Introduction:

Our timetable app is committed to providing users with a simple, practical, and efficient time management tool. It helps users arrange their daily life, work, and study affairs reasonably, improves time utilization efficiency, and achieves a convenient lifestyle. Our app can use in calendar view which means provides information on the Gregorian calendar, lunar calendar, and holidays, allowing users to quickly understand the date and related festivals, time planning which means supports schedule planning and to-do list management. Users can customize event reminders to ensure important matters are not missed and suitable for various scenarios which means whether it’s daily life, work arrangements, or study plans, the small calendar app can provide you with considerate time management services.

Requirements Specification:

As a timetable app, including many user requirements. Such as conveniently view the current date and future dates, manage daily schedules and to-do items, set reminders to avoid missing important events, simple and easy-to-use interface and save mobile storage space and battery life. There are also many functional requirements, such as calendar view including display Gregorian and lunar dates, highlight holidays and support switching between month, week, and day views, schedule management including add, edit, and delete schedule events, set recurring events, such as weekly meetings or daily alarms and categorize events, such as work, personal, entertainment, reminder function including set event reminder times, support multiple reminder methods, such as notifications, ringtones, vibrations and reminder summary for a quick view of all reminders for the day, user experience including interface design that is simple and intuitive to operate, support gesture operations, such as swiping to switch dates, multilingual support and performance optimization including optimize memory and battery usage, fast startup and response.

Overall Design:

Structure:

First, our project will have several interfaces connected, so we designed two large folders, namely the library folder and the sample folder. The library folder stores content code and data, while the sample file stores execution code to call the class or function of the file in the library folder. Each interface corresponds to a code file.

For example: our project provides users with three calendar marking methods, namely single day marking, multi-day marking and marking within a certain range of days. They correspond to different interfaces respectively, so the execution code provides jump functions of the code files corresponding to these three marking methods.

Primary function:

Users can choose multiple marking methods to mark the calendar to help them remember important days.

User Interface Design:

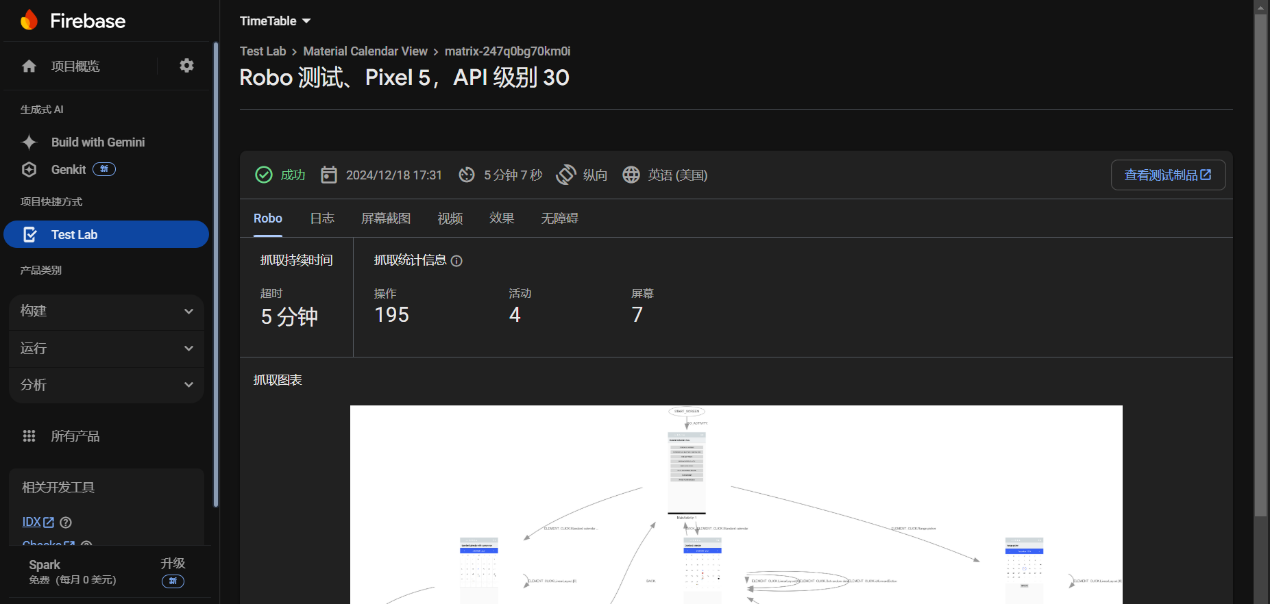
There are only two people in our team, and we put a lot of energy on the realization of the main functions, so we didn't put too much effort on UI design. Despite this, we are still trying to make our interface look better. For example, we have tried to design the size, position and color of each button as beautiful as possible under the premise of simple completion. As for the design of the calendar, we refer to many calendar templates that have been uploaded on GitHub.

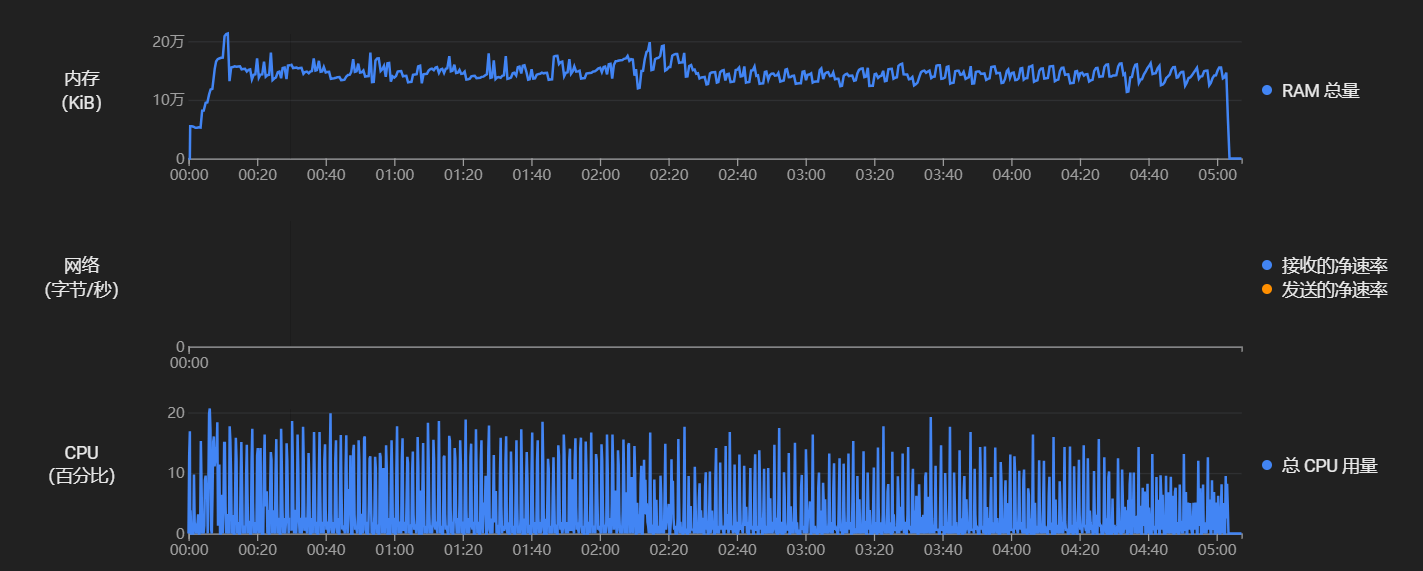
Key Technologies:

Our project mainly relies on the Calendar View component that comes with Android Studio, which can already realize the basic calendar display. We searched for the usage of this component on the Android Developer website and tried various methods on this component in our own code. However, our project has many other functions that this component cannot do, such as marking the days in the calendar. We need to find ways to implement many functions and transplant them into our project. The biggest obstacle we encountered was that when implementing these methods, the packages and components involved were incompatible with our project. For example, a certain package requires gradle version 6.0, but the gradle version in our Android Studio is the latest, which makes this package unusable due to different gradle versions. Of course, there are solutions to all problems. In the end, we searched for solutions to different problems on the Internet.

Testing and User Experience Analysis:

Here are some figures about the data of testing on the third-party cloud platform.







As the result, here are two main problems, that are optimization issues which means the memory usage and CPU usage are too high, far exceeding our expectations and accessibility issues which means some icons are designed too small, which is not suitable for accessible use.

Here is a figure about the feedback and suggestion of other classmates.



As the result, most students agree with the minimalist style, but inevitably there will be people who want to use their favorite pictures as the interface. Students have given a lot of suggestions on functional expansion, including but not limited to displaying to-dos on the lock screen and inheriting calendar data used by previous users. Our installation guide and compatibility are still inadequate.

Conclusion:

All in all, as a schedule management app developed by us, aiming to provide users with a simple and efficient time management tool to help them arrange their lives, work, and study. The app includes calendar view, time planning, schedule management, reminder functions, and more, suitable for various scenarios. About app’s requirements specification, including user requirements, functional requirements, interface design, performance optimization, and more. We used the Calendar View component built-in with Android Studio and overcame compatibility issues to implement calendar marking and other functions. Test results show that the app has problems such as high memory and CPU usage, accessibility issues, and interface design that needs improvement. User feedback indicates that most students agree with the minimalist style, but some also want to add personalized features such as custom pictures, and suggest functional expansions such as displaying to-do lists on the lock screen and inheriting calendar data used by previous users. We plan to improve these issues in subsequent development and further enhance the app’s functionality and user experience in the future.