Monash Permaculture Club Application

Adam J Judd, 29/03/2018

Application Concept

Rationale/Introduction

The Monash Permaculture Club (MP) currently manages Monash Permaculture Garden (MPG) and Monash University Community Farm (MUCF), each acting as community gardens and educational facilities.
The members of MP are responsible for planting, harvesting and watering the produce and permanent horticultural fixtures at their two locations, in additional to performing general maintenance on the locations and – as of Semester 2 2018 – contributing to the "Knowledge Bank" overseen by the club's committee.

The Permaculture Club App should make relevant information readily accessible and help facilitate the club's activities by streamlining group coordination and communication.

Where currently tasks are assigned and delegated via several different streams, the app should be able to consolidate the organisation of duties into a single medium. This will increase awareness of urgent tasks and increase the likelihood of their timely completion.
The Knowledge Bank will be introduced within the app. This will be a repository of information the members can use to more efficiently and safely go about their tasks, which should double as a means of communication between members. Documents will include (but are not limited to):

- Standard Operating Procedures (SOPs)
- Administrative Procedures (APs)
- Club Induction Hand-out
- Club Constitution
- Meeting Minutes
- Financial Audits
- Infrastructure Issue Reports (IIRs)
- Plant Issue Reports (PIRs)
- Plant Profiles (PPs)

The ultimate objective of the Permaculture Club App is to create a virtual environment from which a user can easily glean the tasks that need doing, how those tasks should be performed, and the context surrounding the task (locality, assets involved, cause of the task).
This will be achieved by a three-factor model, incorporating the following overarching elements:

- Virtual Garden (both MPG and MUCF)
- Task Scheduling Service
- Knowledge Bank (by Member Contribution)

These elements will link together to form a cohesive whole, enabling users to gain a clear understanding of farm operations without the input of a more experienced member.

Target Audience

The target audience includes all ordinary members of MP, committee members, prospective members, previous members and potentially non-member workers at MUCF and MPG. This audience will include a range of undergraduate and postgraduate Monash University students, including mature age students. These students are undertaking courses ranging from Computer Science, to Marketing, to Biomedical Sciences.
With the incorporation of non-student club participants, the age demographic extends from late adolescence (roughly 18 years) to at least 60 years. Extrapolating from the Club's 2018 member-base, the demographic of users will span both sexes in roughly equal proportion.

The members of MP have, for the most part, an extensive involvement in site operation and have horticultural expertise either learned by experience or through tertiary study in a related field. Therefore, the app can assume most knowledge relating to practical tasks on the ground.

MP has consistently maintained a group of approximately 20 regularly participating members and non-members each year since the establishment of MUCF in 2011 (as defined by people who attend working bees at the two sites and social events at Wholefoods, Monash Clayton).
People in this group typically: have face-to-face contact with one-another at least once a week and are invested in both the social and agricultural aspects of MP. They are likely to be the frequent users of the Permaculture Club App and are thus the most critical target audience.
Other workers are likely to install and use the app, however it is only this core group that are likely to make frequent use of it.

The design of the user interface should therefore heed the following invariants:

- The varied age and background and thus technological proficiency of users
- The mutually shared passion for and knowledge of agriculture and horticulture
- The intermittency of user interaction with the app (especially the Knowledge Bank)

The problems faced by workers on-site can be divided into the following categories:

- **Informational –**

  the sharing of educational and administrative information between Club members.

- **Organisational –**

  the division, understanding and fulfillment of duties that physically sustain the sites.

- **Bureaucratic –**

  interactions with staff and the general bureaucratic environment at Monash.

Each of the above categories contain several issues. At the time of writing, the primary ones are as follows:

Informational	Organisational	Bureaucratic
Lack of horticultural and agricultural education	No method of identifying plants	The presence of Health and Safety issues
Lack of education on Health and Safety issues	Inconsistent watering and water unavailability	Members' unfamiliarity with site stakeholders at Monash
Inconsistent understanding of Permaculture practices	Inconsistent approach to planting and harvesting	Lack of communication channels to staff
Members lack understanding of club administration	No clear method of issue reporting and resolution	Stakeholder uncertainty regarding site progression
	Lack of clear member communication streams	

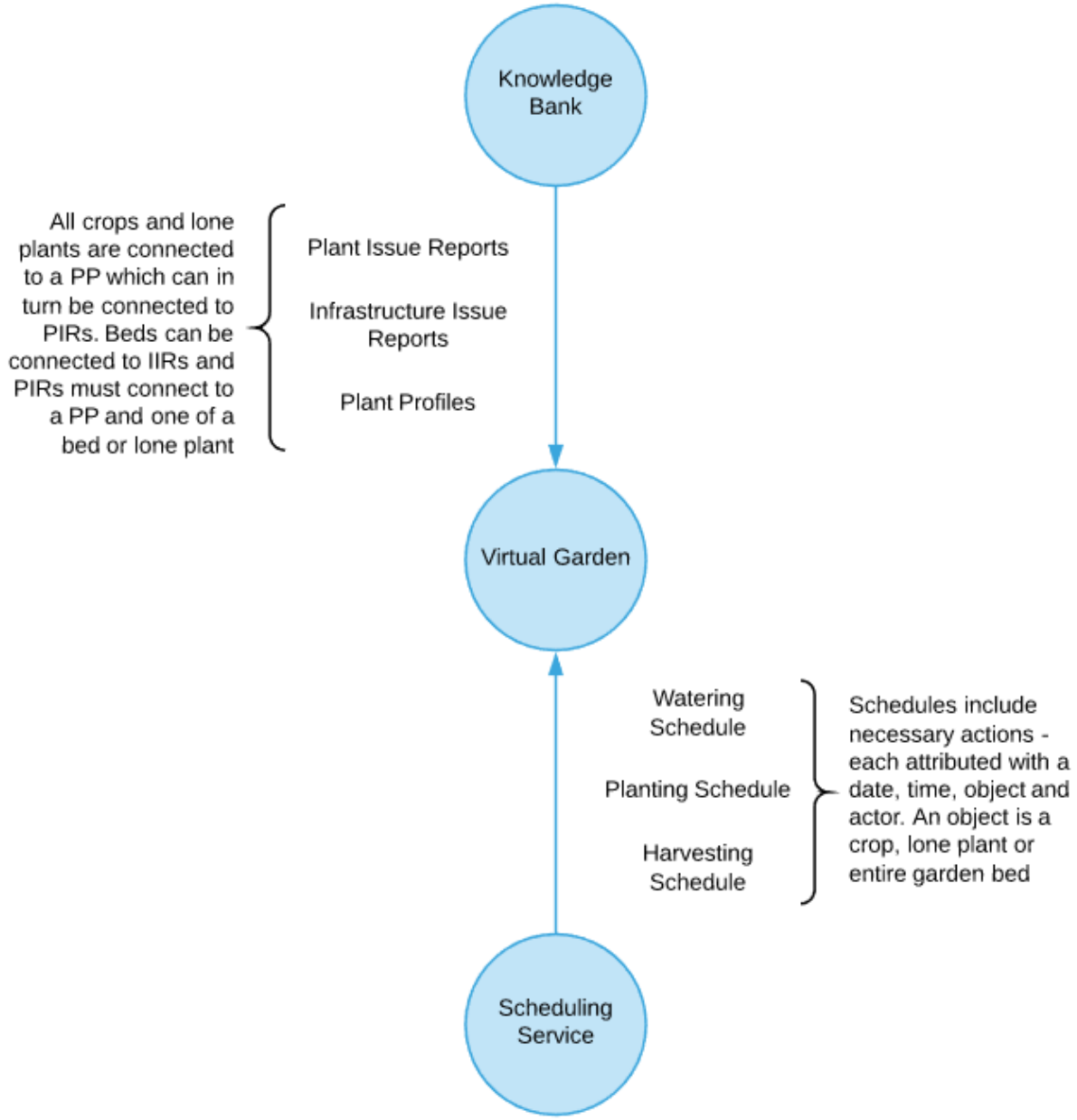
The above are all recurring problems that prevent member cohesion and complication free site management. At present, a handful of MP members perform the decision-making processes that should incorporate the wider community of site-goers. Additionally, many of these decisions should realistically take days or weeks of group deliberation but are currently compressed into a single working bee (two hours).
This is a situation arising from a lack of both communication and member/non-member participation. While site management still occurs, this method does not draw on enough opinion and data, so techniques are not applied to maximise crop growth nor efficient farm maintenance. The alternative is to create a system that coordinates members in a simplistic and intuitive way to promote maximum user engagement and input.

The Permaculture Club App will make use of the extensive knowledge of its user base to avoid sourcing data directly from the net and will incorporate an interface that is minimalistic and navigable, while clearly denoting elements as plants, beds, trees and other fixtures in the Virtual Garden.

Using the three overarching application elements, all Informational and Organisational problems can be mitigated, and ideally resolved permanently. The app presenting a permanent solution will be dependent on a high proportion of site-goers utilising it and those users opting for the app rather than more manual communication streams such as Facebook or Email.
Bureaucratic problems will be more difficult to alleviate, as they involve people and assets external to the Club. Nonetheless, the app may substantially increase the ease of member-bureaucracy interaction through helpful guides uploaded to the Knowledge Bank (Aps and Meeting Minutes). This may reduce the end-to-end (conflict or issue detection to resolution) time of an interaction between MP and Monash's bureaucratic environment.

Application Functionality

As mentioned, the app can be divided into three core elements. A high-level visualisation of their connection can be garnered from the following diagram:

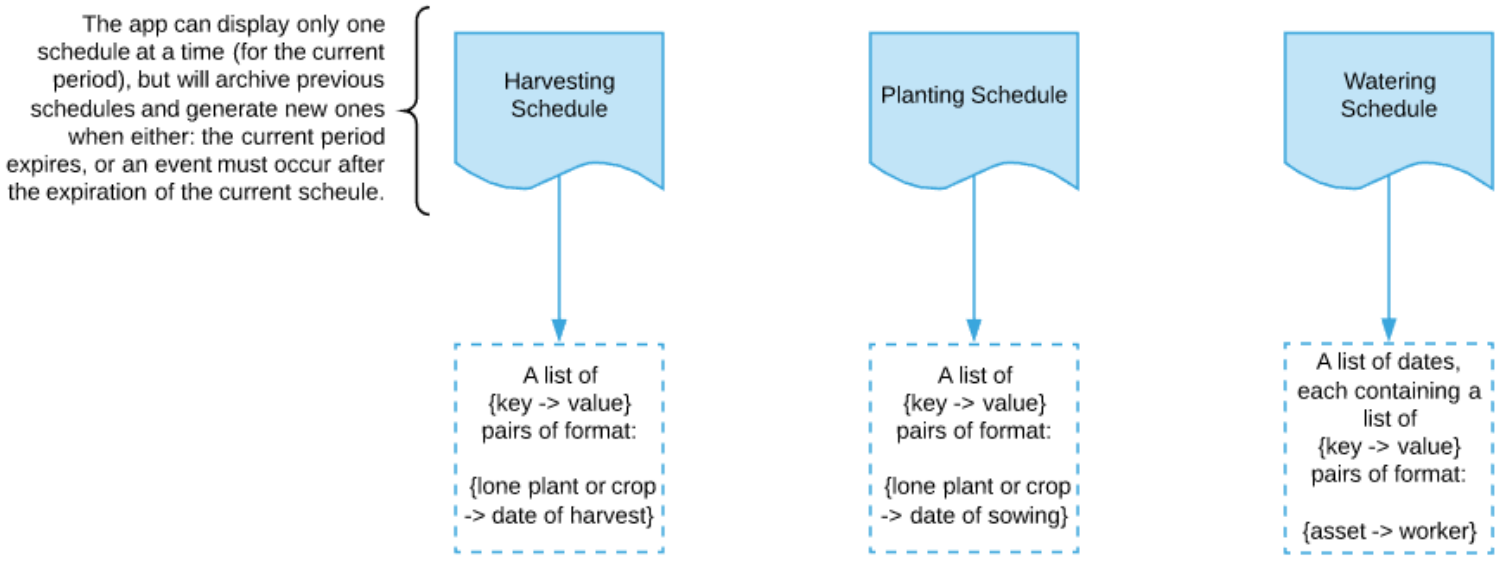


Below is a set of three tables outlining the core functionality of each of the three overarching app elements, in addition to general administrative functionality.

Virtual Garden Function	Functionality description
Add Asset	A broad function – an asset is any physical object at one of the sites that is not ephemeral. A user should be able to expect that if they visit the appropriate site, they will find the asset at that site most of the time. Assets include crops, beds and lone plants; however, the breadth of this functionality is in its potential – many dozens of asset types could be added to the application. Some of the more prominent candidates include compost bays, sheds, shelters, tap locations, water tanks, underground infrastructure and growing lattices. For the three asset types that will exist at the time of deployment, each supports links to documents from the Knowledge Bank and some force the user to establish these connections on creation. When the user is accessing the virtual garden and taps the "Add Asset" button, the asset type can be selected and then the asset is either dragged and dropped onto the Virtual Garden from an asset selection pane then snapped to the grid, or in the case of beds, the shape can be drawn onto the screen by the user. For assets dropped inside a bed, that data is recorded. Each plant-based asset will be linked to a PP (and cannot exist unlinked). This will provide a connection to data such as recommended watering frequency (as time of day and quantity per week), expected time until germination (in days) and expected time until harvest (in weeks), among others. When schedules are devised, this information can be consulted to increase accuracy.
Hover Asset	A user can tap on an asset to bring up the near-future schedules, the primary photo and a brief description of that asset. This creates and overlays a screen for the asset selected onto the virtual garden. If the asset being targeted is within a bed, then the bed is first selected, and then assets within it can be selected by tapping on them in the expanded view of the bed in its screen. This means that, at any given time, there may be 3 activities layered in an order like [bed-internal crop/lone plant > bed > site virtual garden] while "hovering" on an asset.
Asset Specifications	When a user is hovering on an asset, the user can tap again on the asset (the expanded image) to navigate to another screen that will give a comprehensive overview of that asset. This includes planting history, links to all PPs, PIRs and IIRs, all schedules that exist for the asset and a full outline of all items and quantities of those items that exist as part of the asset (especially in the case of a bed, but also for crops).
Remove Asset	Assets that are more permanent (lone plants and beds) should be removed manually. If an asset is defined in its PP to have a specific end date for its lifecycle (i.e. date of sowing + number of days in lifecycle up to harvest) then an in-app notification will appear for each user on the resulting day asking any user to confirm the harvest. This notification will also appear on each consecutive day unless the asset is manually removed, or a user confirms its harvest. Any user can perform the confirmation.

Knowledge Bank Function	Functionality description
Adding/Creating Documents	Documents are added to the knowledge bank repository to be accessed by members. The document type is specified, and either an upload is performed, or an empty form is presented to the user for them to fill. The upload file format or form will change depending on the document type. This is how users share important content with one another, eventually creating a database of most conceivable Club documents and permaculture information. Some specific document types will require administrator privileges to upload (all excepting PPs, PIRs and IIRs). Users can upload photos on their phones to documents they create while invoking this function.
Accessing Documents	Any user can access the documents uploaded to the knowledge bank by other users. All documents are fully visible to **everyone**. Documents will be linked in logical ways to other documents and to the Virtual Garden. This includes PPs, PIRs, IIRs and SOPs. The former 3 are linked to crops, beds or lone plants, and the latter can be linked to any fixture in the virtual garden as appropriate. Therefore, users can easily navigate between a garden bed and the IIRs that have been filed for it, or a PP and the PIRs filed.
Removing Documents	Document removal is another important feature that is accessible to **only** administrator users and the user that created/uploaded a document. This ensures protection from malicious document deletion by non-authorised users.
Link Document to Another Asset	A document in the knowledge bank can be connected to other documents and to objects in the Virtual Garden. These links are forced implicitly (e.g. when a crop is created an it must be linked to a PP) or manually configured (e.g. when a PIR exists and is linked to a specific bed, then occurs again in another bed – a new connection must be established).

Some background on the internal structure of Schedules generated by the Scheduling Services:



Scheduling Service Function	Functionality description
Generate Schedule	When an asset is created, if the system has the information required (by means of data scraped from PPs) to generate an item in a schedule (such as a harvesting date) it will do so. Otherwise, the information must be added, and the user will be alerted that the asset cannot be incorporated into a schedule. Assets in beds are grouped together for the purposes of watering schedules, thus all assets in the current period's watering schedule will be beds or assets outside of beds. Only these schedules require users to be linked to each individual duty. Assets appear individually for planting and harvesting schedules. Schedule periods are initially planned as 1 month for watering and 1 season (or 3 months) for planting and harvesting schedules.
Collect Availability	When a watering schedule exists for the current period and there are assets that require watering, the Scheduling Services will generate a local availability sheet that users will be prompted to fill in upon log-in (and will remain as a notification in-app until the schedule is filled or 'resolved'). This form will collect availabilities from the users until all necessary duties for the period are attributed to a worker. This process also occurs when a watering schedule is due for expiry. 7 days before the period on the schedule ends, availability forms will be generated and distributed to the users so that the schedule can be resolved. Local sheets are uploaded whenever a user has made a change and is connecter to the internet. Scheduling Services collates the sheets and uses the entries to fill in the 'master sheet' once per day. This process occurs on each user's device. The Master Sheet generated for a given day will then become the local sheet distributed the following day, if there are still tasks unassigned. The priority with which a worker is assigned to a duty they are available for is negatively proportionate to the number of duties they are already assigned to in the master sheet (i.e. always attempt to distribute duties evenly between users). If a user does not connect to the internet after inputting their availabilities and before the sync time, their entries cannot be used to resolve the schedule. They also may not see the most recent master sheet reflected on their phone's local sheet.
Generate User Notification	At a designated time on each day, a Scheduling Service process will identify all users who are required to perform watering on that day, in addition to identifying plant-based assets that require planting or harvesting. In the case of the former, the appropriate user will receive a notification (out-of-app), and for the latter **all** users will receive a notification.
View Scheduled Tasks	A user can log in and view the master sheet in a clear and readable format to visualise who is performing duties. That user's local sheet will reflect the master sheet if the app has synchronised via the internet. This functionality will also enable the user to filter assigned tasks by the worker – meaning that they can see the days in the current period that they need to perform watering. Filtering by plant (PP) is possible for planting and harvesting schedules.

Administrative Function	Functionality description
Account Setup	Users will need to set up an account on the system through which they interact with the app. This account can be an ordinary member or Committee Member account. One account is designated as the Committee President (initially designated in development as the original account), and that account can designate other accounts as Committee positions and revert those accounts to ordinary members. When the President account designates another account as President, the former reverts to an Ordinary member. All user data is stored in the database, including a 'User ID', a general identifier.
Synchronisation	App functionality is both online and offline. Where a component of the application is modified, this change will need to sync between devices and so modifications (including editing and deleting documents, and moving, adding and removing assets in the Virtual Garden) will require internet connectivity. Many of these actions can be performed by only one user at a time so the app needs to moderate users in such a way that they can be barred from specific online actions temporarily.
Module Completion	The app should record a list of all SOPs and APs each user has read, in addition to the Club Constitution and Induction Hand-out. This is recorded when the user checks a button at the bottom of each document affirming that they have read it in full. This can be used by the Committee for administrative purposes where necessary and to assert the expertise and awareness of MP members to Monash staff.
User Viewing	A user can navigate to the list of all users and select one to bring up their profile, comprised of the data they input in their registration including a photo. From here the President account can change the Committee status of other accounts.

Innovation via Mobile Technology

Portability and Native Components

Phone portability is the most significant attribute of a mobile application leveraged by the Permaculture Club App. Portability allows for the app to be used on-site, granting users access to the information required to perform tasks efficiently, safely and effectively when they require it. Workers do not need to be concerned with understanding all aspects of Permacultural Practice or having an in-depth knowledge of the state of either site to visit a site and perform some tasks, yielding far greater member autonomy.

The primary native components that the application utilises are the phone's camera, file upload from the device, internet connectivity and persistent storage. The latter two feature an in-app separation whereby most app functionality is offline due to regular downloads from the knowledge bank to the phone's persistent storage. Functionality that changes the state of the repository, or the Virtual Garden, is only accessible when an internet connection exists. And scheduling availability can be filled offline but will only be submitted when an internet connection exists, before being processed at the end of each day.
The camera will be used to capture photos on-site (or off-site, where appropriate) for attachment to Knowledge Bank documents such as PPs and PIRs. Lastly, file upload from the device provides a method by which a user can transfer a file such as Committee Meeting Minutes into the Knowledge Bank from the device (if it is stored locally). This allows for the bulk of the documents to be written on any other device (i.e. a PC or laptop) and then transferred into the repository from the phone.
These use of these two native technologies allow for timely and more accurate reporting of issues, and an increased probability of information upload occurring since a worker will almost always carry their phone with them to the sites.

Internet Connectivity

Creating a mobile application provides the prospect of online features via the cellular network and WiFi, whereas the current method by which information is communicated is verbal, or when online (Facebook, Email) correspondence is split over many mediums. Internet connectivity can be largely assumed while at the sites since the Eduroam network is available at both and all student MP members have access to it. Connection provides the following features:

- Document submission (including removal and editing) to the Knowledge Bank
  Users can submit documents from their phone, and edit or remove documents that are identified with their User ID. In the case of Committee Member users, removal and editing of documents is always permitted, and all document types can be submitted.
- Asset editing in the Virtual Garden
  Users can add, update the details of, or manually remove any asset in the Virtual Garden. If necessary, the app may automatically generate schedules for that asset.
- Schedule resolution by the Scheduling Services.
  All the various local availability sheets can be collated at the end of each day (assuming an internet connection) and users can each contribute to the master sheet to resolve the schedule.

These three functional areas together grant better group coordination through access to a consolidated repository of all information. Scheduling presents a method of clearly communicating all

site management roles at a given time, reducing worker dissatisfaction resulting from being out of the loop.
Collective contribution to the app is made possible by online services and reduces development workload since – once the app is deployed – all assets, documents and schedules will be either automatically generated or created by one of the app's users.

Touchscreen

The use of touch in the app provides a vessel for simplicity of design and straightforward user interaction. This is especially true of the Virtual Garden, where the use of touch allows a user to very easily zoom and hover on assets to rapidly gain an overview of what is currently being grown and tended at the sites. All assets will be interactable, and the space between assets will be consistently differentiable from the assets themselves to ensure that users immediately understand how to interact with the Virtual Garden.

Asset creation will be akin to drawing beds onto a map (in the case of beds) or drag-and-dropping objects. These are both intuitive mechanisms and are incredibly efficient to perform – courtesy of touch controls.

Platform Technology Considerations

The following table outlines the considerations made for each platform, for functionality that is likely to be implemented differently across operating systems. For example, designating an account as a "Treasurer" from the President account will be done in the same way on both platforms since it is not a function that leverages any device-specific software/hardware elements.

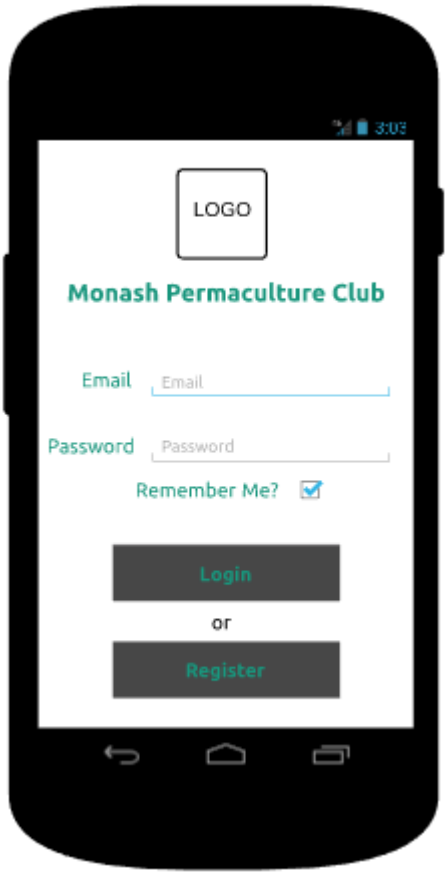
Functionality	Android Implementation	iOS Implementation
Photo Capture	In Android we can rely on intent filters to launch an activity which will return to us a URI for the chosen photo. In this case we will pass the URI for the Phone's primary external storage with the intent. We can use the photo's URI to get the path to it in memory and perform the upload to the database.	Swift provides the class UIImagePickerController. Creatng a 'delegate' for the class and setting the 'sourceType' to photoLibrary will permit the user to select a photo from the library. Once selected, a URI can be retrieved, and the photo uploaded to the database.
Virtual Garden Assets	The app does not support landscape orientation, and therefore all assets will be defined by absolute positions on a base layout (the ground, in physical terms). Due to the presence of the Floating Action Button, a Coordinator Layout will serve as the base layout. The 'ground' layout will be an AbsoluteLayout which allows each child view (assets) to be defined in terms of their absolute position. A subclass of View will be written to accommodate the functionality of assets.	iOS Virtual Garden implementation will be largely like that of Android. The key difference is the omission of the FAB, which is instead accessed from the options menu. Views will operate in the same draggable manner, ultimately, they will be defined in terms of a set of x, y coordinates in order to ensure they remain fixed after editing is complete (i.e. they will not conform to constraints defined relative to other views).
Local Availability Sheets	A basic ConstraintView arranged into a grid with each user who has entered at least one availability appearing as a row, columns representing days. Hovering over a cell displays the beds that need to be watered, which are tooltips which could be implementing using the View.setToolTipText() method. A more complicated tooltip could also be devised by writing a ToolTip class (extending View). This class would be instantiated in the onClick method of an onClickListener attached to asset Views.	The CMPopTipView class provides functionality that will enable tooltips for UIViews (assets) defined. However it is, like with Android, probably more favourable to extend the UIView class and instantiate those Views in order to have greater control over the tooltip's functionality.

Out-Of-App Notifications	The NotificationCompat class provides functionality that supports lock-screen and heads-up notifications. Both types could be implemented by the app. This involves creating a notification channel of a specified "importance" (one of four potential values). The tap action for the notification will direct the user to the activity responsible for handling local availability sheets for watering (or in the case of harvesting or sowing, to the Virtual Garden for the appropriate site).	Using the Apple Push Notification Service, a background process checking for scheduled activities due can activate a Push Notification which will in turn display the message to the user. This notification will be configured so that, like the Android counterpart, the notification directs the user to the appropriate screen within the app.
Knowledge Bank Database Technology	Firebase is a simple and accessible cloud database tool to which the app can upload Documents and Tables (filled with records generated by user interaction with the app). The Firebase SDK will be installed, and its dependency built into the Gradle. Once this is done, the FirebaseDatabase class provides the functionality required to	Firebase must be specifically authorised for iOS. While Firebase appears to be the logical choice for a database technology, iOS compatibility issues mean that Firebase is more difficult to implement than on Android.

Interface Storyboard Mock-ups

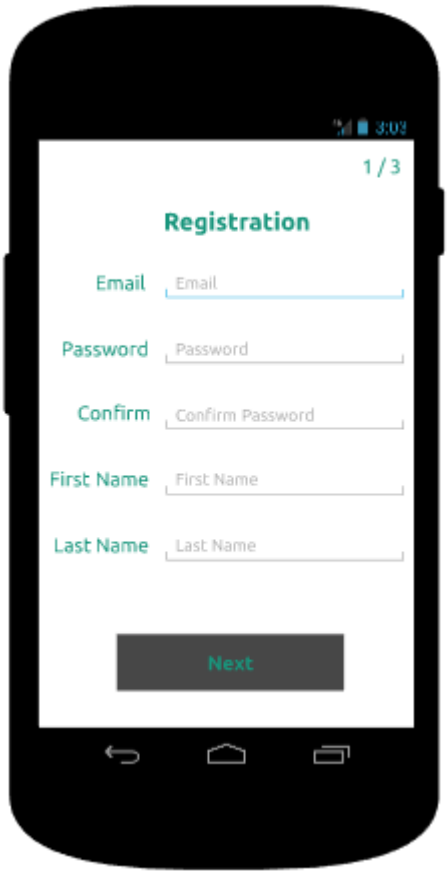
Login and Registration

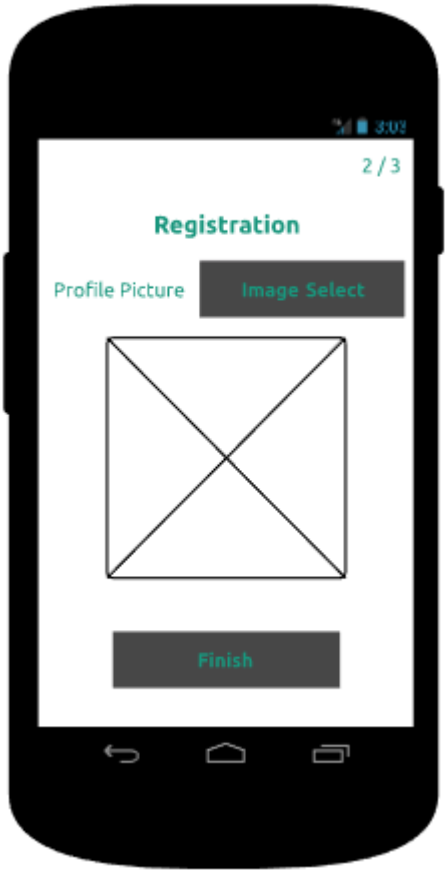
The login and registration screens are to the right and below (respectively). The login screen provides credential memorisation functionality, and the registration screen collects the user's details. These details purposefully are separable from Monash University so that the broader community that interacts with the Sites also has access to the app.

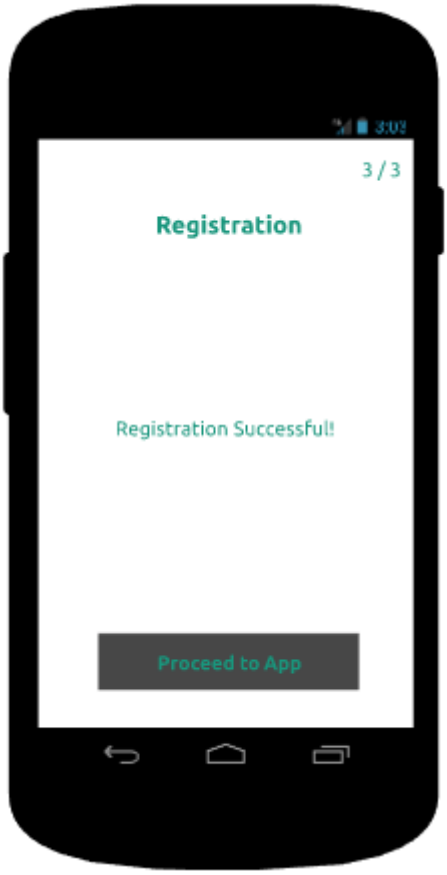


The login button is positioned above the register button since it is a natural progression for a user to enter their credentials and tap the next button vertically down.

The registration form is separated into three sections which are labelled so that the user understands how far through the process they are. One design consideration that could be made here is the incorporation of a back button to make it more efficient to correct a mistake that has already been submitted.



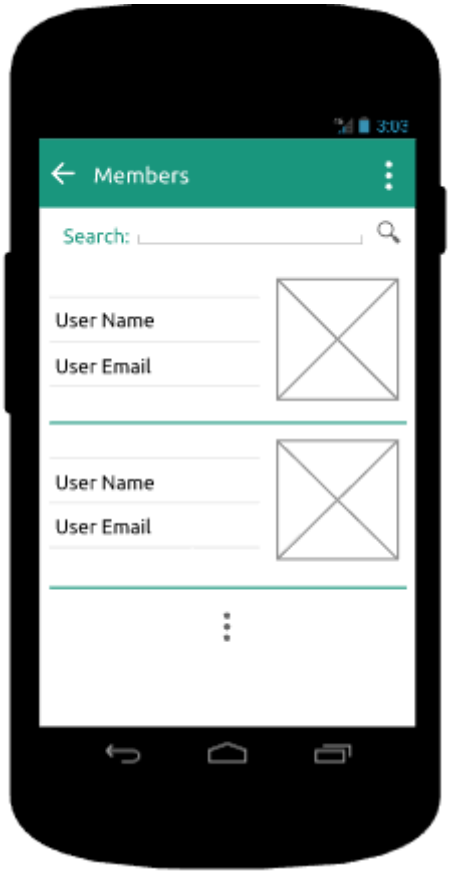


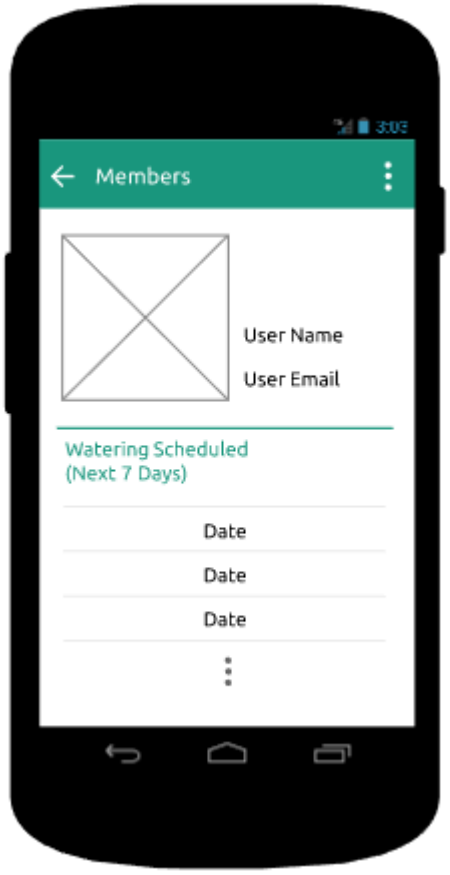


Members View

The following are the members views, which give access to the entire database of members (including a search functionality by name). A user can navigate to a specific members' profile page to view their information and all watering they have scheduled within the next 7 days.

The search functionality and displaying of profile pictures allows for users to rapidly identify other users – especially if users trend towards using a photo of themselves, since most users will have interacted with one-another face-to-face. Dividers between the profiles displayed are used in order for users to be able to make clear distinctions between one profile and the next.



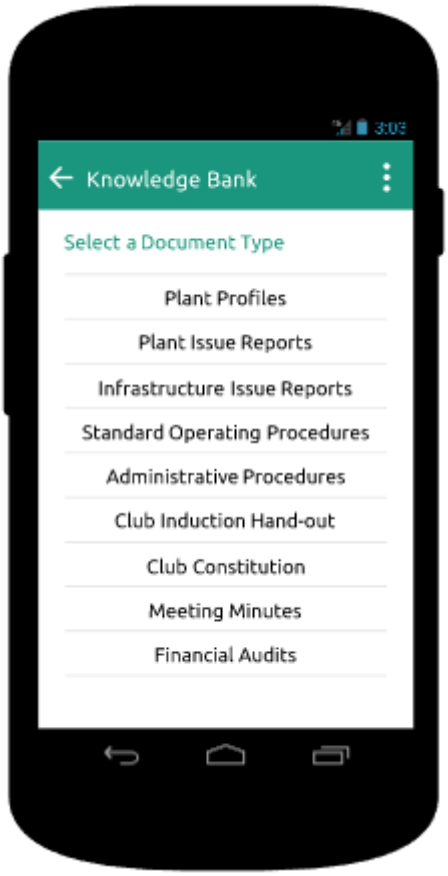


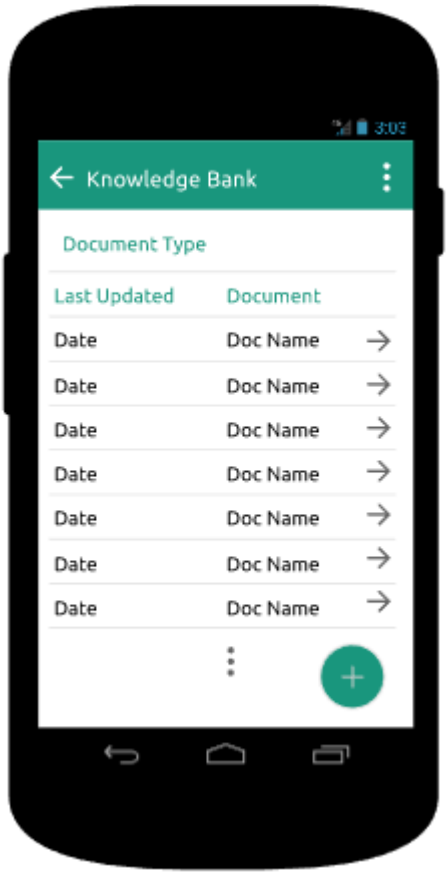
Knowledge Bank – Document Access

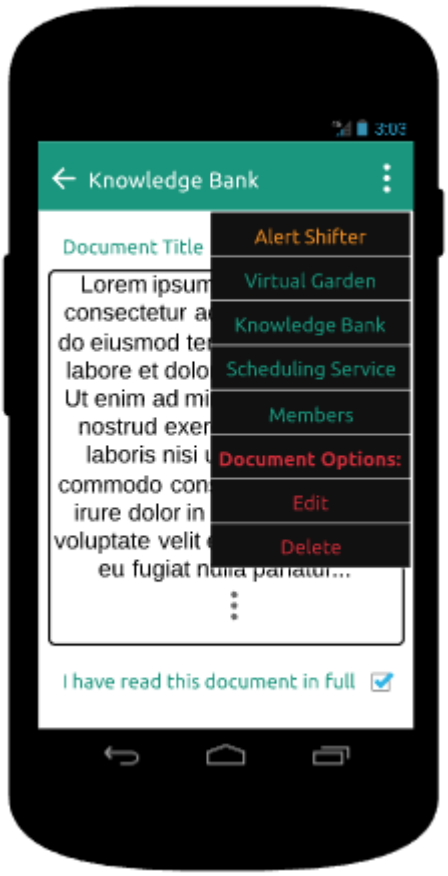
The right mock-up shows the general options menu when in the Knowledge Bank. The options menu here includes (as it always will) an “Alert Shifter”, which is an element that scrolls through the alert that a user has at a given time. This is only ever a maximum of 3-4 alerts. The user can swipe the options menu item to scroll through alerts faster. When the user is viewing a document, Edit and Delete options will exist in the menu for that document if the user is a Committee Member or the creator of the doc.

The below images describe the sequence of locating and viewing a document. First, the user selects a doctype. The user then sees a list of all documents of that time, sorted chronologically. Note that at this stage a floating action button appears that will permit the user to create a new document.
Next the user can select a specific document and open and read it. Located at the bottom of every document screen is a checkbox to confirm that the user has read the document. This is reflected in the user’s account details for administrative purposes.

Simple list views are used in the second and third mock-ups, providing an intuitive interface for locating documents. The separation of docs by type further increases efficiency. One design consideration that could be made here (time permitting) is the addition of search functionality in the screen represented by the third mock-up.



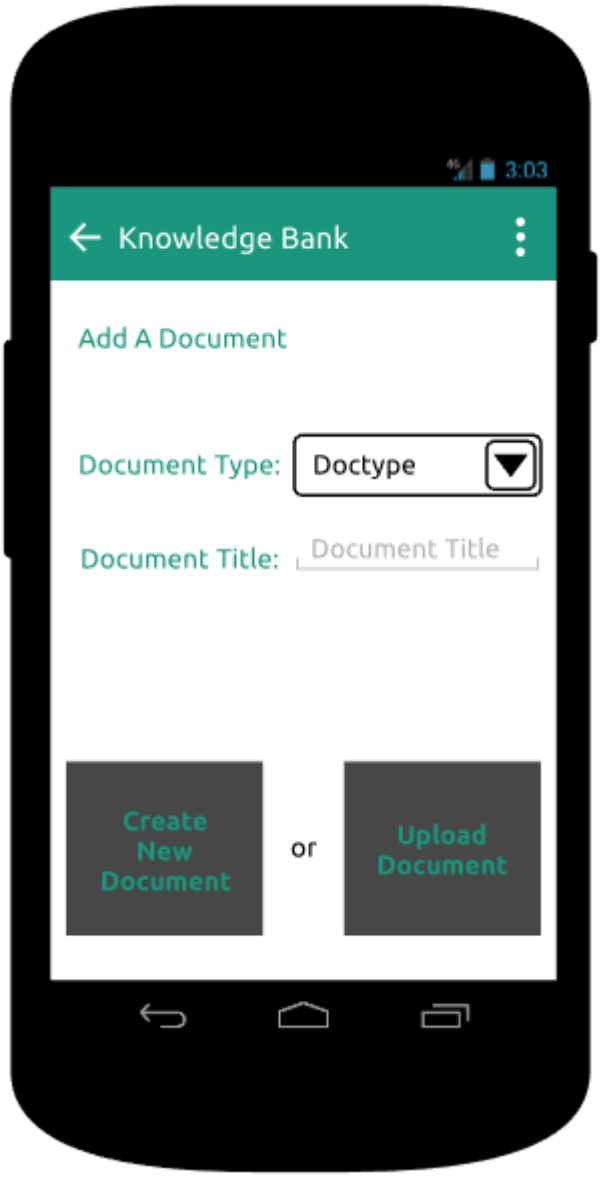




Knowledge Bank – Document Adding

Adding a document to the Bank is a feature accessed through the FAB in the document accessing mock-ups. The document type and name are specified, then the user can either create the document (this option applies only to PPs, PIRs and IIRs), or upload the document from the phone's persistent storage.

The document uploading is shown on the right, involving a simple document picker.



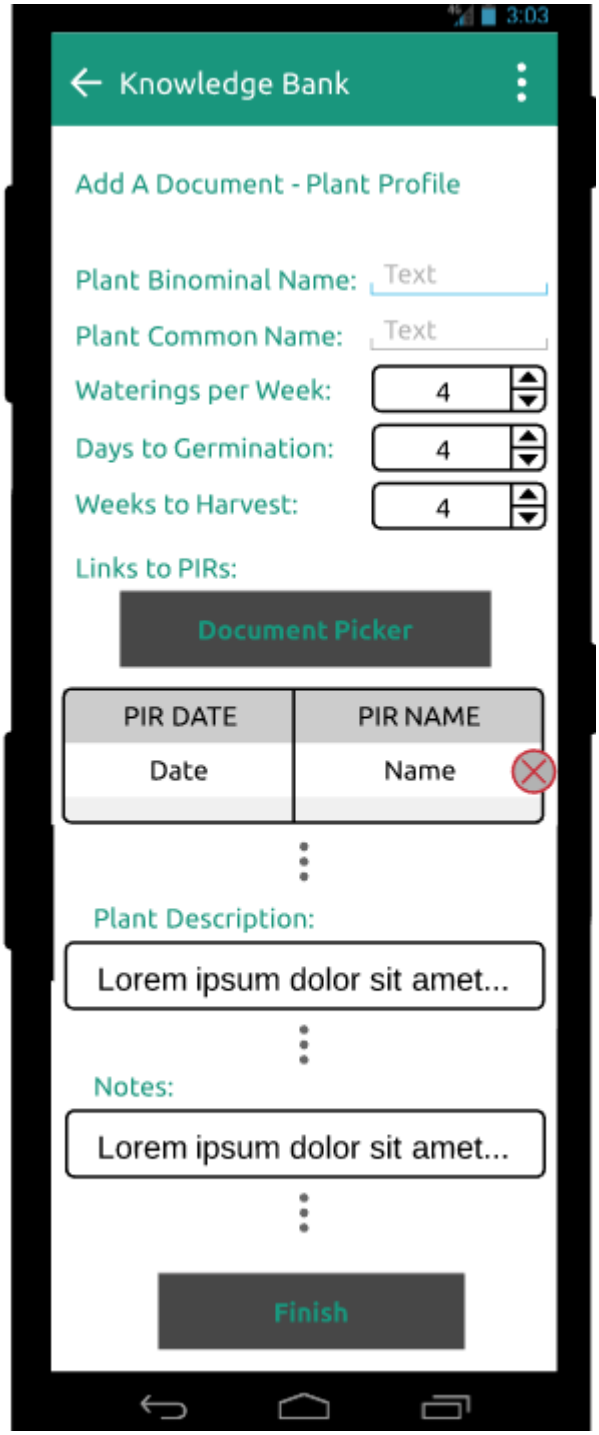
On the right here is shown the document creation tool. This screen prompts the user for a wide variety of inputs to collect information that is as comprehensive as possible. Thus, all fields are compulsory except PIR linking and the Notes.

This mock-up represents a screen which would require scrolling to access all the fields.

The first two fields are used to identify the plant in a PP and on the Virtual Garden. The next three fields are used for scheduling – Waterings per Week is not used automatically but can be considered by users when they are filling availability for watering schedules. Days to Germination and Weeks to Harvest will together determine Harvesting schedules and the end of a harvesting schedule can be used to determine the beginning of the next sowing schedule.

PIRs are linked for the benefit of users who examine the plants – PIRs will ideally detail the exact context in which an issue occurred, how it can be resolved, and how reoccurrence can be prevented.

The Description and Notes again provide extra detail to workers on the ground, Descriptions can be used to identify the plants in cases of uncertainty, and the Notes can be used to detail any special considerations that should be made when tending the species.

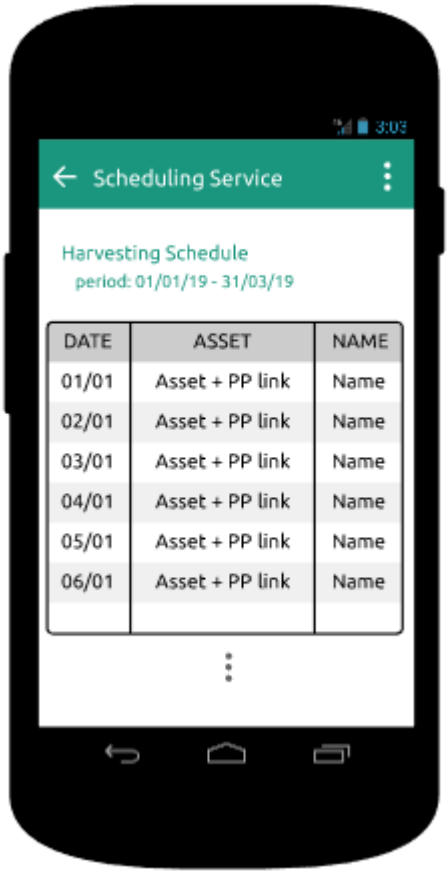


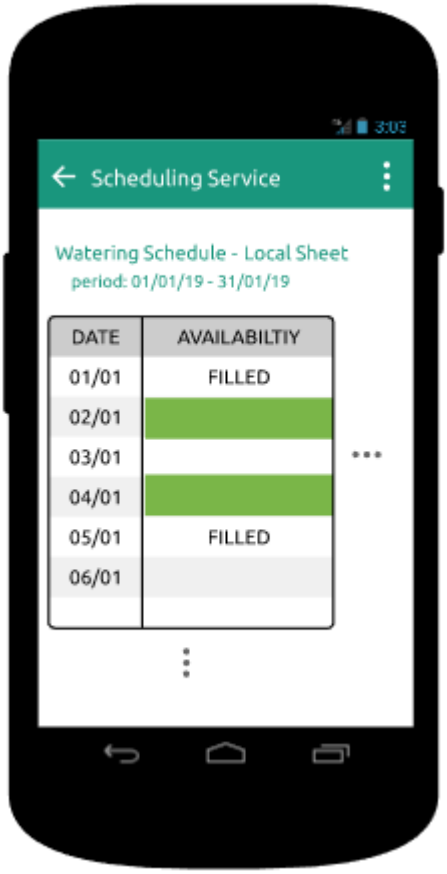
Scheduling Service

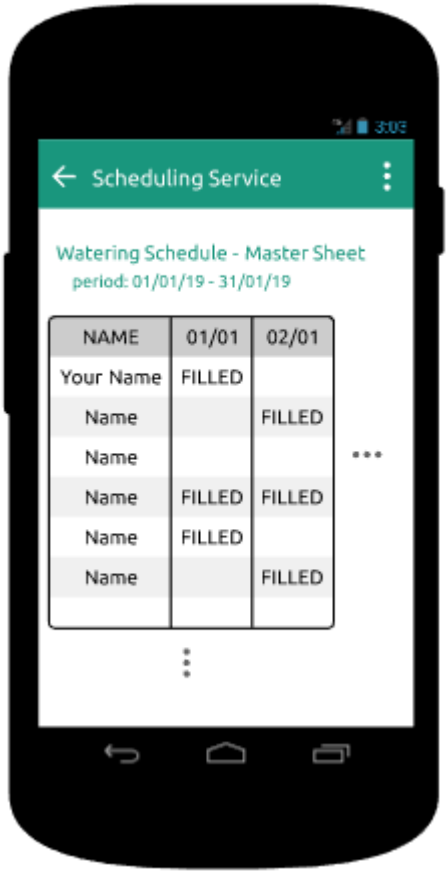
These mock-ups detail the scheduling services. The specific schedules are accessible from the new menu items.

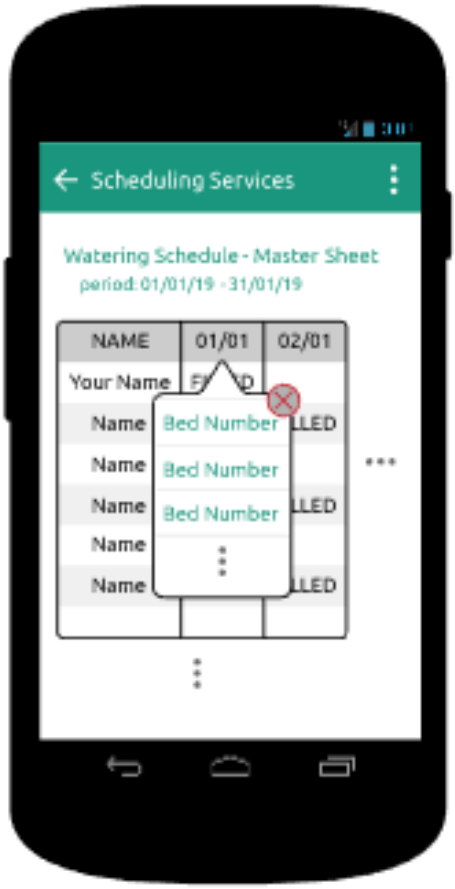
Local availability sheets are filled in simply with a tap on a timeslot and those timeslots can be unchecked with another tap. The master sheet (compiled once per day) will fill already accounted slots with “FILLED”.

Harvesting and sowing schedules take the same form (to make it intuitive and to indicate that they are very closely related). Tapping on the master sheet’s dates will bring up a list of beds to be watered in a tooltip.







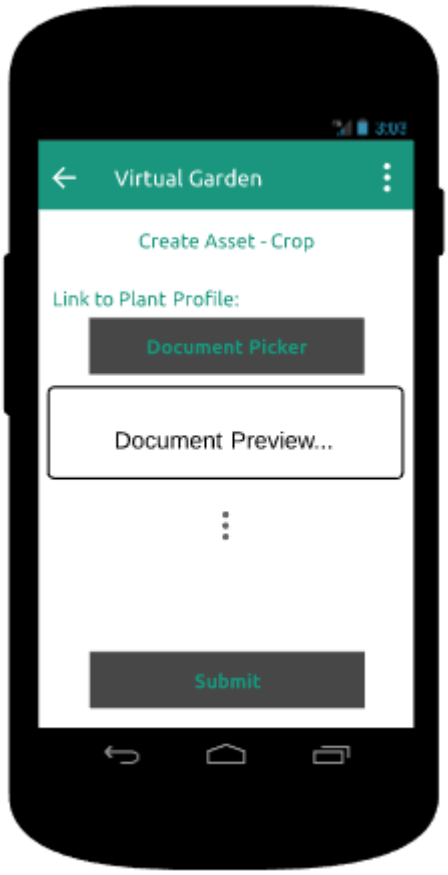


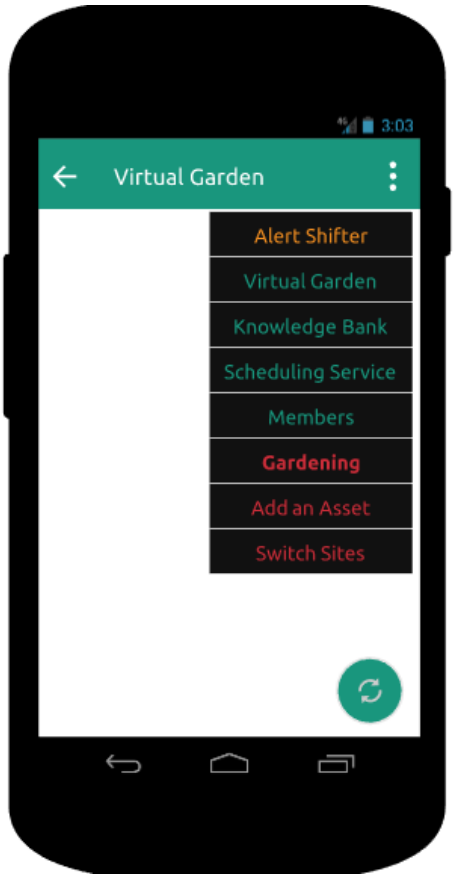
Virtual Garden – Asset Viewer

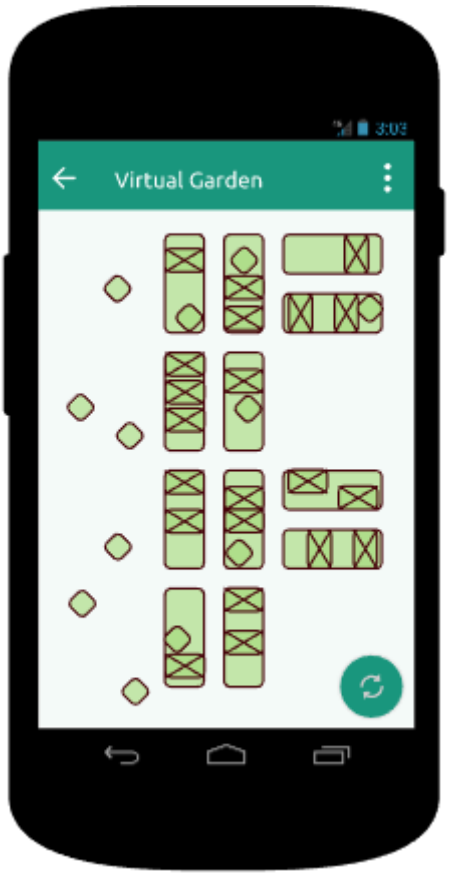
The Virtual Garden mock-ups show that the menu has changed to incorporate Asset Adding and Site switching. The latter can also be achieved by tapping the FAB in the bottom-right.

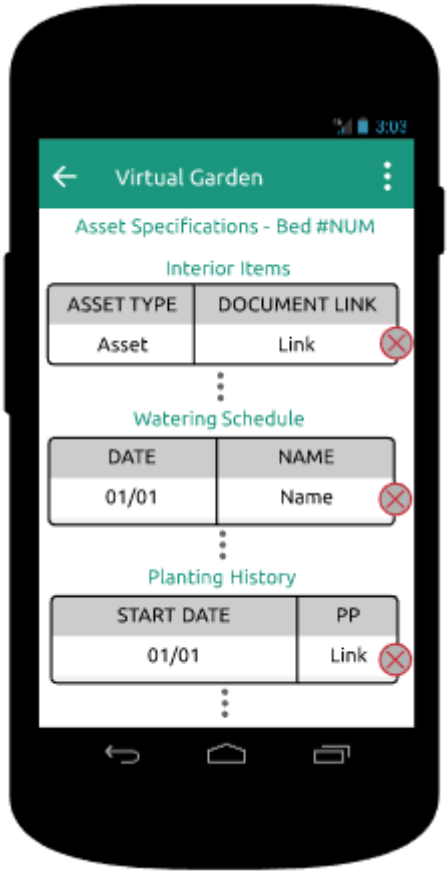
Tapping on a bed or asset “hovers” on it. This shallow view of an asset that a user can quickly acquire allows users to scope out changes that have been made to the farm rapidly. The more detailed Asset Specifications will need to be viewed for an in-depth understanding of all crops to be established.

The simple UI makes for easy ability to discern structural changes in the sites.









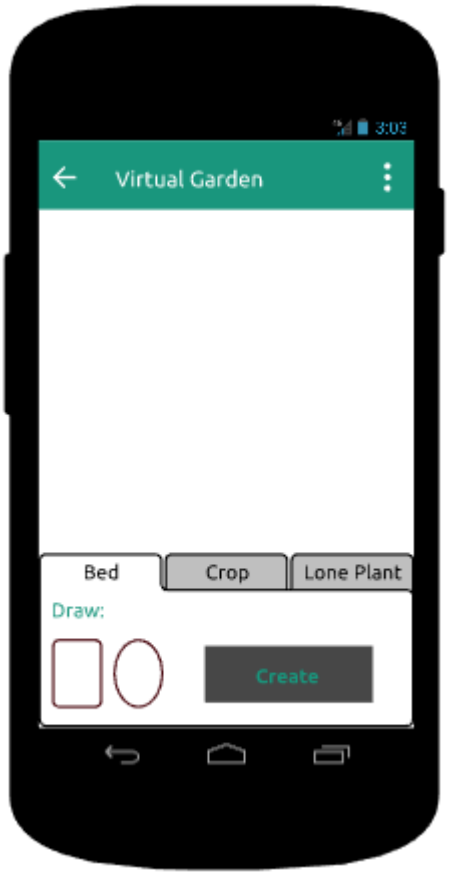
Virtual Garden – Asset Creator

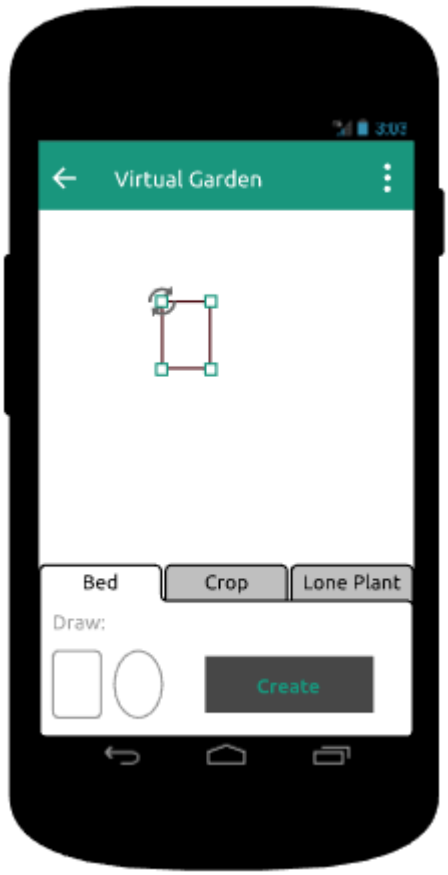
These mock-ups detail the asset creator, the specific view shown is for beds. Crops and Lone Plants can also be drawn to the screen when in a similar fashion.

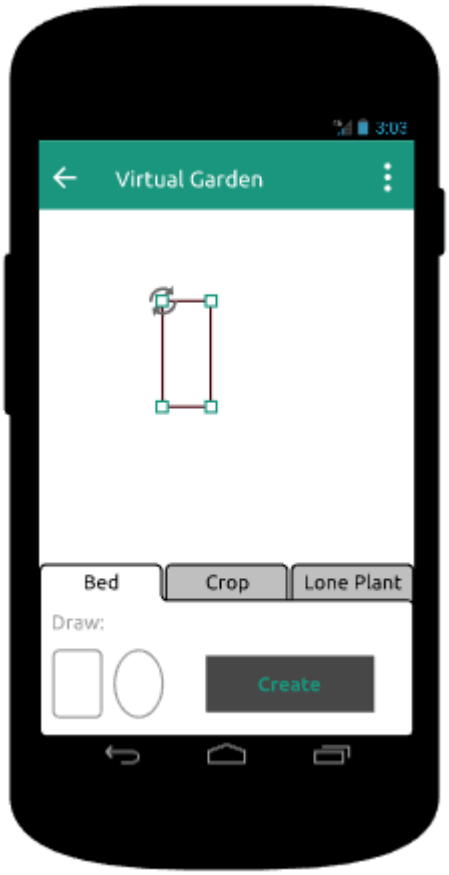
Under normal circumstances, the screen above the creation toolbar would be populated with pre-existing assets.

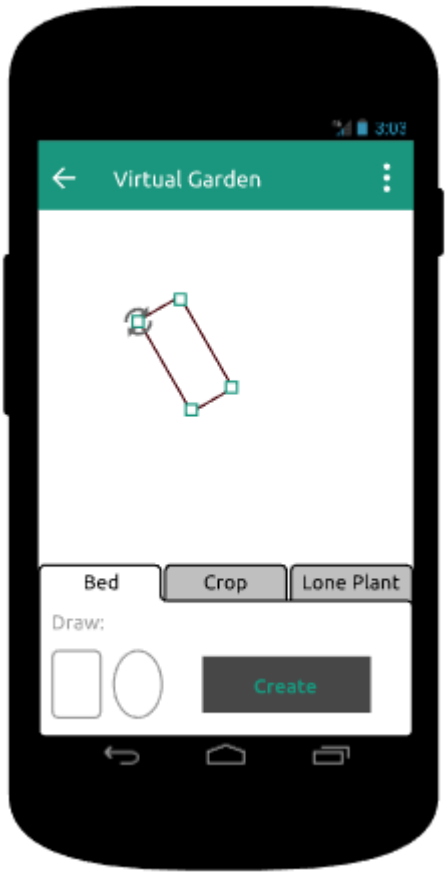
Assets are dragged onto the site from the creation toolbar (of a specific shape). The shapes can then be manipulated in terms of width, height and rotation degree. These modifications combine can create assets of incredibly varied shape and design, and effectively encompass all possible configurations of a site.

The shape manipulation techniques were modelled after those used by Microsoft Office programs, to ensure that most users would have a degree of familiarity with the methods before engaging with the creator.









Design Modularity

This Application Brief outlines mock-ups for only one platform (Android). This is because of the extensive similarity between the Android designs presented and any potential iOS layout diagrams drawn.

The UI has been designed in such as way that each component is modular – it should apply to any mobile device and function intuitively irrespective of OS. Since the limitations on platform availability within MP's member-base mean that Android is the first platform that needs to be targeted, the mock-ups were created with an Android phone in mind (i.e. utilising material design characteristics and bearing in mind the inevitable use of ConstraintView during development). Fonts and colour schemes were selected such that the interface represented a blend of the traditional and intuitive Material Design and the iOS styles that have been seen since iOS 7.

The differences that would be presented in an iOS application would pertain mostly to the floating header (toolbar), which would more closely conform with iOS' minimalistic standards.

Scope and Limitations

The maximal scope of the project could be extensive enough to span many months of development. The scope can be simplified by way of prioritisation of tasks. Of the overarching elements the app is comprised of, the order of precedence is Knowledge Bank, then Virtual Garden, then Scheduling Service. This is based upon the problems the club's members currently face, which largely stem from poor communication and a lack of educational and procedural content made available.
At the point of completion of each element, the product is viable. As such, the minimum viable product would be an application that consolidates the club's knowledge and resources into the Knowledge Bank and provides that to users in a clear, highly accessible format that supports contribution from all users.
In terms of time management, this means that each of the elements should be completed (necessary functionality to be deployed exists and is usable) before development progresses to the next element.

The time allowance (roughly 9 weeks from time of writing) may not permit the inclusion of all functionality, however ignoring time constraints the development is unlikely to experience major setbacks. The most difficult aspect of the application programmatically is the Virtual Garden, specifically drawing shapes to the screen and formatting the layout so that all assets are situated in their correct positions and remain fixed.
A simplification of this task is possible, should setbacks become severe. This would involve using a grid-style layout, such that each cell is a bed. Beds could then be filled with lone plants and crops. This is because those assets are reducible to implicit (invisible data) assets. Hovering on a bed or viewing its specifications would then allow the user to view the details of the plants in it as normal.
An additional source of setbacks will likely be lack of knowledge relating to web-based components such as Firebase, a technology unfamiliar to the development team. This is likewise true for drawing shapes onto the layout, and accessing images and documents stored locally on the device. The time taken to become experienced with these technologies will mean an increased workload in the weeks where these tasks are the focus.

Based upon a survey of the MP member base, most users will be on Android (62.5%), however since the margin is not far removed from a 1:1 split, the number of users that can use it at deployment may not cover a large enough proportion of the active club members.

This will mean that either: the users at deployment will need to be highly active and compensate, or

- the Knowledge Bank will take too long to be comprehensive,
- the Virtual Gardens will not always be an accurate reflection of the state of the physical sites they represent, and
- there will not be enough people filling in availabilities to resolve schedules, and many members completing scheduled duties will not be represented in the schedules.

Presently there is no electricity at either site, meaning that batteries cannot be charged. If the application is particularly hardware-intensive to run, then battery life may become a concern when using the app consistently at working bees and other events. Mitigating this issue will largely be a matter of minimising online functionality except when strictly necessary, and making interfaces closer to minimalistic, as opposed to graphics-heavy.

Estimated Project Timeline

The following is a breakdown of milestones and the tasks that must be completed by the **end** of the milestone period.

Period/Milestone	Tasks for Completion
Easter Break	- Creating a Database via Firebase and connecting it to the app - Designing UI for Home (Login/Sign-Up) Screen - Designing UI for Sign-Up Screen - Establishing Database tables for user information and user logon credentials (linked via User ID) - Handling user registration and credential verification - UI for Knowledge Bank home screen
Week 6	- Designing UI for Knowledge Bank document lister – accesses from local storage - Designing UI for Document Viewer – testing performed with sample documents, screen configuration differs based on document type - Setting up photo capture from phone's camera
Week 7	- Setting up functions to upload and retrieve documents from Firebase (.PDF, .DOC, .DOCX, .XLS) - Create document adding function – uploading from persistent phone storage - Designing UI for document creation, screen configuration differs based on document type - Create document adding function – creation from scratch, including photo library uploads - Create document editing function - Create Document removal function
Week 8	- Copying Knowledge Bank repository to local phone storage with internet connectivity - Set up database tables for Virtual Garden assets, including linking to Knowledge Bank documents - Virtual Garden bare-bones UI - Setting up Virtual Garden grid and zooming - Sourcing images for asset types - Adding assets UI – drag and drop then overlayed screen for asset specifications. Including link to PPs, PIRs and IIRs
1st Demonstration	The Knowledge Bank should be completely functional and can operate alone, and the Virtual Garden should be buildable, however assets cannot yet be edited nor implicitly or manually removed.
Week 9	- Create asset editing function - Create asset manual removal function - Create asset hovering UI - Create asset specifications UI – accessed from the hovering UI - Set up in-app notifications - Set up asset manual connection to Knowledge Bank documents

	- Handling automated deletion of elements – in-app notification seeking confirmation of deletion
Week 10	- Set up notifications external to app - Set up designation of other Committee Member accounts from the President account - Create functionality to automatically generate a schedule when an asset it created - Handle schedule notification – if a user is scheduled for a task generate in and out-of-app notifications - Build schedules into the UI for assets – both hovering and details specifications
Week 11	- Create function to distribute availability sheets to each user when necessary - Handle availability notifications – in-app notifications each day where tasks need to be scheduled - Set up master sheet synchronisation on user's devices when internet connectivity is available - Handle schedule resolution – stop sending notifications and discard all data but the maser sheet on each device and in the database
2nd Demonstration	The Knowledge Bank and Virtual Garden are now fully functional and virtually complete, and if possible, a sample should be developed prior to the demonstration that shows the Virtual Garden's capabilities. The Scheduling is now nearing completion, and availabilities and schedule resolution can be demonstrated.
Week 12	- Build assets into the schedules – a specific task when tapped should display the beds that require watering or plants that need harvesting/sowing - Create functionality to jump to the Knowledge bank when a link to a document is tapped from the Scheduling Service or Virtual Garden. - Create a back-functionality that returns you to the previous screen when pressed - Create user account deletion – authorisation checks are necessary - Set up notifications for the MP President's account so that each year the device logged into that account is prompted to remove the current committee then pass the Presidency on - Devise the app's icon and other key visuals to appear on the app store, etc.
Week 13	- Make any final changes and add additional small functionality if deemed necessary
Final Submission	All three components will be complete now if the timeline remained accurate throughout the semester. If the timeline milestones are not being reached, it is possible that the scope will be cut down so that the Final Submission is only the Virtual Garden and Knowledge Bank.