

The tutorial for making WeChat small programs

In this year 's, team BIT project built an interactive communication system based on WeChat mini-app. Through our WeChat min-app, we can realize an instant detection operation platform based on smartphones that can support biosensor modules and sliding chip modules. In this process, we met many difficulties and doubts, but on the basis of drawing on a large number of existing information and sharing with the previous team, we finally got a complete and accurate interactive communication system through continuous exploration and improvement. WeChat mini-apps are widely used in China and many iGEM teams have ideas or have applied them. In order to enable other teams to obtain experience from our experience, we will share relevant research, ideas and technologies in the process of building the mini-apps, which can be used as a reference and help for other iGEM teams in the future.

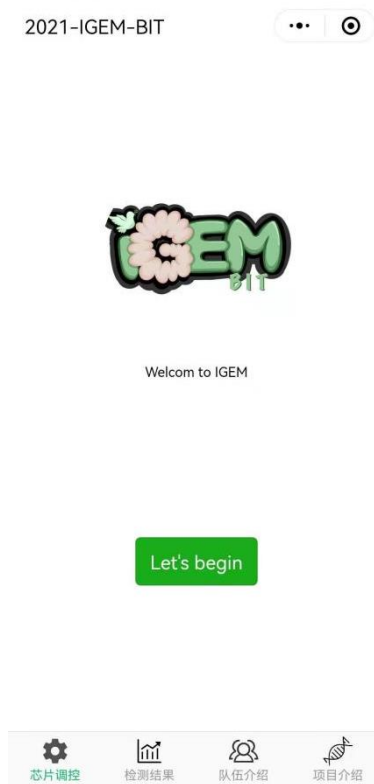


Fig1. Our WeChat mini-apps interface

Then we will introduce the related technologies of WeChat mini-app, the design method of the detection system, the implementation method of the detection system and the deployment and testing process of the detection system. This can guide and reference the future iGEM team by showing the main processes and methods of building WeChat mini-programs.

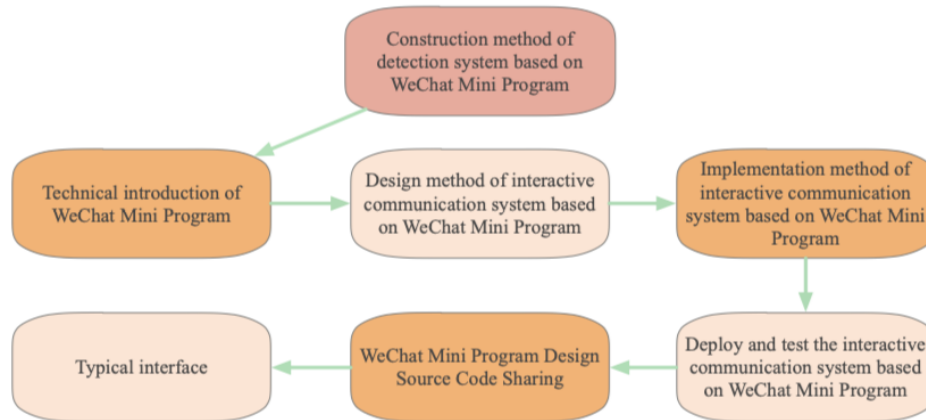


Fig2. Steps to build WeChat min-apps

1 Technologies introduction of WeChat mini-apps

1.1 Introduction of WeChat Mini Program

WeChat mini-app is a new way to match users and services. It can be easily obtained and disseminated within WeChat and it does not need to download and install. It can be used only by scanning QR code or searching, which has a fantastic use experience. At the same time, WeChat mini-apps provide a simple and efficient application development framework and rich components and APIs to help developers develop services with native APP experience in WeChat.

WeChat mini-app is composed of five parts: namely APP, window, page, component and routing. It has the advantages of easy to use, simple development, cross-platform operation, rapid distribution and rapid iteration. Its positioning is very clear: make an application with low frequency and uncomplicated business logic that complements native apps. WeChat mini-apps have their own set of development code and are written through WeChat developer tools.

1.2 WeChat Developer Tool

In normal web browser, when loading the html5 page, the html and css of the view layer are loaded first and then the javascript of the logic layer is loaded. Finally, the data are returned and the page is displayed in the browser. The system layer of WeChat developer tools is based on Native System so the view layer and logic layer will be loaded at the same time. This logical way of WeChat mini-apps greatly optimizes the page response speed, reduces the waiting time for page loading and improves the user experience. WeChat developer tools can implement a complete set of processes such as synchronous local files, development and debugging, compilation preview, upload, and release.

1.3 WeChat Mini-apps Framework Construction

The whole mini-app's framework system is divided into two parts: the logic layer (App Service) and the view layer (View). The mini-app provides its own view layer description languages WXML and WXSS, as well as the logic layer framework based on JavaScript. In order to make developers focus on the logical level of code and data processing, it also provides two solutions: one is event and the other is data transmission. These two schemes are located between the view layer and logic layer of the mini-app's framework. After the user interacts at the view layer, the data is processed at the logical layer and returned to the view layer for feedback. For the development and design of the view layer of WeChat mini-apps, the mini-apps officially provides a series of components, which can greatly help developers to complete the development of the view layer. At the same time, for

logic layer development, small programs officially provide a series of APIs to facilitate developers requesting additional functionality.

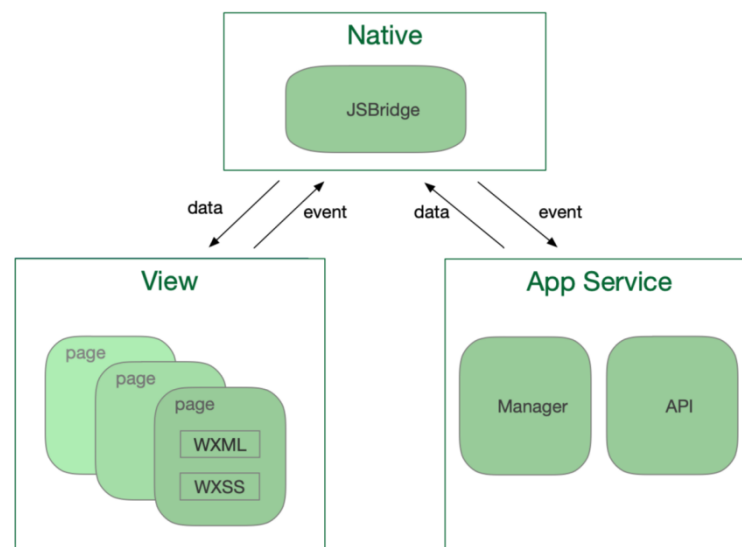


Fig3. Framework System of WeChat Mini-apps

2 Design Method of Interactive Communication System Based on WeChat Mini-apps

2.1 Demand Analysis of Interactive Communication System Based on WeChat Mini-apps

Before the implementation of the system, we first carry out the demand analysis. We need to consider the popularity of WeChat and the portability of the WeChat mini-apps. And also we need to make full use of the convenience and popularity of the mini-apps compared to the web page, so that the interactive communication system based on the WeChat mini-apps can realize the functions required by the project.

2.2 Functional Design of Interactive Communication System Based on WeChat Mini-apps

We recommend that the functions should be enumerated according to the needs of the team project and the functions that need to be implemented. Then the functions should be explained in detail. For example, the overall function to be realized in the WeChat mini-app of our project is divided into two main parts. On this basis, we further refine the function, design four main functions and consider the design that all the four functions need.

3 Implementation Method of Interactive Communication System Based on WeChat Mini-apps

In the third section of the software section, we have designed several functional modules in detail and these functions should be implemented below. We hope to use our practical application process this year to provide reference for other teams.

3.1 Implementation Steps Introduction

The realization of interactive communication system is mainly divided into system framework-view layer design--logic layer design.

3.1.1 Introduction of System Framework

The system framework design is mainly based on the functional design of different main columns in the above functional design. The corresponding file framework is constructed, and the wxss, js, json, wxml files of each page are constructed and initialized. JSON is a data format, not a programming language. In Mini-apps, JSON plays the role of static configuration. Programming uses a combination of WXML + WXSS + JS. They are used to implement different functions: WXML is used to describe the structure of the current page; WXSS is used to describe the appearance of the page; JS is usually used to deal with the interaction between the page and the user, respond to the user 's clicks, access to the location of the user and so on. In the mini-apps, the operation of users is processed by writing JS script files.

3.1.2 Design Method of View Layer

The view layer design needs to design different columns according to the content to be displayed on each page, taking into account the performance of pictures and words. At the same time, considering that the mini-app is limited by the page size compared with the webpage, the reasonable column design will help users to quickly browse the corresponding content. In addition, developers need to consider ways to implement view layer design. For example, our project divides the main interface of the mini-app into four modules according to functional requirements, and realizes the conversion of four different functional pages respectively.

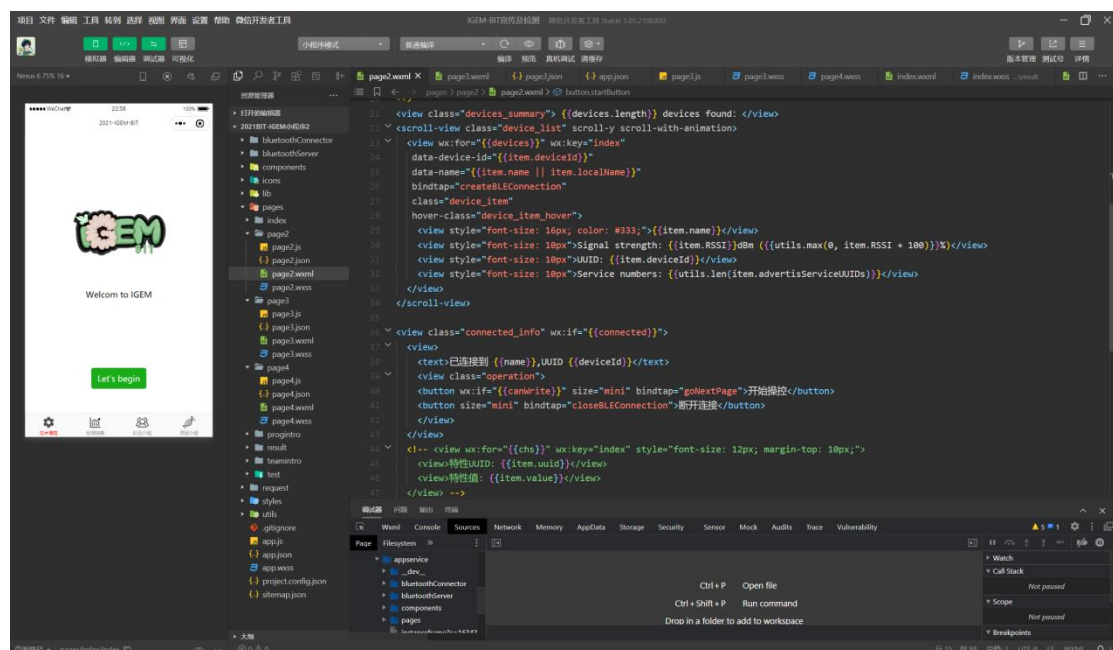


Fig4. Example of WeChat mini-app's interface implementation code

3.1.3 Introduction of Design Method and Logic Layer

The logic layer uses Js Core thread to run JS script. There are multiple interfaces in a mini-app, so there are multiple WebView threads in the rendering layer. The communication between these two threads will be transferred through the WeChat client and the logical layer sends network requests through the WeChat client.

The WeChat client will download the code package of the whole mini-app to the local before opening the mini-app. Then all page paths of the current small program can be known through the

pages field of “app. json”. Therefore, the WeChat client loads the code of the home page and renders the home page through some mechanisms at the bottom of the small program.

Page in Js file is a page constructor, which generates a page. When the page is generated, the mini-app’s framework renders the final structure of data and “index.wxml” together, so the appearance of the small program is obtained. After rendering the interface, the page instance receives an “onload” callback, where the applet can process logic.

3.1.4 Introduction of System Function Realization Process

In order to enable developers to easily adjust the ability provided by WeChat, the mini-app’s development framework provides rich WeChat native API, which can easily adjust the ability provided by WeChat, such as obtaining user information, local storage, payment function, etc.

For Bluetooth, mini-apps provide Bluetooth capabilities of each platform to developers through unified interface encapsulation. Using the Bluetooth interface of small programs, data can be exchanged with other Bluetooth devices by wireless means.

In mini-apps, to use Bluetooth capability (except Beacon), the Bluetooth adapter module must be initialized by calling “wx.openBluetoothAdapter” first, and its effective period is to call “wx.openBluetoothAdapter” until “wx.closeBluetoothAdapter” or small program is destroyed. Only during the mini-app’s Bluetooth adapter module takes effect, developers can normally call the Bluetooth-related mini-apps API and receive Bluetooth module-related event callbacks (binding listening is not limited by this).

Each Bluetooth peripheral has a unique device-Id to identify. Due to some system implementation limitations, for the same Bluetooth peripheral, the device-Id acquired by scanning on different central devices may change. So device-Id can not be hard-coded into the code, only by the user to choose.

After the Bluetooth connection is successful, “wx.writeBLECharacteristicValue” is called in the effective cycle to send data.

4 Deployment and Test of Interactive Communication System Based on WeChat Mini-apps

4.1 Complete System Deployment

It is very important to deploy and test the system in the construction of WeChat mini-app platform. When our system is developed and implemented in the local development environment, the system background is deployed to the local server for certain debugging and testing. In this section, it is best to deploy the system based on its characteristics, such as whether it needs to be deployed to a cloud server.

4.2 System Testing

System test should be carried out after the system is built. System testing can be divided into functional testing, performance testing and compatibility testing. Functional test mainly verifies whether each functional module in the system can operate normally, and assembles each module of the system to verify the complete operation of the system. Performance testing mainly tests the friendliness of the system interface, the fluency of page loading, and the occupancy of device resources when the system is running. Compatibility test is to test whether the system can run normally on each mobile phone platform.

4.2.1 System Functional Testing

The test can be divided into different classes according to different angles. For different stages of development, the system test can be divided into unit test, integration test, system test and acceptance test. And the test can be divided into static test and dynamic test according to whether it is running.

Static testing is to check the correctness of the system by analyzing the syntax of the source program. In the process of dynamic testing, unit testing is used to test all the functional modules of the system one by one to ensure that each module can run independently. After that, the system is integrated and tested to ensure that all functional modules can work together without any conflict.

4.2.2 Perform System Performance Testing and Compatibility Testing

According to the mobile terminal equipment used by the target user, different types of mobile phones should be used to verify the display effect of the system and whether the function of the system is normal and whether the operation is smooth. There are many things should be considered: whether the test can run well and smoothly on different devices, whether the adaptive layout can be carried out according to the resolution of the device and the size of the screen, whether the user experience can be guaranteed and whether the various functional modules can be used normally, etc. For mini-apps that adapt to multiple operating system systems, it is necessary to test whether they are not limited by the device operating system.

5 Mini-app's Typical Interface

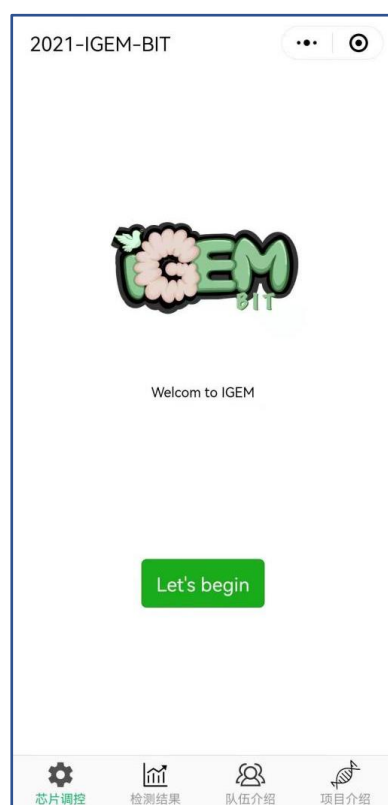


Fig5. The interface of start testing

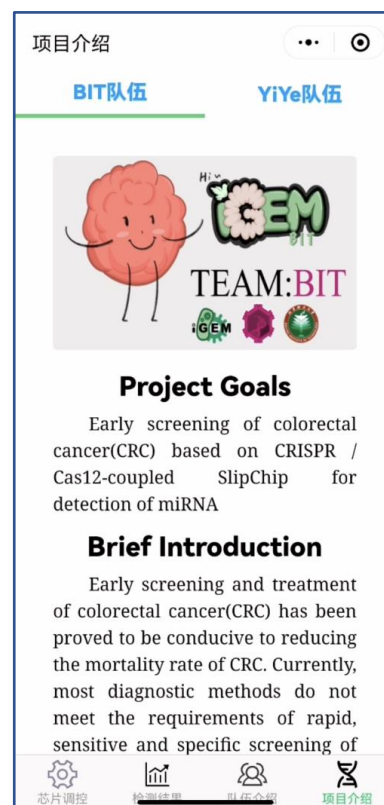


Fig6. Interface of project introduction

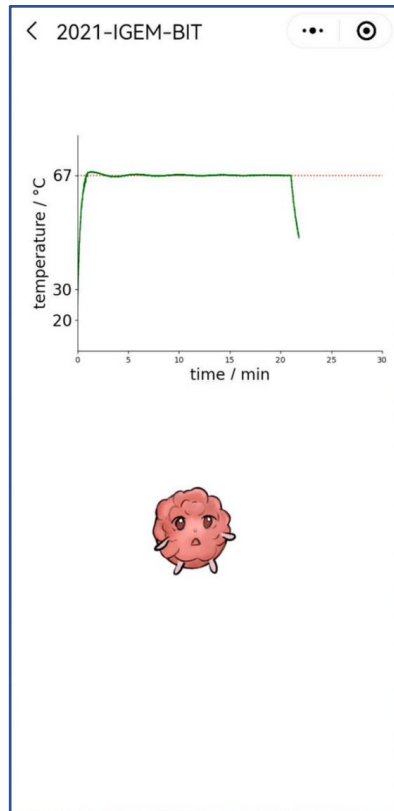


Fig7. Interface of LAMP Temperature

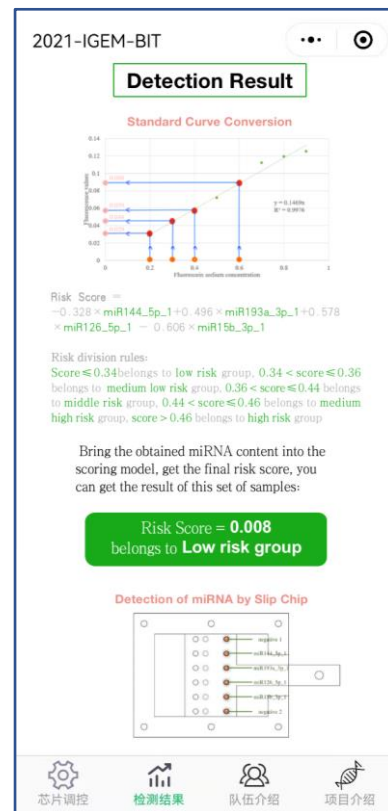


Fig8. Interface of test results

Control

6 Source Code Sharing of WeChat Small Program Design

In addition to the above description, we open source the code related to the development of the entire WeChat mini-app on the GitHub platform for the download of the required iGEM team.

The URL is: https://github.com/jank83667/IGEM_BIT