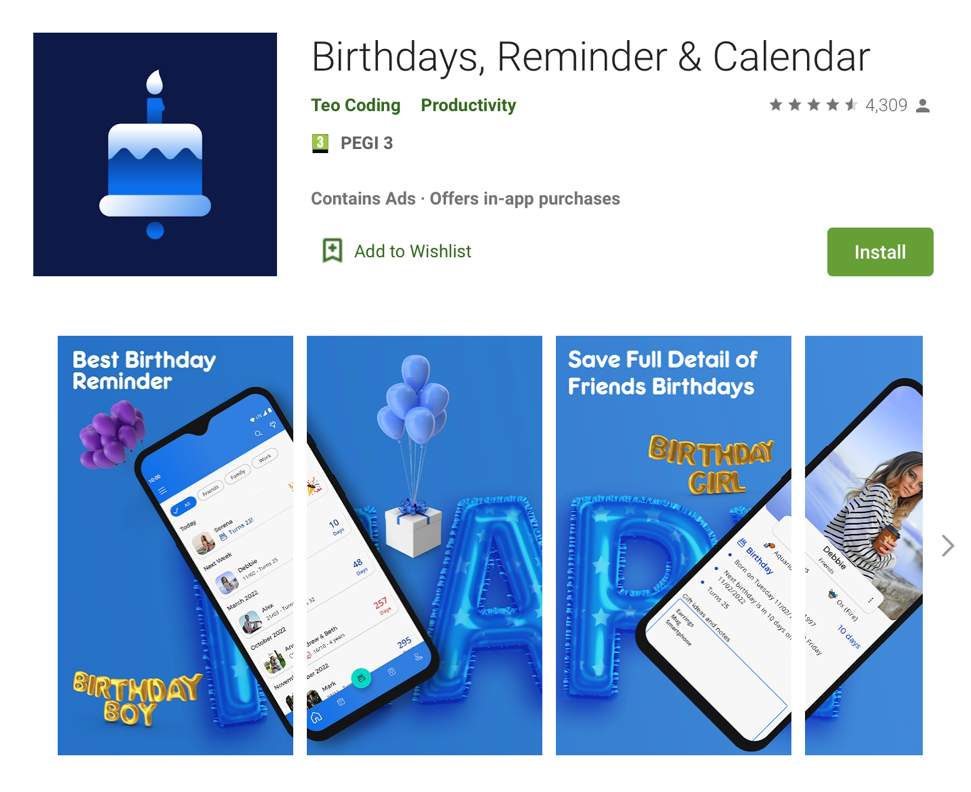
|  |  |  |
| --- | --- | --- |
|  | |  | | --- | | **Birthday** By Lewis Gray  Student Number: 19013887 | |

App Idea

Birthdays is an app designed to ensure that you’ll never forget when someone’s birthday is.

Existing Apps



**Like:**

-Tutorial to show the user how to use the app when they launch it for the first time

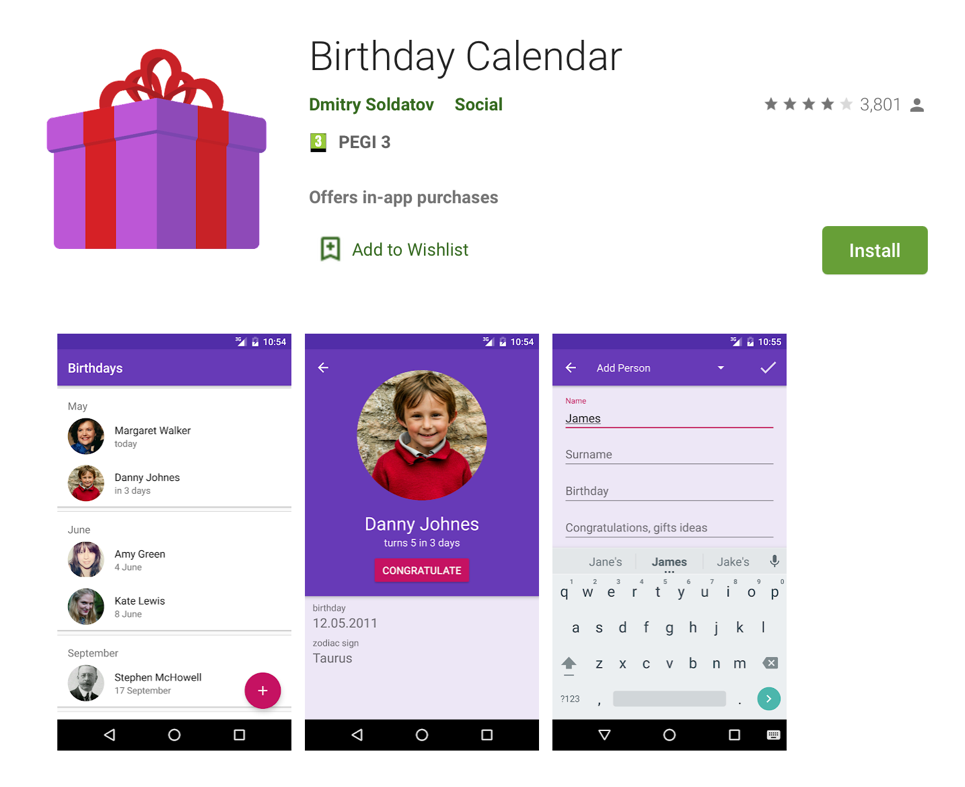
-Dark mode...!

-Tabs at the bottom make navigating the app feel natural and easy.

**Dislike:**

-Advertisement preferences

- Advertisement is placed where the tabs at the bottom should be placed, making it easy to mis-tap and open an advert. (This seems intentional and feels like a big middle finger from the developer to the user 🚩)



**Like:**

-Birthdays are listed in an upcoming order

- Add button in an ergonomic place

- Optional photo can be taken on the spot

-Zodiac sign is told

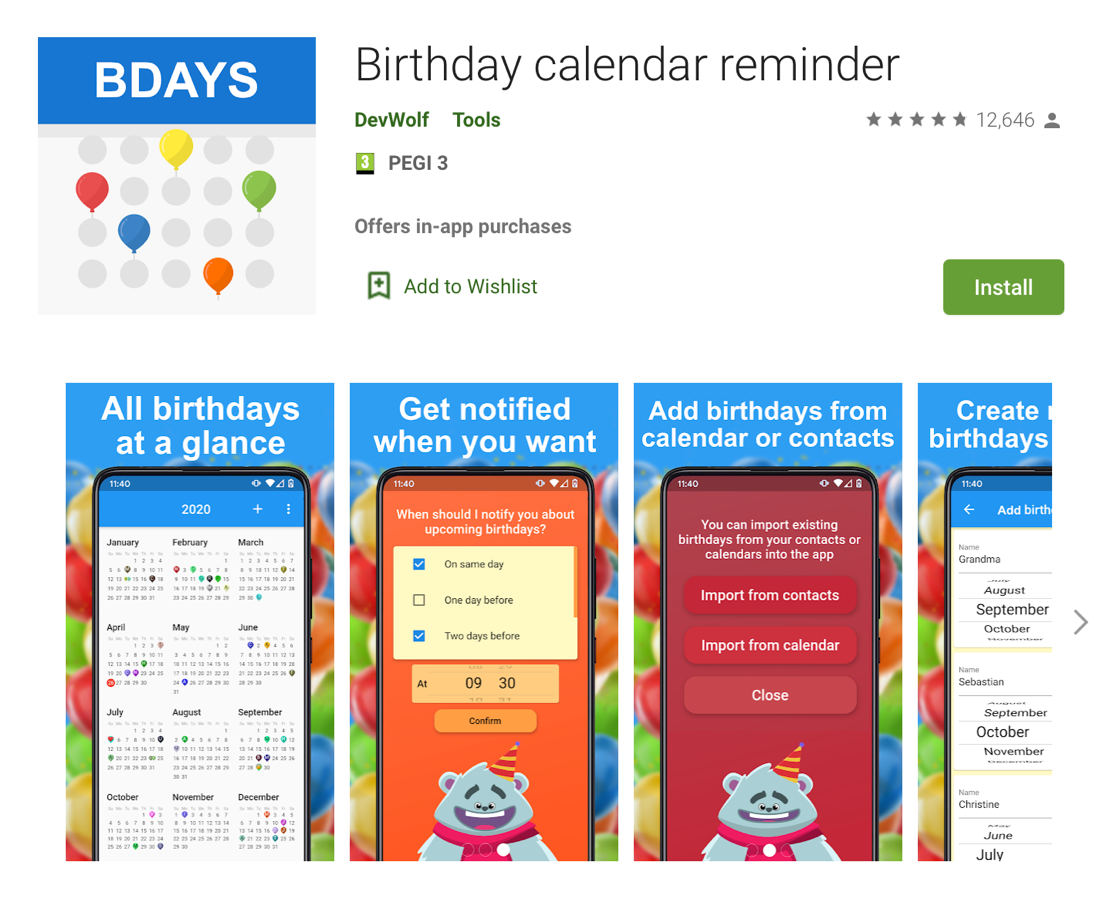
**Dislike:**

-Premium version

-The purple and pink colour scheme

- All on one page design feels restricted and cramped

- Reminders are generic and the user is only notified once, at a time period before the birthday that they set.



**Like:**

-Swipe animations

-Set multiple reminders

-Can add birthday without year

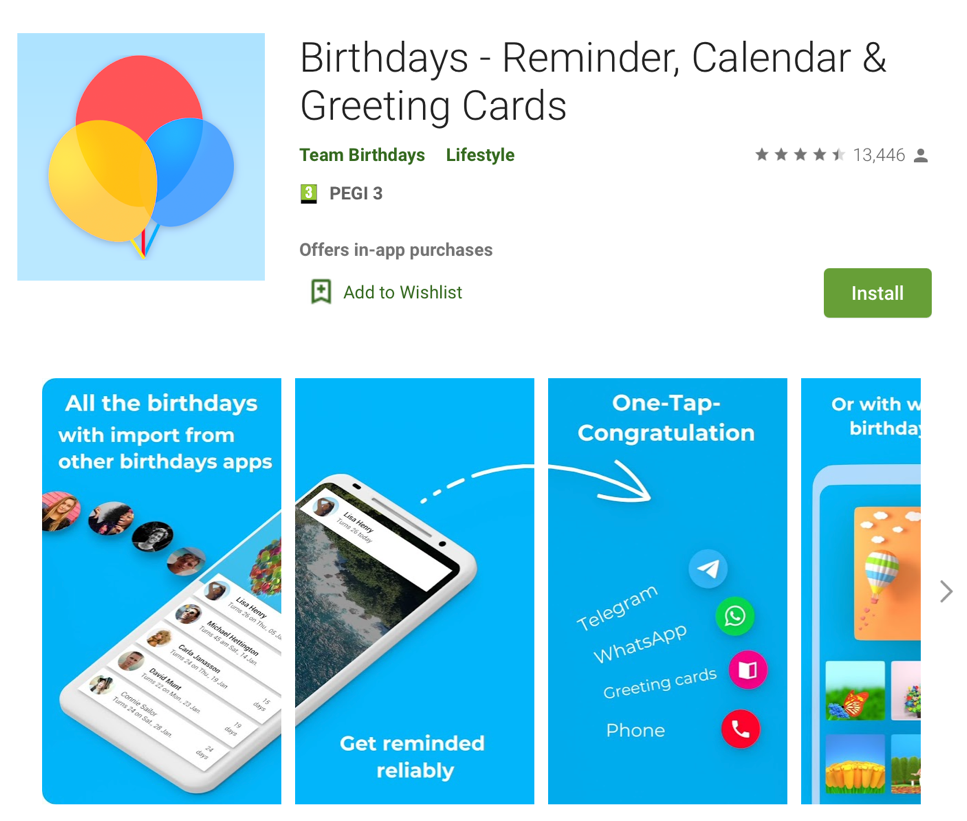
-Supports other important dates other than birthdays

**Dislike:**

-Seeing every day in a year on one screen is quite overwhelming

-Fun swipe animations aren’t consistent throughout the app, behind it is a very generic looking calendar

-Cannot add notes regarding a person or their plans



**Like:**

-Can send messages in one tap from the app (Uses external systems to do so)

-Imports contact names for autocompletion when adding a new birthday

-Input their age instead of the year they were born

-Can search for people to find their birthday

-Animations and transitions feel smooth (swipe navigation)

**Dislike:**

-Premium version frequently shoved in my face whilst navigating the app

Requirements

1) The user can add a friend with a birthday plan

2) The user can modify a friend and their birthday plan

3) The user can delete a friend and their birthday plan

4)The user can view details of a friend and their birthday plan

Friend is used as an umbrella term to describe a person that the user wishes to remember the birthday of. The rationale behind these requirements is that the user needs a way to: Create, Read, Update and Delete information regarding the people whose birthday they need/want to remember.

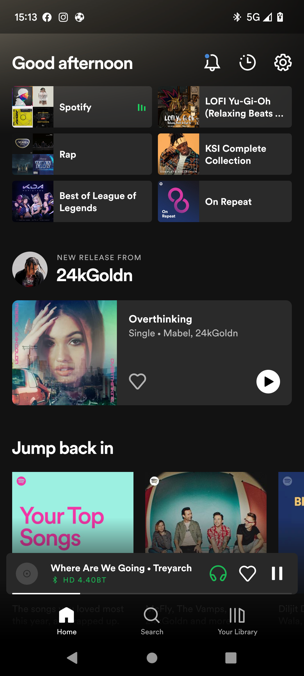
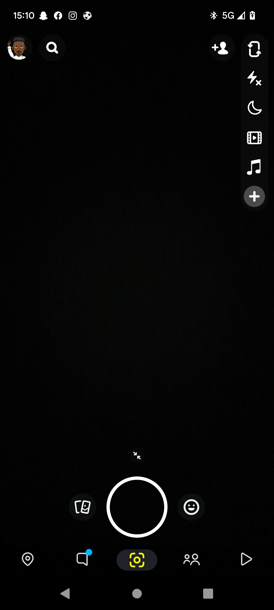
By using Model View Controller architecture, I can create and manage a persistent data store to facilitate CRUD operations whilst keeping the code to a standard which others can understand and implement.

5) The user can navigate around the app using tabs at the bottom of the screen

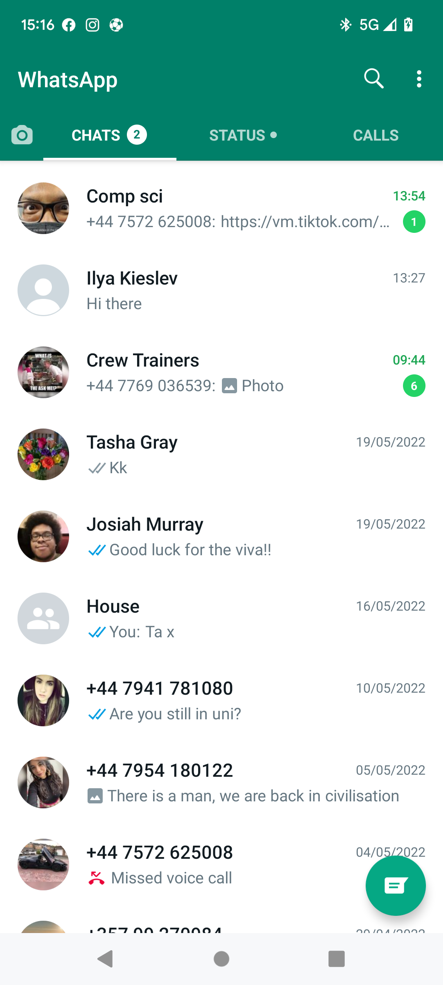
6) The user can navigate around the app using swipe motions

This requirement is directed towards the target audience: everyone. As this app is isn’t for everyday use, it needs to fit in and feel familiar with other apps on your device. By looking at some of the most installed apps on the marketplace such as: snapchat, Instagram, Spotify WhatsApp and Facebook I was able to infer two common methods of app navigation.

Method 1 – Tabs at the bottom.



Method 2 – Tabs at the top with swipe to navigate



Both methods accommodate for the user navigating the app with the bottom part of the screen. This is good practice for apps as most users use their thumb to interact with a mobile device and the bottom of the screen is therefore more accessible. From personal experience, both methods consequently feel as natural as each other when it comes to finding your way around the app. Snapchat was the only app that I found that integrated both methods. It allows the user to use the app how they’d like to us the app and this freedom improves the user experience. As a result, I’ve decided to also implement both methods into my app to ensure the user feels as comfortable as possible while using it. This also eliminates the need for a tutorial.

7) The app should send notifications to the user if they are not prepared for an upcoming birthday

8) The user can be either: unprepared, semi-prepared or fully prepared for a friend’s birthday

All of the similar apps I looked at incorporate sending push notifications to the user in order to remind them of an upcoming birthday. These are set to send at a user defined time period before the actual birthday, and do not repeat. I’m going to build on this idea by having the app send out notifications to the user with a frequency defined by the length of the time period between the current date and the birthday.

In order to stop notifications coming, the user will need to be “fully prepared” for the birthday. This means that they’ve told the system that they’ve:

* Sorted out a birthday card
* Sorted out a present
* Made a note of the celebratory plans

If they have done none of the above, they will be “unprepared”, and if they have done some of the preparation they will be “semi-prepared”

9) The app could implement speech recognition to provide a hands-free method of using the app

Similar to other personal assistants (eg. Bixby, Siri, Google Assistant), the app could use the device’s microphone to listen out for a wake-up phrase. Once heard, the app will recognize speech such as “I have bought Týna’s card” and then perform the appropriate action (in the given case, it would tick the box that says a card has been bought for Týna). None of the similar apps I looked at used a feature like this, which potentially signifies a gap in the market.

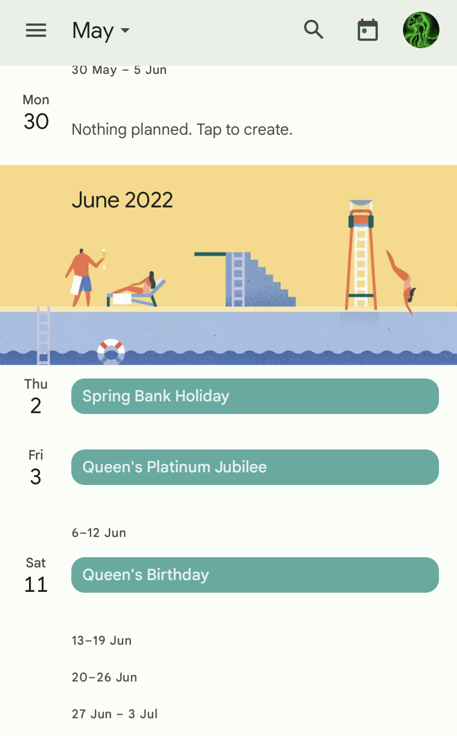
10)The system should populate a calendar with birthdays in their appropriate place

11)The user should tap on a date in the calendar and be presented with all of the birthday plans on that day

12) The calendar should scroll vertically to show its dates

Calendars were used in half of the apps that I found that were similar to mine. Whilst not an essential feature in the app, a calendar provides an atheistically pleasing means to view the upcoming birthdays. To ensure the calendar is implemented properly, I had a look at google calendar and made note of the following:

* The calendar could be viewed in two different ways, horizontal swiping and vertical swiping.
* A specific day on the calendar could viewed and appended to by tapping on a date.
* Each month in the vertical scrolling view had a different theme to it, which makes each month stand out.



As horizontal swiping is being used as a method to navigate around the app, a vertically scrolling calendar makes the most sense to implement.

Wireframe

Composite

