345 Assignment 3 – Report

Overview

The purpose of this report is to serve the users of our application with information about the known issues, and our intentions to solve them.

Currently our application is not perfect, in fact, far from it. We have uncovered many problems after thoroughly testing it (post-submission), and have received a fair amount of feedback from users. We hope to clearly define the issues, which primarily fall under two categories - bugs, and usability problems. A bug is an unintended problem with the application, a problem we missed in testing, for example unusable buttons on the Android keyboard. Usability problems are a misunderstanding or a complete overlook as to what is best for our end-user, for example, swipes not obviously changing the view to indicate that the user has in fact changed to a different month.

The report is structured to firstly discuss what we believe are our current pros of our implementation, secondly to look at the issues we currently know exist from our testing, thirdly issues that have been raised via feedback from users and finally some ideas which we may use in regard to the problems outlined.

Current Usability

The frontend/GUI has the following functionally:

> For the month view, a user can select multiple days to add an event to. When the days are selected they appear grey, and a popup allows the user to enter an event, which contains a title, description, start time, and duration. Once the event is added, the user can select a day and they will get a list of all the events that have been added for the selected day. The month view also circles the current day, and allows the user to swipe back and forth between months.
>
> For the day view (which displays all a day's events) they can also add events, where the popup is the same as the month view. Aside from adding events, the day view also allows events to be deleted. In addition to the views, there is a toolbar that displays the current date, and allows you to switch between views as well as change the default colour of the toolbar.

The backend has the following functionality:

> The backend handles storing and retrieving events for a selected day. Event details are transferred from Java to C/C++ via a JNI call, where the events are stored persistently in an XML file. The current implementation uses the TinyXML2 parser to handle storing and retrieving the event details, which are stored intermediately in C++ classes that represent days and events.

Our overall backend design worked well, using an XML file is a completely valid approach, if you have a good XML document structure. Using JNI is also a valid approach to translating C++ to Java, with some base line understanding of both languages it is very intuitive.

For frontend, we found many ways to make the application more usable. We found a minimalistic approach made the application more intuitive and easy to use, as some other calendars (e.g. Apple calendar) have a cluttering of functionality which, to some extent, detract from usability and sometimes hinder design. Users reported that the multi-event touch was a very nice feature to the application, but was limited by the single month restriction.

Known Issues

Our application did not include all the functionality we had hoped when it was submitted. Due to the procedures involved in syncing a calendar with larger corporations such as Google, Facebook, and Otago University, we could not incorporate the sync functionality in time for submission. In month view the days with events were meant to be shown in a different colour.
We did not get this working bug free before the deadline so it was not included in the first deployment.

There are a few known issues with the app that were overlooked during final testing before submission. There is an issue with selecting multiple days when adding events. If the current day is selected to add an event to, it will not change its colour to grey, and therefore there is no way for the user to know whether the current date has been selected or not when creating a multi-date event. This issue has now been fixed and will be included in the next deployment. Another issue is that the action for swipe and long touch are easily confused by the device. When a user swipes left or right the system sometimes registers the action as a long click, resulting in the day being selected and the view being changed. To solve this issue, we intend to change the long click action to a simple tap, which is registered differently than the swipe.

There is an issue with tinyXML2 as it as lot of memory leaks which could make the app inefficient when run for a long time or have a lot of calls to the xml. We have found an alternative using SQL that appears to be free of memory leaks, which may yield better performance and to some people may be a more logical design.

Feedback

This section discusses some of the feedback from users, and will consider their opinion weighed against the opinion of all users of the application. Essentially, does their opinion hold any merit, or is it simply a one-time use case scenario by a single user. If feedback considers issues of the application rather than additional functionality this section will describe in detail the issue and will state how we intend to solve it.

Some users disliked the option of adding an event to a past date, we feel this is a useful option to have in case you are using the calendar as a log of where and what you have done. This kind of 'logging' is popular in newer generations as can be seen on platforms such as Facebook and Instagram where even their meals and thoughts are recorded/posted.

Many users would like an option during event creation for recurrence such as daily, weekly, monthly, yearly, or custom routine. We agree with the users that recurrence should be added at event creation but we also like the date selection process currently available.
Some users disliked the event information displayed in the day view, too much information was displayed and should be simplified. Also, the count started from 0 which is not common practice for general users. This will be included in our UI rework for an even simpler layout.
A major concern from all users is the lack of notifications for upcoming events. This is a feature we hope to include soon, but currently we are having issues supporting this feature on all devices.

It is worth noting that some users had difficulties using parts of our application, but keep in mind that some of this can be attributed to their understanding of functions and operations of the application they are familiar with the most. In our attempts to be original we have attempted to detract from this approach and copy less from what we see in other applications and rather use that as an example of what we should improve upon. However, there are some instances where it is more understandable to have the common practice feature, i.e. swipe to change month.

Improvement Intentions

We plan to incorporate recurring events into the app for the next version. We also plan to change the event creation system so that all events are created from the same menu and the event date is input by the user. We are planning to attempt the sync functionality again, though if the process is not complete with enough time to incorporate the function we will remove the option all together.

Changes to frontend/GUI:

We also would like to change the UI to look less generic, but still very minimal and simple. One change we are making is the displaying of events in the month view, so that the you can view events on the view without having to further click on the day to find out what it is.

We will add a go to current date button, so if users are looking at a date a few months ahead of time they can go straight back to the current date. Additionally, a go to specified date button, so that users can look at what they have planned to do without having to swipe through the months to get there.

Changes to backend:

Due to the memory leaks associated with TinyXML2, we plan to change the application's internal storage from xml. We will instead use SQLite3, and store our data as a database file.

We will make more efforts to make the application more usable as any developer should. Any opportunity we see that may improve the usability we will first test with users, and then implement to avoid further hindering application usability. In the previous submission we simply put forward what we thought would improve usability without consulting the end-user.

Figure 1 (see Appendix) describes the series of tasks we will undertake, and the expected dates that we will have completed these tasks. Notably these dates should not be treated as exact dates as it is difficult to define how long a task will take. We have estimated the expected amount of time, but this time estimate could be overestimated or underestimated. Tasks that are shown in parallel with other tasks will be completed by different team members or a sub-team of team members so they can focus purely on the task they are allocated.

Summary

The current state of the application is very usable, though we see many opportunities for improvement. The prevailing consensus suggest that the problem with our application lie more so on the side of missing functionality. In the way of bugs, our application does have a few, but nothing remarkably huge, e.g. event number starting at 0, which does not hinder how the application functions, but is just an unintentionally missed detail. Missing functionality is more about functions that further improve functions, e.g. a "Go back to current date" button when we implemented the swipe to change month function. It's essentially another layer which makes it more usable.

While our app is not the popular calendar app like the natives for Samsung and Apple, we hope to make an app that has some merit to users. Though it is incredibly unlikely that this app will in any way, shape, or form be a competitor, it may have some functionality that these native calendars may have missed the mark on. We hope to have some features that may bring into question the usability or functionality of these native applications. With an application that comparatively has not nearly as many resources invested into it, we can only pray that the ideas we have or will have were not already conceived by the paid developers at Samsung or Apple. This is not to say that we have given up, but rather we will persist through development and user feedback in the hope that we are able to satisfy many users.

Appendix

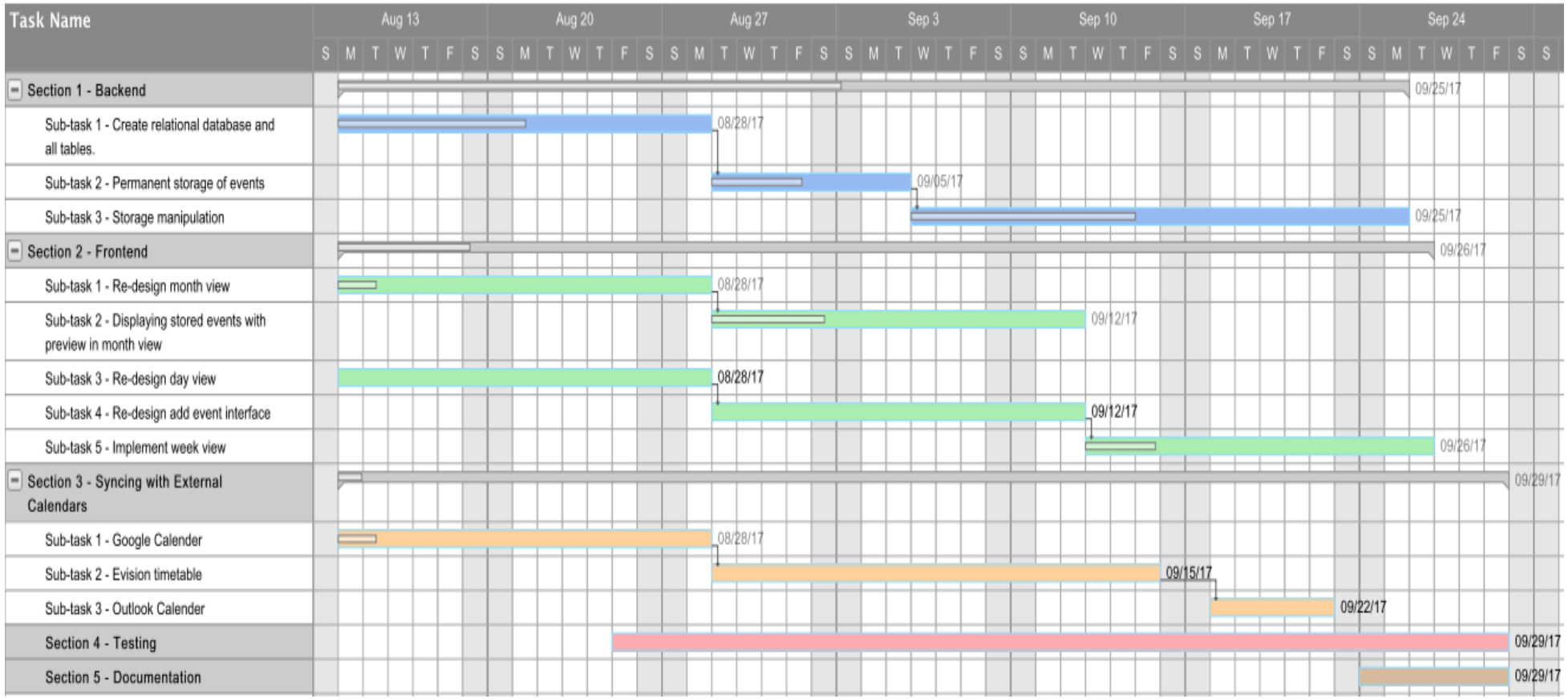

Figure 1 – Gantt Chart of project schedule