

iManager

Mexiang Jin (m38jin), Yi Ren (y63ren), Ziniu Wu (z82wu)

What is your project?

iManager is a set of small “fridge stick” components each has its own functionalities such as weather, alarm and calendar. The user can access and modify each component individually which replace the need of opening numerous other apps; furthermore, the user can even customize their “fridge” by adding, removing or moving around stickers to fit their need.

Why is it useful/interesting? Describe and justify your project selection.

Nowadays, we can find a lot of APPs for checking weather, setting alarm or making to-do list etc. In each APP, there are lots of functionalities; however, users usually need to switch between several APPs to satisfy their daily usage, which is kind of inconvenience. What we want to do is to combine multiple useful functions so that users can be satisfied by only opening our APP. Other than the usual functionalities others have, we are going to add some useful tricks to convenience our users. Our goal is to help our user to have a better life. They only need to focus on the important things. Unimportant matters can be handled by our APP.

Why does this project make sense in a mobile form factor?

In recent years, people are more likely to use mobile devices, which give the convenience to work anywhere, instead of computers. As a result, if we can build our project on mobile devices, it should be more popular and get used by more people. Most importantly, iManager as a life managing app, users need to keep it around and this can't be accomplished by using computers. Moreover, iManager doesn't need that much memory, so it can run perfectly on mobile instead of computer. Therefore mobile is the best choice, since it's more convenient and can access internet without connecting wifi.

Functional properties:

1. Mainframe:
 - 1.1. iManager should provide users the ability to add/remove and drag “stickers” around.
2. Alarm:
 - 2.1. The user can choose the time they need from waking up to getting ready to leave. Then the alarm will set a suitable alarm to wake the user up.
 - 2.2. The user can directly set the time alarm rings like normal alarms.
3. Weather:
 - 3.1. The weather part can get user’s current location and show the weather.
 - 3.2. Additionally, the weather part can give suggestions of what to wear today according to the weather.
4. Calendar:
 - 4.1. The user can update their class schedule and other important events to the calendar.
 - 4.2. iManager will store if the user owns a car to determine the time for moving in calendars.
 - 4.3. According to the time the user needs to get to next event, iManager will send notifications to the user in three modes:
 - 4.3.1. Default mode: iManager the calculate the time user needs to take from the location of the last event happened or last time the user opened iManager (choose the closer one). And send a notification to the user, when they need to depart to next event.
 - 4.3.2. Fixed mode: the user can choose when to send notification in a fixed time before an event.
 - 4.3.3. Fixed before mode: the user can choose when to send notification before an event. The notification will contain when to depart to next event according to the current location at that time.

We suppose that the user’s name is Jack. He has a car and he has already updated his class schedule and car ownership.

Non-functional properties

- Scalability
 - Since the iManager would run multiple sub-apps simultaneously, it is important to make sure the iManager would still have a smooth interaction even if the user has a number (still reasonable) “stickers”.
- Equal Performance
 - Since the iManager provide “stickers” like weather, calendar which has dozens of similar applications in Play Store. The “stickers” in iManager should at least, in some aspects, have the same level of functionality as their counterparts. Otherwise even the iManager integrated all of these into one, nobody is going to use it as the “stickers” have to few functionalities themselves.
- Intelligent management
 - The users don’t want to be notified about what to wear for next months’ event today, nor would the users need notifications about they should leave for class 12 hours ahead, the iManager should have the functionality to not prompt these “spam” notifications, but rather in a reasonable time slot.
- Ease of Use
 - In iManager , we provide multiple frames that each has its own function, within the frame, it will show the images that show in normal life. For example, for alarm frame, it will show the image of alarm that most people will recognize. For simplicity of using, we will provide symbol or button that most user will recognize. If don’t, we will also provide a help or FAQ section that can help users to know iManager better.

Interaction scenarios

One night, Jack wants to set an alarm to ensure that he can catch tomorrow’s morning class at 8:30. He knows that he needs 40 minutes to get ready to go to school; therefore, he opens iManager, goes to the alarm part and sets his prepare time to be 40 minutes. iManager calculates that he need 15 minutes to reach his classroom. As a result, iManager automatically sets an alarm of 7:30.

On the next day, Jack goes to the class on time and gets a good sit in the second row, the time calculation from iManager was perfect. He is so grateful that iManager reminds him to wear sweater and coat due to a sudden drop in temperature. His friend John complains a lot about the bad weather. Obviously, Jack is not affected.

After all the classes today, Jack go to Tim Hortons to have a rest. Suddenly, he gets a notification, which remains him to go to the appointment with his dental doctor. According to iManager’s calculation, he may be late, if he doesn’t depart in 5 minutes. “Thanks iManager, I need to wait so long for next appointment if I miss this one.” One sound arises in Jack’s mind. With iManager’s help, Jack had a wonderful day.

Mockup

