COSC 345, Software Engineering, 2020

Year-long project: Assignment 1

Leon Hook, Magnus McGee and Tiaan Stevenson-Brunt

Overview

The report is the first of four on the year-long project for the Software Engineering paper, COSC345, at the University of Otago. The theme for the project is ‘anarchy’ and the students have been asked to “….build an app that irritates the maximum possible number of people”.

The app is a productivity app for use by groups of people living in a shared household.

Introduction

A small team of anarchists are going to build an app to organise chores for a shared household.

The archetypal shared household consists of adults who are not linked by any hierarchy such as familial bonds. Without such links it takes time to establish how and when the important but menial household chores are done. These can be as regular as doing the dishes, as timely as playing the bills and as mundane as taking the rubbish out. The natural state of play is to assume that someone else will do it.

The app will offer a shared household a layer of organisation. There is also a hidden agenda: To irritate those that think they can get away with not participating in the day to day chores.

Although there are some shared households that find a way to structure the chores, many will follow the path of least resistance by avoiding conflict. The result allows socially normative behaviour to be established, namely, that the person with the lowest tolerance mess will initiate the work and is also likely to do the work. The app assists in breaking this socially normative behaviour.

The app will allow the household to:

- Make a list of chores for the household.
- Allocate the chores that are to be completed by a given date to household members.
- Share the allocated chore list for distribution to the household.
- Provide with each chore a 'how to do' guide.

The app allows the user to follow the advice of Theodore Roosevelt: "Speak softly and carry a big stick; you will go far". The chores are assigned in a passive manner but those that think they can get away with not doing their chores have to do so in knowing that their participation has been documented and assistance is available.

Project description

The purpose of the app is to help busy shared households organise essential activities.

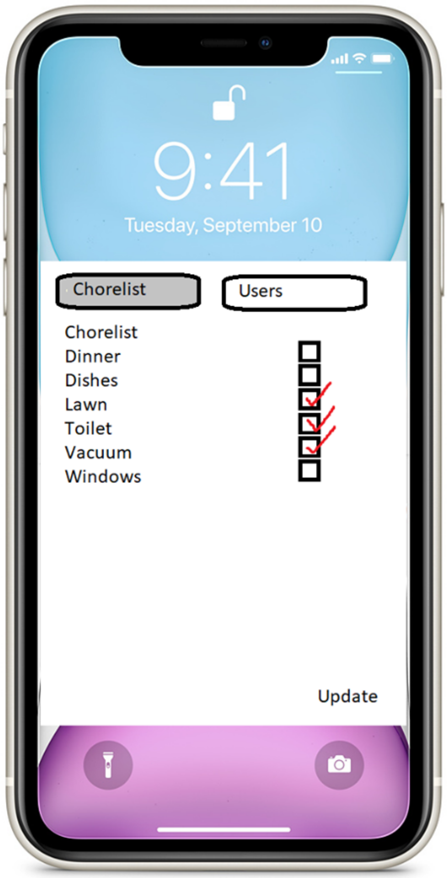
The team have all experienced shared households where dishes would end up piling up and with only a few doing them all. The app 'tells' everyone what they are to do so everyone has the same encouragement to do their chores. For those that wish to avoid doing so, they have to deal with the knowledge of breaking a social contract.

By encouraging people to stick to their chores, the house should be cleaner. With this comes being in a home that is more organised and potentially less stressful.   The app will benefit households that have busy lifestyles.

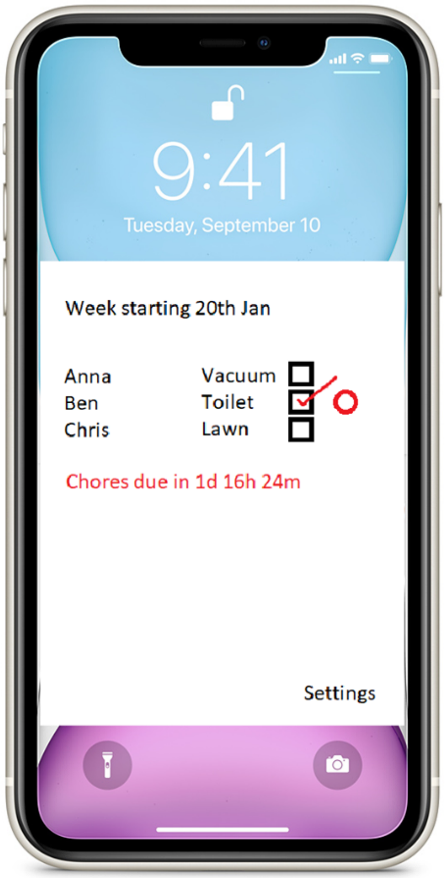
The project is to build an app to organise chores for a shared household with all members of the household having the option to have the app on their phone. The alpha version proposed here is a single user app.

Features of app

Exhibit 1: The design of the alpha version of the app describing the household tasks, allocation of chores, people's details and the how to guide.

<table>
<tr><td colspan="2"><b>Household tasks</b> are selected from the master chore list (below, left).
Selected chores are then <b>allocated</b> to household members with information of when they are to be done (below, right). Both screens use tick lists.</td></tr>
<tr><td>

9:41
Tuesday, September 10
Chorelist
Users
Update

Chorelist
Dinner ☐
Dishes ☐
Lawn ☑
Toilet ☑
Vacuum ☑
Windows ☐</td><td>

Week starting 20th Jan
Anna Vacuum ☐
Ben Toilet ☑ ○
Chris Lawn ☐
Chores due in 1d 16h 24m</td></tr>
<tr><td><b>People's details</b>. For each member of the household their name and email address.</td><td><b>How to guide</b>: Chores are grouped into similar tasks with advice and resources on how to do the task.</td></tr>
</table>

Point of difference

Productivity apps, also known as planning and organiser apps, are widely available on Google Play and Microsoft Store. These are mainly for individual use and come with various levels of detail often based around either the to-do list or note taking. A few of these have been extended to households aiming at the family unit (whether that be a traditional nuclear or otherwise). Although both the individual and family unit apps appear to be similar to our proposal, each has a point of difference.

The individual apps are concerned with assisting in organising only one person's life. Here only one person gets to see what is on the app. The proposed app moves beyond this by encouraging sharing of the chore list with all members of the household.

In the family unit apps, the family unit often has a hierarchy and with that, a few people in the unit that can lead (or dictate) decisions around how chores are done, assigned and what an acceptable standard of work is.

The app is aimed at a shared household generating to-do lists for more than one person for a group of people that are still to develop the structure of a family unit.

Exhibit 2: Examples of productivity apps for the individual on Google Play.

App	Creator	Description
Get S**t Done!	Plafhof	Uses a sardonic tone to manage projects of a few hours long.
Study Bunny: Focus timer	SuperByte	The cute, primary school drawing front end hides a separate time logging for 12 projects with text based to do lists.
Tasks: Todo list, Task List, Reminder	Stephen Nottage	Multiple reusable lists in a simple ‘grocery list’ style where the user can also export the list by email.
Microsoft OneNote: Save Ideas and Organize Notes	Microsoft Corporation	A text editor app with features to format and annotate lists.

Exhibit 3: Examples of productivity apps for households on Google Play.

App	Creator	Description
Allowance & Chores Bot	Wingboat.com LLC	Aimed to encourage family members to see the connection between chores and pocket money.
OurHome	OurHome	Single user interface aimed at organising a family unit.

Resource requirements

The resource requirements to deliver the project are time, administration, software and hardware. The schedualbe resources for each of these is under review as New Zealand, where the team are based, is responding to the COVID-19 (previously known as “2019 novel coronavirus”) pandemic. Travel is highly restricted and team members are obliged to remain in isolation in their respective shared homes.

Time

The team estimates that the project will require one person at full time equivalent for the duration of the project. This would be around 24 hours per week, six each for each team member and six hours of administration and meetings.

The team are all studying full time at the University of Otago. Two are taking a DipGrad which is an intensive degree to be completed in 14 months, making it a busy year. The third is working part-time in addition to their full time study schedule.

The team has found other ways to organise and have meetings but managing workflow is still of concern. In response the team has made an effort to make the project modular. The planning stage is still under development due to the changes brought about by the pandemic. This will be reported in Assignment 2.

Administration

The team has used Google Drive for Assignment 1 and will move more of the project over to GitHub.

Software and hardware

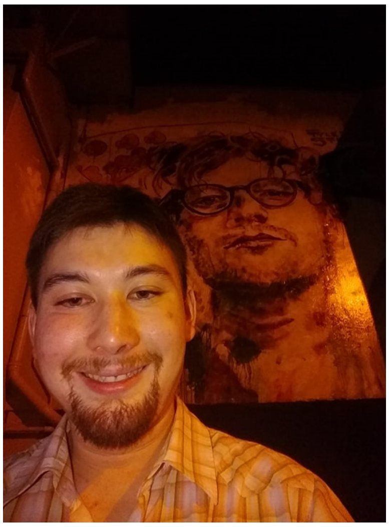
During the COVID-19 lockdown, the team doesn't have access to the University computing laboratories to work on the project.

The original idea was to use Swift to create an app that works on iOS devices however only one member had access to a Mac computer to code on at home so we changed so that everyone could code at home using Android Studio which is available for both Apple and Windows devices.

Organization

There are three members in the team: Leon Hook, Magnus McGee and Tiaan Stevenson-Brunt. The team are all from a class of around forty students in the 2020 intake for the COSC345 Software Engineering paper. Many of the class had already studied together and quickly formed small teams.

Exhibit 4: Photographs of the team.



Leon Hook

Magnus McGee

Tiaan Stevenson-Brunt

The remaining twelve students discussed the project as part of tutorials during the first month of the paper. During group discussions between the remaining twelve students as part of tutorials during the first month of the paper, it became apparent that all three are good listeners and this brought the team together.

The team

Leon Hook

Leon completed a BA majoring in music in 2019 and has returned to full time study a DipGrad in Computing Science in his fifth year studying at University of Otago.

He has experience coding in Python, Java and a bit of C. He will bring enthusiasm and has skills in design and interfaces. Leon is also the most artistic of the team and has created images used in the report.

Magnus McGee

Magnus has returned to full-time education to do the DipGrad in Computing Science. He has twenty years of experience as a research statistician working on medical studies mainly in epidemiology. Projects have included mental health, cognitive function and ageing and screening methods in breast cancer. He brings to the project experience in data analysis, including handling large datasets and statistical modelling, writing academic papers and developing standard operating procedures for a clinical trial.

During his research career, Magnus often found himself having to work for more than one master. Although a project would have a lead researcher, the politics of medical surveys meant that Professors from different departments wanted a say in how the survey was run. This helped develop skills in organising team meetings.

Tiaan Stevenson-Brunt

Tiaan was born and raised in Auckland but now is “lucky enough to call Dunedin [his] home”. He is a fourth year student at the University of Otago University, studying for a BSc in Computer Science and a BAppSc (Bachelor of Applied Science) in Software Engineering.

Tiaan also works part time as the resident DJ at Suburbia Nightlife and performs in different venues throughout New Zealand on a regular basis.

After living in what some call a “typical scarfie flat” on Castle Street, Tiaan brings an in-depth understanding on what the flat cleaning rituals (or lack off) are really like including first hand experience.

Tiaan brings to the team a range of skills that positively contribute to a small team setting. His skill set comprises being a keen and able coder; has demonstrated leadership capability and has a clear and concise communication methods. He is a valuable team member and is looking forward to working alongside his classmates whilst they undergo the project.

Roles

The project roles are under review as the teams’ travel has been limited by the COVID-19 pandemic (see *Resource requirements* for details).

Project breakdown

The following table gives the essential and desirable features of the alpha version of the build. Further features are under consideration, each comes in modular form to allow the app to expand as the build progresses. This has been taken for this report since planning a timeline has great uncertainty given the circumstances of the COVID-19 pandemic (see *Resource requirements* for details).

Exhibit 5: Essential and desirable features of the alpha version of the app.

Essential	**Desirable**
★ Single user app. ★ List of possible chores. ★ Selection relevant chores. ★ Allocation of chores. ★ Report of chore allocation with time lines. ★ A guide on how to do the chore.	❑ Multiple user access, some with admin privileges. ❑ Users can add chores to app.

Exhibit 6: Further features are under consideration in ascending order of potential irritation

- More user features around chore progress such as viewing and recording progress.
- Additional notification features.
- Allow chore swapping between household members.
- Admin access to allocation of chores.
- Extra chores the following week as well for not completing chores.
- Biased allocation of chores with non-compliers receiving the task perceived as unpleasant.
- Admin able to penalise householders who were not honest about doing their chore.

Methods

The project will follow the five phases of the System Development Life Cycle :

1. Planning,
2. Design,
3. Development,
4. Quality Assurance and
5. Release.

Each one of these stages will also be our milestones for our project which will enable us to track our progress through development.

Planning

The planning is essentially what's included in this report and is our first milestone we will tick off. The planning phase identifies who our target market is, what our app idea is, how we will make the app, why this app will meet the brief of causing anarchy, and lastly when will we complete each phase of the development cycle.

Design

The second Phase of our application will be the Design. The primary focus of the design phase is creating the User Interface/Experience. Firstly, using sketches on paper, we will draft the design of the app and then will transfer these drafts to wireframes. Our design phase will be a similar approach to that of designing a website with the use of wireframes of pages. The last step of the design phase is making a prototype of the app. This 'Prototype' or draft is a complete front-end version of the app, it will act as a skeleton of the app. It will look like the finished product visually but won't have any functions and so is like a façade of a working app.

Development

Next up is the nuts and bolts of the project, the development. During the development, our team will make the code completing the back end of the application. We have decided to create an android based app using Android Studio and will work as a full collaborative coding team, equally distributing miniature assignments throughout the process. We initially planned on creating an iOS app but due to hardware constraints we decided Android would be a better suit for our current situation in lockdown. GitHub will play a crucial role during this phase allowing us to work at distance while maintaining a productive workflow. We have agreed to have weekly meetings using Facebook's messenger calls where we will review team members' progress on their individual assignments, update our GitHub master branch with new code and gauge what we need to work on next.

Quality Assurance

Phase 4 is an ongoing process, it's where we constantly test the app itself. Testing will be incorporated thoroughly throughout the development of the app to limit bugs. We will mainly focus on testing the functionality of the application to make sure everything is working as it should be, but will also test the interface ,and device compatibility.Our primary method of testing will be using the android studio with its in-built emulator itself and then transferring the app to phones with our alpha release. We will mainly test our app by generally using the app and finding bugs as we go but also trying to break the program where we can. Some ways we can try to break the program are inserting invalid inputs where, using the app under stress e.g. low battery and general misuse of the app. Once we have the app fully working our alpha release will involve the team testing the app with the above methods fixing bugs along the way. Our beta release will expand our testers to close friends and family and also will potentially include some added features of the app. The Quality Assurance phase as mentioned above will be an ongoing process with the team testing each new added feature as we go. This will minimise large scale bugs with our initial releases.

Release

The final phase and last milestone for our project will be the release. Typically, this milestone would be where the app reaches the market and involves the ongoing maintenance of the app but our group is changing this slightly and our release is the final hand in of the assignment at the end of semester 2.

Discussion

Software development models allow efficient use of resources and aid in planning. The waterfall and agile model are two primary models used in the software development industry. The waterfall model consists of different stages where you work from top to bottom ticking one stage off completely before moving onto the next. It's called the waterfall model as the process moves like water in the waterfall, only traveling down and cannot go back up. The waterfall model is extremely inflexible and can make the development process a slow and lengthy one. Agile on the other hand is an extremely flexible model where change is encouraged. This unstructured approach of development allows for continuous additions to the development throughout. Agile developers regularly meet and discuss progress and what's needed next in the project. The System Development Life Cycle we are following does appear to favour a waterfall development model due to the nature of each phase following a natural order but we will in fact be using a combination of both agile and waterfall. Our development will need agile as we plan to go back up the phases and add new features later in the development such as adding multiple users which will require us to restart the 'waterfall' from phase 2 again.

We can't fully estimate how long each stage will take due to the agile nature of our model with additions being added in throughout. In saying this we naturally will be following the deadlines of the course and can roughly estimate when each phase of our development will be complete. Phase 1, the planning will be fully ticked off with the hand in of this report on the 22nd of April. The design and development (Phase 2 & 3) will be complete on the deadline of the 2nd assignment on the 29th of May with the alpha release. Although with the alpha release phases 2 and 3 won't be fully complete as we will traverse back up once we add new features for the beta. The beta's deadline is currently the 31st of July and will require us to once again go through Phase 2 and 3 but also phase 4, Quality assurance. During the beta release, immense testing will be done to ensure the final release is without a hiccup and so lastly the final phase the release will be complete with the final deadline of when we hand in the app on the 25th of September.

Identifiable activities

- ➢ See exhibit *Essential and desirable features of the alpha version of the app*.
- ➢ Project costing. Discussed in *Resource requirements* above.
- ➢ Project monitoring and reviews. Under review as consequence of COVID-19 restrictions.
- ➢ Staff availability. An estimate of staff time is provided in *Resource requirements* above. This includes the 'training' time acquiring new skills.
- ➢ Report writing and documentation. Needs to be reviewed given the experience of preparing this report.

Milestones and Deliverables

- See exhibit with *Essential and desirable features of the alpha version of the app*.
- The length of build is outlined in the following exhibit.

Exhibit 7: Build times based on the *Project breakdown, Methods*.

Date†	**Deliverables‡**
22nd April 2020	Planning.
22th May 2020	Planning review; Design; Development
24st July 2020	Development ctd. Quality Control.
18th September 2020	Release
Key: † Based on report deadlines with an adjustment for writing time. ‡ Phase as described in Project breakdown	

Risk analysis

We have identified a few foreseeable problems that we will need to work towards conquering. Firstly, this is all our first time making an app and so will need to learn the ins and outs of android development. Our team has already taken steps towards improving this situation by downloading essential software and beginning to learn the syntax. As all of us have prior Java experience this shouldn't be too much of a problem. The next problem is due to the COVID-19 pandemic we are all working at distance. We aim to solve this problem by having weekly meetings using Facebook messenger calls and also regularly contacting each other by message through Facebook messenger. By staying in contact so often we can regularly review our progress, allocate new assignments for individuals and generally discuss where we are at on our project. Our group has the invaluable attribute of good communication so we should deal with the distance issue without a problem.

Planning review

There is a need to review the planning of the project (as mentioned in several p0laces in the report) around the uncertainty created by the COVID-19 restrictions . The review will also need to consider:

- Further consideration of the time to milestones.
- Allocation of people to tasks with careful consideration of the time required for monitoring and reporting.
- How to use the features in GitHub to aid project monitoring.

References

- Department of Computer Science, University Of Otago, 2020 (2020, April 21) Welcome to COSC345 - Software Engineering.
- The phrase finder (2020, April 20) Speak softly and carry a big stick.
- World Health Organization (2020, April 22) Naming the coronavirus disease (COVID-19) and the virus that causes it.
- New Zealand Government (2020, April 22) What you need to know.