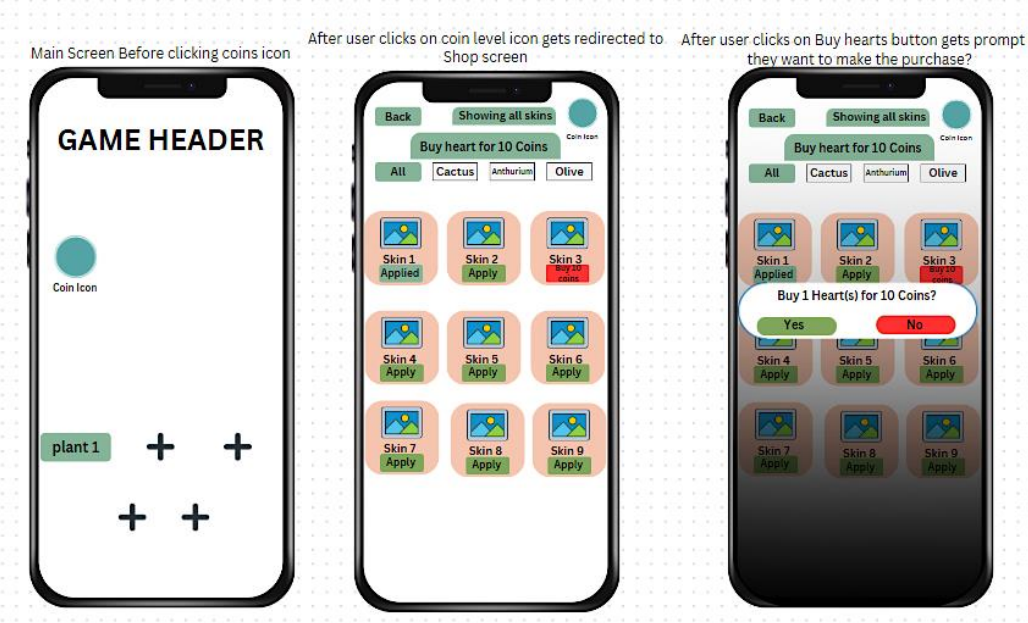
The hearts display should be prominently visible and easily distinguishable on the main screen.

Consider incorporating subtle animations or effects to highlight the increase in hearts.

Ensure the updated number and countdown timer are both clearly visible and does not clutter the main screen.

| US40-V1 | Players can trade coins for hearts at the in-game shop. | | |
|---|---|---|---|
| Description | As a player, I want to have the option to trade earned coins for additional hearts from the shop, so that I can further play some quiz levels for which I need hearts. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US40-V1-AC1.** | I can access the shop from the main game screen. | **Pass** |
| | **US40-V1-AC2.** | The shop displays a buy hearts button. | **Pass** |

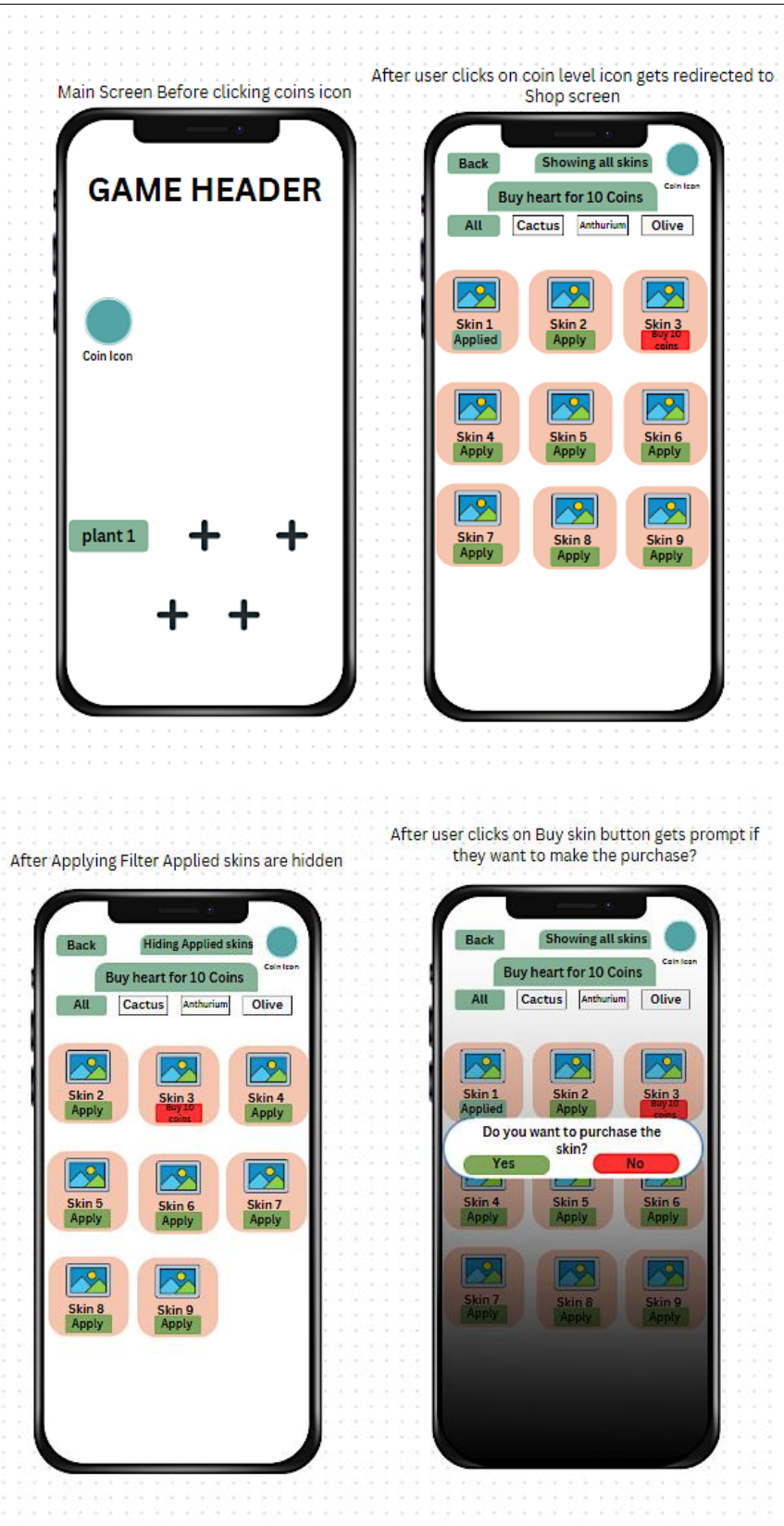
| | | | |
|---|---|---|---|
| | US40-V1-AC3. | It has a visible price tag. | Pass |
| | US40-V1-AC4. | When I click on the 'Buy' button, I am prompted to confirm the purchase. | Pass |
| | US40-V1-AC5. | Upon confirmation, I get 1 heart | Pass |
| | US40-V1-AC6. | The purchased heart is reflected along with the deduction of coins. | Pass |
| | US40-V1-AC7. | If I don't have sufficient in-game currency, I'm unable to buy the heart. | Pass |
| Requirement Use Case | **Use Case:** Purchasing hearts from shop with coins.<br>**Scope:** In-game customisation.<br>**Level:** Shop feature.<br>**Context:**<br>(1) Provided they have earned sufficient coins, players should have the option to buy hearts to be further able to play the quiz if out of hearts.<br>(2) The shop is accessible from the main game screen.<br>**Frequency of occurrence:** Whenever a player decides to explore and purchase hearts and has sufficient coins. | | |
| Design Use Case | **Scope:** In-game customisation.<br>**Level:** Shop feature.<br>**Primary actors:** Player<br>**Description:** Players can access the shop from the main game screen through coin icon to explore shop and purchase hearts. Price tag is displayed on the buy hearts button and users will be prompted before making a purchase. The transaction process involves confirmation, and upon successful purchase, the heart is added, and corresponding coins are deducted.<br>**Dependencies:**<br>(1) The user has successfully installed and launched the game.<br>(2) The game assets related to hearts are correctly integrated into the code base.<br>**Assumptions:** Player has enough coins to purchase a heart.<br>**Main Flow:**<br>(1) Player accesses the shop from the main game screen.<br>(2) The shop displays buy hearts button.<br>(3) Player clicks on the 'Buy' button.<br>(4) Confirmation prompt appears.<br>(5) Player confirms the purchase.<br>(6) Purchased heart gets added and is displayed on main screen.<br>(7) Coins are deducted and reflected on the coins icon on main screen.<br>**Postconditions:** The updated number of hearts and coins are both reflected on the main screen.<br>**Alternative Flows:**<br>**Valid:**<br>(1) Player cannot buy hearts due to insufficient coins. | | |

| | |
|---|---|
| | (1) Player exits shop using “back” button without making purchase.<br>**Invalid:**<br>(1) Player able to buy hearts despite insufficient coins.<br>(2) Player made purchase, but the updated number of hearts is not reflected on main screen.<br>(3) Player made purchase, but the updated number of coins is not reflected on main screen.<br>**Frequency of occurrence:** Whenever a player decides to purchase hearts from shop. |
| UI Design | **Key UI components:**<br>Main game screen.<br>Coin icon.<br>Shop screen.<br>Back button.<br>Buy hearts button.<br>**Wireframe:**<br>  <br><br>**Interaction Flow:**<br>User clicks on a coin icon on the main screen.<br>A shop screen opens on the screen displaying on the top the buy button with price tag for buying hearts.<br>User can buy hearts if he has sufficient coins.<br>Bought heart will be added.<br>Once a heart is bought the price is deducted from the coin icon.<br>User can go back to main menu by pressing back button.<br>**UI Behaviour:**<br>The shop should have a smooth animated effect when appearing on the screen.<br>Confirmation pop-up should appear in centre of screen.<br>**Visual Design Guidelines:**<br>Styling of the shop interface should align with the game's defined typography and colours. |

| | |
|---|---|
| | Items in the shop should be aligned in a way that enhances user friendliness.<br>The price tag should be easily visible for the heart item sold in shop.<br>The heart item should be displayed as a top item in the shop.<br>The hearts display should be prominently visible and easily distinguishable on the main screen.<br>Consider incorporating subtle animations or effects to highlight the increase in hearts.<br>Ensure the updated number and countdown timer are both clearly visible and does not clutter the main screen. |

| US39-V1 | Players can trade coins for plant skins at the in-game shop. | | |
|---|---|---|---|
| Description | As a player, I want to be able to purchase skins for my in-game plants from the shop, so that I can customize the appearance of my plants and enhance my overall gaming experience. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | US39-V1-AC1. | I can access the shop from the main game screen. | Pass |
| | US39-V1-AC2. | The shop displays a variety of available skins for in-game plants. | Pass |
| | US39-V1-AC3. | Each skin has a visible price tag and there is a “buy” button next to each skin. | Pass |
| | US39-V1-AC4. | When I click on the 'Buy' button, I am prompted to confirm the purchase. | Pass |
| | US39-V1-AC5. | Upon confirmation, the selected skin is applied to all instances of the corresponding in-game plant type. | Pass |
| | US39-V1-AC6. | If I don't have sufficient in-game currency, I'm unable to buy the skins. | Pass |
| Requirement Use Case | **Use Case:** Purchasing plant skins from shop with coins.<br>**Scope:** In-game customisation.<br>**Level:** Shop feature.<br>**Context:**<br>(1) Provided they have earned sufficient coins, players should have the option to buy skins that can be applied to in-game plants.<br>(2) The shop is accessible from the main game screen.<br>**Frequency of occurrence:** Whenever a player decides to explore and purchase a plant skin and has sufficient coins. | | |
| Design Use Case | **Scope:** In-game customisation.<br>**Level:** Shop feature.<br>**Primary actors:** Player<br>**Description:** Players can access the shop from the main game screen through coin icon to explore and purchase different skins for their in-game plants. Each skin comes with a visible price tag, and players can view the appearance before making a purchase. The transaction process involves confirmation, and upon successful purchase, the selected skin is applied to the corresponding plant and added as applied. | | |

| | |
|---|---|
| | **Dependencies:**<br>(1) The user has successfully installed and launched the game.<br>(2) The game assets related to hearts are correctly integrated into the code base.<br>**Assumptions:**<br>(1) Player has enough coins to purchase skins<br>(2) Player has at least one instance of the plant type they are purchasing skins for.<br>**Main Flow:**<br>(1) Player accesses the shop from the main game screen.<br>(2) The shop displays available skins with visible price tags<br>(3) Player selects a specific skin to view a detailed preview.<br>(4) A 'Buy' button is available next to each skin.<br>(5) Player clicks on the 'Purchase' button.<br>(6) Confirmation prompt appears.<br>(7) Player confirms the purchase.<br>(8) Purchased skin is added to the players inventory and applied to in-game plant instances.<br>**Postconditions:** The player's in-game plant (all instances of the same plant type for which the skin is purchased) now has the purchased skin applied. The skin item in the shop also displays the “apply” option for the skin just purchased.<br>**Alternative Flows:**<br>**Valid:**<br>(1) Player cannot buy skins due to insufficient coins.<br>(2) Player exits shop using “back” button without making purchase.<br>(2) The skin is not applied as player do not yet have an instance of the compatible plant type.<br>**Invalid:**<br>(1) Player able to buy skins despite insufficient coins.<br>(2) Player made purchase, but the number of coins is not updated to reflect the purchase.<br>(3) Player made purchase but the new skin cannot be applied even though player has an instance of the compatible plant.<br>**Frequency of occurrence:** Whenever a player decides to purchase skins from shop. |
| UI Design | **Key UI components:**<br>Main game screen<br>Coin icon<br>Shop screen<br>Back button<br>Plants named horizontal scrollbar.<br>Plant skins items displayed in shop.<br>Buy/apply button.<br>**Wireframe:** |




**Interaction Flow:**

User clicks on a coin icon on the main screen.

A new shop screen opens on the screen displaying all the plants as a horizontal scrollbar and their skins available to purchase along with hide/ show all skins filter with back button and available coins.

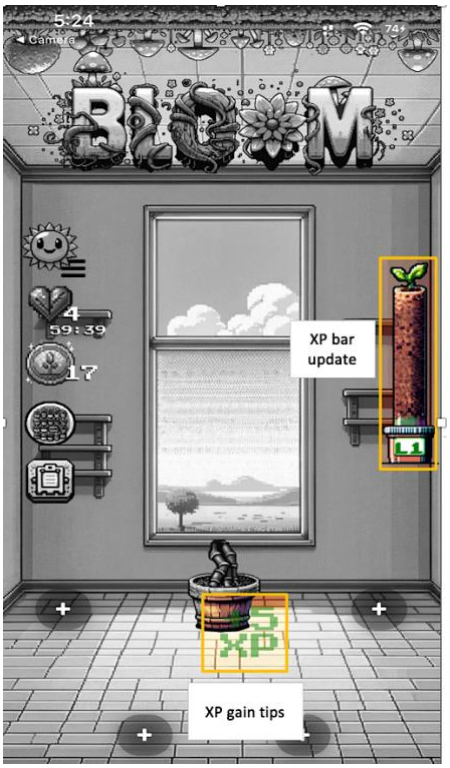
| | |
|---|---|
| | User can buy hearts if they have sufficient coins.<br>Selected skin will be added to the plant.<br>Once a skin is bought user can toggle between available skins of that plant.<br>User can also view different skins available for different plants.<br>User can click on apply button to apply a skin.<br>User can go back to main menu by pressing back button.<br>**UI Behaviour:**<br>The skin items in shop should have a smooth animated effect when appearing on the screen.<br>The player should be able to view all available skins by scrolling left and right between a;; in-game plant types in the form of a scrollbar.<br>**Visual Design Guidelines:**<br>Styling of the shop interface should align with the game's defined typography and colours.<br>All skins should show as image icons with their names labelled next to the icon.<br>The font size and style of skin name labels should ensure it is legible but not consuming too much space.<br>The size of each skin icon should be enough for the player to recognize the skin.<br>Each skin icon should be of a similar size.<br>The size of each image icon should be sufficient for the user to select effortlessly.<br>Each option should have an icon with an appropriate image that matches the style of the game.<br>There should be enough spacing between each option for the player to choose and select. |

| US44-V1 | Receive plant care tips from the Oracle. | | |
|---|---|---|---|
| Description | As a player, I want the Oracle to provide tips on plant care, so that I can learn new information and apply it to enhance my gameplay experience. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US44-V1-AC1.** | The Oracle's tips should cover various aspects of plant care, including watering, sunlight, soil, fertilizing, and more. | **Pass** |
| | **US44-V1-AC2.** | Tips should be relevant to the plants and levels currently accessible in the game. | **Pass** |
| | **US44-V1-AC3.** | Tips should be presented in an engaging and interactive manner, possibly through dialogues or pop-ups. | **Pass** |
| | **US44-V1-AC4.** | Tips should be easily accessible proactively provided by the Oracle. | **Pass** |
| | **US44-V1-AC5.** | The UI should be consistent with the game's overall design, enhancing the visual experience. | **Pass** |
| | **US44-V1-AC6.** | Tips should be progressively disclosed based on the player's level and progress in the game, ensuring relevance and avoiding information overload. | **Pass** |

| | |
|---|---|
| Requirement Use Case | **Use Case:** Learning plant care knowledge through Oracle's tips.<br>**Scope:** Standard gameplay<br>**Level:** Core game features.<br>**Context:**<br>(1) The player encounters the Oracle character in the game.<br>(2) The Oracle offers tips on various aspects of plant care, tailored to the player's current level and game progress.<br>(3) The player reads and can interact with the tips, applying the learned concepts in gameplay.<br>**Frequency of occurrence:** Regularly, depending on the player's interaction and progression within the game. |
| Design Use Case | **Scope:** Standard gameplay<br>**Level:** Core game features.<br>**Primary Actor:** Player<br>**Preconditions:**<br>(1) The player has reached a level in the game where plant care tips are relevant and helpful.<br>(2) The Oracle is functionally integrated into the game, capable of providing educational content on plant care.<br>**Main Flow:**<br>(1) The player interacts with the Oracle character to receive tips.<br>(2) The Oracle presents tips on plant care in an engaging format.<br>(3) The player reads and interacts with the tips.<br>(4) Successful application of the tips in gameplay leads to rewards or game progression.<br>(5) Learning from the tips may unlock new levels, rewards, or achievements within the game.<br>**Alternative Flows:**<br>**Valid:** At any point, the player can choose to skip or exit the tips.<br>**Invalid:**<br>(1) The trivia can relate to another plant.<br>(2) The dialog box can display information in improper way.<br>(3) The player clicks the "back" to main screen button, but stays at the tips level. |
| UI Design | **Key UI components:**<br>The Oracle.<br>Dialog box or similar UI component for displaying plant care tips.<br>**Wireframe:** |

| | |
|---|---|
| |  |

| US9-V1 | Gaining XPs as player progress through the game. | | |
|---|---|---|---|
| Description | As a player, I want to receive some progressive rewards by playing this game. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US39-V1-AC1.** | An XP bar is visible on the main screen representing the user's experience points level | **Pass** |
| | **US39-V1-AC2.** | Every time XP points are gained; it is reflected in the XP bar. | **Pass** |
| | **US39-V1-AC3.** | The XP bar shows the level user is currently at. | **Pass** |
| Requirement Use Case | **Use Case:** Adding Experience Point (XP) function to the game.<br>**Scope:** Standard gameplay<br>**Level:** Core game features.<br>**Context:**<br>(1) Adding level feature to the game that player levels up with certain amount of XP gain<br>(2) Adding an XP bar on the main screen to show the level of the player.<br>**Frequency of occurrence:** XP bar should record player experience points consistently throughout the game on the main screen. XP are gained when user completes a quiz, waters, or fertilises plants. | | |
| Design Use Case | **Scope:** Standard gameplay<br>**Level:** Core game features.<br>**Primary actors:** Player<br>**Description:** Player gain XP and level up accordingly after completing in-game tasks, including watering / fertilising plants and increasing levels for different plants.<br>**Dependencies:**<br>(1) The user has successfully installed and launched the game. | | |

| | |
|---|---|
| | (2) User completes in-game tasks that increases XP.<br>**Assumptions:** Player has not yet completed all XP levels built into game algorithm.<br>**Preconditions:** Player has started the game by planting plants onto virtual screen.<br>**Main Flow:**<br>(1) Player always sees XP bar visible on the main screen, reflecting the XP they currently have.<br>(2) Player completes in-game tasks that increases XP: watering, fertilising, passing quizzes for various plants.<br>(3) XP indicator accordingly increases on the XP bar.<br>(4) Once a bar has been "filled", player progresses to the next XP level.<br>**Postconditions:** Player gains XP and levels up, visually reflected on the XP bar.<br>**Alternative Flows:**<br>**Valid:** Player does not complete in-game tasks that increase their XP, in which case the XP bar does not change.<br>**Invalid:**<br>(1) Newly gained XP is not reflected on the XP bar.<br>(2) XP bar does not level up when player has filled the bar.<br>(3) XP bar does not reset after levelling up.<br>**Frequency of occurrence:** XP bar should record player experience points consistently throughout the game on the main screen. XP are gained when user completes a quiz, waters, or fertilises plants. |
| UI Design | **Key UI components:**<br>Main game screen<br>XP bar<br>**Wireframe:**<br>  <br> |

| | |
|---|---|
| | **Interaction Flow:**<br>XP bar automatically updates whenever user earns XP.<br>XP bar automatically updates whenever user reaches next level and the progressive bar resets.<br>**UI Behaviour:**<br>The XP bar should be displayed on the side of the main screen, clearly visible to users.<br>Visual Design Guidelines:<br>The XP bar should display the user's current level, with a progressive bar shows the progress until the next level. |

| US10-V1 | Unlock new rooms to increase available place space. | | |
|---|---|---|---|
| Description | As a player, I want to have more plant space as I progress through the game. | | |
| Acceptance Criteria | **Criteria ID** | **Description** | **Test Status** |
| | **US10-V1-AC1.** | When I level up as a player, new virtual rooms are unlocked at levels 2, 5 and 8. | **Pass** |
| | **US10-V1-AC2.** | The new room should be accessible by scrolling left / right from the main screen. | **Pass** |
| | **US39-V1-AC3.** | The new room has five more placeholders to hold more plants. | **Pass** |
| Requirement Use Case | **Use Case:** Unlocking new rooms as player reach higher levels.<br>**Scope:** In-game customisation.<br>**Level:** Core game feature.<br>**Context:**<br>(1) As player processes through the game, they may wish to have more plants on display in the virtual rooms.<br>(2) The game gives players the opportunity to unlock new rooms by earning XP and achieving higher XP levels.<br>**Frequency of occurrence:** Whenever player reaches one of the designated levels programmed to unlock a new room. | | |
| Design Use Case | **Scope:** In-game customisation.<br>**Level:** Core game feature.<br>**Primary actors:** Player<br>**Description:** Once reaching certain pre-set levels, the user can unlock new rooms for additional plant space.<br>**Dependencies:**<br>(1) The user has successfully installed and launched the game.<br>(2) The XP bar functionality is complete.<br>**Assumptions:** Player has completed various in-game tasks to gain XP.<br>**Precondition:** Player is currently at a level where the next level unlocks a new room<br>**Main Flow:** | | |

| | |
|---|---|
| | (1) Player levels up by gaining XP.<br>(2) XP bar resets and shows new level.<br>(3) New room is unlocked in the background.<br>(4) Player scrolls right from the main screen to access new room and new plant spots.<br>(5) Player can switch between available rooms any time by scrolling left and right.<br>**Postconditions:** New room is available to plant more plants, supporting all functionalities as the first background. User able to switch between rooms at any time by scrolling left and right on the screen.<br>**Alternative Flows:**<br>**Valid:** Player levels up, but the level is not one that unlocks a new room.<br>**Invalid:**<br>(1) Player levels up and unlocks new room even when the new level reached should not unlock new rooms.<br>(2) Player levels up to a level that unlocks new room, but new room isn't unlocked.<br>(3) Player successfully unlocks a new room but the placeholders in new room does not function as expected.<br>**Frequency of occurrence:** Whenever player reaches a level that offers a new room as reward. |
| UI Design | **Key UI components:**<br>Main game screen<br>XP bar<br>New room<br>**Wireframe: (snapshot from end-product)**<br>   |

| | |
|---|---|
| | **Interaction Flow:**<br>User gains XP and reaches the next XP level.<br>A new room is automatically made available in the interface.<br>User can access the new room by scrolling right on the main screen.<br>**UI Behaviour:**<br>New room is automatically made available in the background. User can scroll between all available rooms they currently have.<br>When scrolling from room to room, main screen components (coins icon, heart icon, mastery level icon, collection icon, XP bar, game header, and Oracle) are statically rendered in the same relative screen position and do not move with scroll movements.<br>**Visual Design Guidelines:**<br>New rooms should align with previous rooms in art style.<br>Plant placeholders should be in reasonable spots in the new room (i.e., floor, shelf, windowsill etc.)<br>The scrolling effect between rooms should be smooth and seemless. |

Additional tests performed:

| Criteria ID | Description | Test Status |
|---|---|---|
| US11-V1-AC2. | When player purchases more hearts in the shop, the number is immediately updated on main screen. | Complete |
| US11-V1-AC4. | The number of hearts replenished at regular intervals, until the user is back with 5 hearts. | Complete |
| US50-V1-AC2. | The advice contains the instructions of how to fill in the page and how to determine the stage of a plant. | Pass |
| US23-V1-AC4. | The water timer of the plant will be reset. | Pass |

### 3.7.6. CRC Cards

The following CRC cards were developed for the functionalities built in Sprint 7.

| Class: XPBarComponent | |
|---|---|
| Responsibilities: | Collaborators: |
| Display the player's current experience points (XP) and level. | levelUpConfig |
| Dynamically update the XP bar image based on the percentage of XP earned towards the next level. | playerConfigContext |
| Calculate the current and next level's XP requirements. | GameText |

| | |
|---|---|
| Visually represent the level progression with changing XP bar images. | |

Relevant user stories: US9-V1, US4-V2

| Class: LevelUpConfig | |
|---|---|
| Responsibilities: | Collaborators: |
| Define the experience points (XP) thresholds required to reach each level in the game. | levelsConfig |
| Serve as a reference for game logic to determine when a player has accumulated enough XP to level up. | |
| Provide a clear progression path by setting specific XP milestones for each level. | |

Relevant user stories: US9-V1, US10-V1

| Class: ShopScreen | |
|---|---|
| Responsibilities: | Collaborators: |
| Provide a shop interface for users to buy skins and hearts using in-game coins. | playerConfigContext, CoinComponent, plantsDataContext |
| Update player's coins and hearts upon successful purchases. | GameText |
| Display a modal for purchase confirmation and handle user responses. | |

Relevant user stories: US40-V1, US39-V1

| Class: backgroundsConfig | |
|---|---|
| Responsibilities: | Collaborators: |
| Provide configuration data for different backgrounds used in the game. | Image Assets |
| Define unique identifiers, names, and associated images for each background. | |
| Specify the level requirements needed to unlock each background. | |
| Detail the plant positions for each background, including their bottom and left offsets. | |

Relevant user stories: US10-V1, US4-V2

| Class: playerConfigContext | |
|---|---|
| Responsibilities: | Collaborators: |
| Maintain and manage the player's state, including hearts, experience points (XP), level, coins, and plant progress. | levelUpConfig |
| Handle the logic for leveling up based on the XP and thresholds. | backgroundsConfig |
| Determine which backgrounds are unlocked based on the player's level. | plantsConfig |

Relevant user stories: US9-V1, US10-V1

| Class: plantsConfig (V2) | |
|---|---|
| Responsibilities: | Collaborators: |
| Maintains the state of the plant, including its health, growth stage, current skin, and other properties | playerConfigContext |
| Simulates the growth of the plant based on time, care actions performed. | UIComponents |
| Handles actions such as watering, fertilizing. | |
| Calculates and updates the plant's health based on care and environmental conditions. | |
| Manages the selection and application of different skins or appearances for the plant. | |
| Keeps track of the plant's progress through different growth stages. | |
| Controls the visual representation of the plant, including animations for growth and interactions. | |
| Responds to user interactions such as clicking or tapping for care actions or viewing details. | |

Relevant user stories: US23-V1, US24-V2, US16-V1, US15-V1, US57-V1

| Class: HeartsDisplay Component | |
|---|---|
| Responsibilities: | Collaborators: |

| | |
|---|---|
| Display the player's current heart count in a visually engaging way. | playerConfigContext |
| Dynamically adjust text and shadow colors based on the number of hearts. | backgroundsConfig |
| Increase the player's hearts count at regular intervals (every 15 minutes). | GameScreen |

Relevant user stories: US11-V1, US8-V1, US40-V1, US57-V1

| Class: PlantLinkComponent | |
|---|---|
| Responsibilities: | Collaborators: |
| Display and manage the modal for a plant's real-life care screen. | Oracle |
| Allow users to add or edit a nickname, stage, and last watered time for a plant. | PlantsDataContext |
| Enable users to add a photo of their plant from the gallery or camera. | |
| Provide care instructions, tips, and a countdown timer for watering. | |
| Integrate with the Oracle component for displaying animations or images. | |

Relevant user stories: US19-V1, US20-V1, US21-V1, US50-V1

### 3.7.7. Team Stand-up

#### 3.7.7.1. Team Stand-up 1

| | |
|---|---|
| Date and Time | 1st Dec 2023, 12:00 noon |
| Location | Microsoft Teams |
| Attendees | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |
| Meeting | **Updates on task-wise progress:**<br>**Claire:** Process documentation complete for Sprints 1-4. Made a list of all user stories on Teams that are pending development.<br>**Alec:** Making progress with installation and maintenance guide.<br>**Marat:** CRC cards near completion, can jump onto user stories.<br>**Favour:** All plant level feature complete.<br>XP bar in progress.<br>Implementing new background feature, still need to make sure plants stay in the same place after changing background.<br>Push notification feature aborted (too difficult to implement).<br>**Udit:** Building the shop component, making progress but incomplete. Will proceed to user stories soon after. |

| | |
|---|---|
| | **Ivan:** Plant link function complete including timer function. Ready to jump onto user stories. |
| Task allocations | **Everyone** on documentation to claim a section of user stories from Claire's user story list (communicate via WhatsApp).<br>**Favour, Ivan, and Udit** to continue with game feature development and to join documentation as soon as all features are complete.<br>**Claire** to continue with process documentation. |

### 3.7.7.2. Team Stand-up 2

| | |
|---|---|
| Date and Time | 3rd Dec 2023, 12 noon |
| Location | Microsoft Teams |
| Attendees | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |
| Meeting | **Updates on task-wise progress:**<br>**Eddie:** Completed user stories for main screen.<br>**Udit:** Completed user stories for mastery levels.<br>**Marat:** CRC cards complete.<br>**Alec:** Structure set for installation guide.<br>**Ivan:** Completed user stories for quiz.<br>**Udit:** Shop feature on track to completion before customer meeting.<br>**Favour:** Game development complete except for minor bugs. Ready to start user stories. |
| Task allocations | **Favour** to fix remaining bugs in the code.<br>**Marat** to move on to user stories.<br>**Everyone** else to continue with assigned documentation tasks. |

## 3.7.8. Exception Handling

| Exception | Actions | Outcome |
|---|---|---|
| Carried over from Sprint 6 | | |
| Change of team structure and roles | **Sprint 7 actions:**<br>Four members are now focusing exclusively on documentation.<br>**Sprint 6 actions:**<br>Everyone agreed to only keep 2-3 people on the coding stream in Sprint 7 while everyone else move to the documentation stream.<br>**Sprint 5 actions:**<br>Previously Favour and Claire were co-leading the documentation workstream. Since Favour has moved to be the oversight of coding, Claire is now solely in charge of keeping documentation. In upcoming weeks, especially by the time people wrap up code development work, the team has to split up documentation tasks again in some way given the heaviness of the workload. | Success |
| Falling behind on | **Sprint 7 actions:** | Success |

| | | |
|---|---|---|
| documentation. | With four members collectively working on documentation, the team has made significant progress by Sprint 7, completing most of what is required in the process and product documentation.<br>**Sprint 6 actions:**<br>Progress still behind for product documentation. More people will move to documentation in Sprint 7 where hopefully the problem could be resolved.<br>**Sprint 5 actions:**<br>As Claire was amongst those struggling with CM50260, documentation was put on hold throughout Sprint 5. The process documentation so far contains only meeting minutes and backlog items, with substantial content still awaiting completion. On the product documentation front, user story development is also at an elementary stage, with only a handful of stories elaborated with use cases and test requirements. UI designs have been completely revamped after moving to React Native, hence needs recreation. As the team has officially kicked-off game development, it is crucial that the relevant documentation is in pace with feature delivery. This must be resolved in Sprint 6. | |
| User story development proving to be a difficult task as game feature development is condensed into one week without time intervals to develop use cases. | **Sprint 7 actions:**<br>Significant progress made with four members collectively developing user stories.<br>**Sprint 6 actions:**<br>People from the development stream will move back to documentation in Sprint 7 where they can reverse-engineer what they have coded into user stories and use cases. | Success |

### 3.7.9. Sprint 7 Backlog and Completion Status

| TASK | Relevant User Story | DUE | PRIORITY | ASSIGNED TO | STATUS |
|---|---|---|---|---|---|
| Implementation of player XP/Levels and progress, and unlocking new rooms. | US9-V1<br>US10-V1 | 5 Dec | High | Favour | Complete |
| Plant real-life link and timer showing plant growth. | US23-V1<br>US21-V1 | 1 Dec | High | Ivan | Complete |
| Implementation of “skins” and “hearts” shop. | US40-V1<br>US39-V1 | 5 Dec | Medium | Udit | Complete |
| Implementing heart replenishment feature. | US11-V1<br>US8-V1 | 5 Dec | High | Ivan | |
| General styling and user friendliness additions | | 5 Dec | Medium | Favour | Complete |
| Complete user story development for all features implemented prior to Sprint 7. | | 5 Dec | High | Eddie, Ivan, Udit | Complete |

| | | | | | |
|---|---|---|---|---|---|
| Complete process documentation for Sprints 1 to 6. | | 5 Dec | High | Claire | Complete |
| Begin creating user manual, installation guide, and maintenance guide. | | 5 Dec | High | Alec | Complete |
| Develop CRC cards for all game components. | | 5 Dec | High | Marat | Complete |

### 3.7.10. Individual Retrospectives

As a result of rushing to complete the workload, no individual reflections were completed for Sprint 7. As the project is near the end, everyone stayed well-aware of the exact list of remaining tasks and have chosen to focus on completing delivery.

However, a project retrospective meeting was held on Monday 11th December 2023. In addition, several informal individual conversations between documentation lead and individual members took place over the final week before submission. The outcomes of the project retrospective are presented in the project retrospective section of this document.

## 3.8. Sprint 8: 5th December – 12th December

### 3.8.1. Overview

The processes over Sprint 8 were kept to a minimum as everyone focused on completing submission requirements on all fronts. Documentation was the priority for everyone as all game development work is finished. In the first part of the week, the plan is to complete development of all user stories relating to features developed in Sprint 7 as well as completing Sprint 7 and Sprint 8 process documentation. From Sunday onwards the process and product documentation sections will be merged into one file, chronologically ordered by sprints. Additional responsibilities include creating an Android APK file for app installation, populating the team questionnaire, and completing the user manual and installation guide.

### 3.8.2. Review

Early in the Sprint, Favour successfully created the APK file which was downloaded and tested on Udit's Android phone. The team was also able to complete all user stories by Sunday as planned. For the rest of the week, Claire focused on merging product documentation and process documentation and everyone else assisted Alec with building the user manual and installation guide. A project retrospective meeting took place where the team discussed in depth the achievements and missteps over the project delivery period. Over the Wednesday lab session, the team collectively completed the questionnaire with unanimous a, wrapping up all prescribed deliverables for the project.

### 3.8.3. Summary of Key Events

- Success with creating Android APK file for mobile download.
- Completion of process documentation and product documentation.
- Completion of installation guide, maintenance guide, and user manual.
- In-depth project retrospective session.
- Completion of team questionnaire

### 3.8.4. Sprint 8 Preparation: Customer Meeting and Planning Discussion

| | |
|---|---|
| **Date** | 5th Dec 2023 |
| **Location** | Campus – Chancellor Building 4.7 |
| **Attendees** | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |
| **Customer Meeting** | **Presentations:**<br>• **XPs:** Levelling system with XP bar implemented recording progress each time player gains XPs. Player can unlock new rooms when they level up on the XP bar, currently new rooms are unlocked at levels 2, 5, and 8.<br>• **Quiz completion:** Player awarded with coins and XPs. Bonus coins are awarded if first time completing the level.<br>• **Plant mastery level:** Indicator pop-up in home screen showing mastery level for each specific plant.<br>• **Shop:** Complete, although currently only sells one skin for the cactus.<br>• **Link to real life plant:** Tips given to determine what stage your plant is based on size (dependent on the type of plant), as well as the steps on how to make link – take photo, |

| | |
|---|---|
| | set name, last watered. "How to" button available at all times for instructions. After returning to main screen, time remaining until next watering is shown below the plant together with uploaded photo.<br>• **Fertilise and water:** Gives XP to player every time an in-game plant is watered/fertilised. Timer refreshes every 12 hours.<br>• **Hearts:** Time shown when they heart refresh. Player can also buy extra heart in shop for 10 coins.<br>**Customer comments:**<br>• Customer seems content with all features. No comments made other than to keep playing the game and catch major bugs. |
| Planning Discussion | **Additions/changes to product design based on customer meeting:**<br>No additions arise from the customer meeting.<br>**Analysis / reflections of current standpoint:**<br>With just one week before submission, it is critical that everyone does their best to ensure that documentation is completed on time. To simultaneously wrap up several documents, efficient division of tasks and strictly adhering to intermittent due dates is especially crucial as some people's tasks will be passed downstream to others.<br>**Determining Sprint priorities:**<br>The immediate priority for Sprint 8 is to complete what's left of the product document components (user stories) for features developed in Sprint 7, which the documentation lead will merge them with the sprint-wise process documentation. In the final days before submission, everyone except for the documentation lead will jump onto writing the installation guide, maintenance guide, and user manual. Meanwhile, Favour is asked to check over the user story list and add any new stories of versions built in Sprint 7. This must be completed by Wednesday evening to enable allocations of new stories to individual members. Throughout the week, Favour and Ivan will also be running tests and fixing last-minute bugs introduced by the various features added in Sprint 7. Packing an Android APK file for app installation file is set to high priority after all major bugs are fixed.<br>**The Sprint 8 backlog can be found in** Section 3.8.11 |

### 3.8.5. User Stories, Requirement Use Case, Design Use Case, UI Design

Only one user story is relevant to Sprint 8: US99-V1 Running game on device.

| US99-V1 | Running game on device | |
|---|---|---|
| Description | As a player, I want to be able to download and run the game on a mobile phone. | |
| Acceptance Criteria | US99-AC1. | I can download the game as an application on a phone. |
| | US99-AC2. | I can successfully run the game once downloaded. |
| Requirement Use Case | **Use Case:** Downloading the game as an application on mobile phone and run it successfully.<br>**Scope:** N/A<br>**Level:** Game installation.<br>**Context:**<br>(1) After complete game development, the game should be accessible to users who wish to play it. | |

| | |
|---|---|
| | (2) The game should be accessible as a downloadable file which the user can install onto their phones as an application.<br>(3) The application should be able to run just like other apps, accessible to the player at any time.<br>**Frequency of occurrence:** Every time a user wishes to play the game on a new mobile device.<br>**Open issues:** Only tested on one Android device, hence unsure if it runs as expected for other Android versions. |
| **Design Use Case** | **Scope:** Android mobile devices.<br>**Level:** game set-up.<br>**Primary actors:** Player, Android mobile device<br>**Description:** Player can download an APK file onto Android device to install and run the game as an Android application.<br>**Dependencies:** N/A<br>**Assumptions:** N/A<br>**Preconditions:**<br>(1) The user has a compatible Android mobile device.<br>(2) The user has stable internet connection.<br>(3) The user needs to enable installation of applications of unknown sources. This could be done by navigating to “device settings”, “securities”, and enable the relevant option.<br>**Main Flow:**<br>(1) Scan QR code or access link to start download.<br>(2) The user navigates to the directory where the APK file is saved (usually in the “Downloads” folder).<br>(3) The user clicks on the APK file downloaded, then consent to the confirmation popup seeing user consent to install.<br>(4) After installation is complete, the user can access and run the app from the home screen.<br>**Post conditions:** The game application is successfully downloaded, installed, and running on the user’s mobile device.<br>**Alternative Flows:**<br>(1) The APK file cannot be downloaded or installed. (invalid)<br>(2) The installed app cannot run seamlessly on device. (invalid)<br>**Frequency of occurrence:** Every time a user wishes to play the game on a new mobile device.<br>**Open issues:** Only tested on two Android devices, hence unsure if it runs as expected for other Android versions. |
| **UI design** | **Main menu icon design for game application:**<br>  |

### 3.8.6. Tests

| ID | Acceptance Criteria | Test Outcome | Notes |
|---|---|---|---|
| US99-AC1 | I can download the game as an application on a phone. | Pass | Tested on Android system version OnePlus 8T and Android 13. |
| US99-AC2 | I can successfully run the game once downloaded. | Pass | Tested on Android system version OnePlus 8T and Android 13. |

### 3.8.7. CRC Cards

N/A – No CRC cards are deemed relevant to this sprint.

### 3.8.8. Team Stand-up Minutes

Only one stand-up took place over Sprint 8.

| | |
|---|---|
| Date and Time | 8th Dec 2023, 12:00 noon |
| Location | Microsoft Teams |
| Attendees | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |
| Meeting | **Updates on task-wise progress:**<br>**Favour:** Android APK file complete, all bugs fixed; App downloaded and tested on Udit's Android phone; checked through user story list and added those from Sprint 7.<br>**Claire:** Completion of Sprint 7 process documentation and most of Sprint 8.<br>**Everyone:** Progress on track for on user story development. |
| Task Allocations | **Claire** to merge product documentation sections into Sprint 8.<br>**Everyone else (except for Alec)** to complete user stories by Sunday, and to help Alec with user manual and maintenance guide from Monday onwards.<br>**Alec** to continue writing installation guide, maintenance guide, and user manual.<br>**Everyone** to communicate progress and individual capacity in WhatsApp group. |

### 3.8.9. Additional Sprint Retrospective Discussion

The team gathered for a deep dive into the process design, team dynamics, achievements and oversights over the whole project duration. The outcome of this session is recorded in Section 2 Project Retrospective.

| | |
|---|---|
| Date and Time | 11th Dec 2023, 12:00 noon |
| Location | Microsoft Teams |
| Attendees | Alec, Claire, Favour, Eddie, Ivan, Marat, Udit |
| Meeting | **Tops covered in discussion:**<br>• Review of team processes. |

| | |
|---|---|
| | • Review of project trajectory over eight Sprints.<br>• What was done well.<br>• Major mistakes made and what corrections would have been made in retrospect.<br>• Individual contributions.<br>• Evaluation of team dynamics |

### 3.8.10. Exception Handling

N/A – No exceptions were raised in Sprint 8.

### 3.8.11. Sprint 8 Backlog and Completion Status

| TASK | RELEVANT USER STORIES | DUE | PRIORITY | ASSIGNED TO | STATUS |
|---|---|---|---|---|---|
| Create Android APK file and arrange app installation test with Udit (only member with Android phone) | US99-V1 | 11 Dec | High | Favour | Complete |
| Check user story list and update with new features introduced in Sprint 7. | | 6 Dec | High | Favour | Complete |
| Complete Sprint process documentation for Sprints 7 and 8 | | 11 Dec | High | Claire | Complete |
| Complete user story development for all new features introduced in Sprint 7. | | 11 Dec | High | Udit, Ivan, Favour, Marat, Eddie | Complete |
| Merge all user stories and CRC cards into sprint-wise process documentation. | | 14 Dec | High | Claire | Complete |
| Complete Installation guide. | | 14 Dec | High | Alec, Udit, Ivan, Favour, Marat, Eddie | Complete |
| Complete user manual. | | 14 Dec | High | Alec, Udit, Ivan, Favour, Marat, Eddie | Complete |
| Complete maintenance guide. | | 14 Dec | High | Alec, Udit, Ivan, Favour, Marat, Eddie | Complete |

### 3.8.12. Individual Retrospectives

As a result of heavy workload and rushing for time, no individual retrospectives were completed for Sprint 8. As the project is near the end, everyone stayed well-aware of the exact list of remaining tasks and have chosen to focus on completing delivery.

However, a project retrospective meeting was held on Monday 11th December 2023. In addition, several informal individual conversations between documentation lead and individual members took place over the final week before submission. The outcomes of the project retrospective are presented in the project retrospective section of this document.

# 4. Conclusion

This document has presented detailed records of what has clearly been an eager yet chaotic attempt at agile software development. Although a satisfactory product was to the customer, the steps along the eight-week journey to the end result was far from adequate. Implementing agile processes and practices has proved to be the biggest challenge for the team, which in turn led to fiasco in later stages down the timeline. With the prevalence of agile methodologies across the software engineering industry, this first-hand experience at formulating a self-organising team is tremendously meaningful for all seven peers as part of the conversional MSc course. There is little doubt that everyone in the Plum team has gained a deeper appreciation of the value of agile development framework, having self-inflicted the wreck that stemmed from disorganised process implementation.

Despite so, it is nonetheless commendable achievement for the team, most with little prior exposure in the field. All peers actively contributed teamwork with no exceptions, and no frictions occurred between teammates. Several peers have described this project as “the best teamwork experience I’ve had in university”, reflecting the collaborative and supportive environment the team has built. Meanwhile, on the technical side, the final product also appears to meet all initial specifications listed by the customer, which is undoubtedly a milestone for everyone in their learning path to becoming computer science professionals.

To conclude, this is merely a starting point in everyone’s academic and professional life as software engineers. It is anticipated that the achievements, the mistakes, the lessons recorded in this document will be something everyone can take into their future as an invaluable asset, beholding not just the soft and hard skills gained in the process, but also the strength, persistence and diligence everyone has shown.