University of Nottingham
UK | CHINA | MALAYSIA

COMP2043.GRP Final Group Report

Project: A Software Tool for Learning Tang Poetry

Team Identifier: Team 10

Supervisor: Heshan DU

Group Member: Zhiyu ZHAO(slyzz6), Shengyi ZHOU(scysz1), Xinyang LIU(scyxl6), Yunpeng GUO(slyyg3), Zhongyi WANG(scyzw2), Mingjia ZHANG(scymz3)

Date: April 23, 2020

School of Computer Science

University of Nottingham Ningbo China

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Tang poetry is a treasure and carrier of heritage of Chinese traditional culture. Today there are more than two thousand famous poets and nearly fifty thousand well-preserved poems (Hou and Frank, 2015). Tang poems cover a wide range of themes. Some reflect the class contradiction of society and reveal the darkness of the feudal society. Some criticize wars and express patriotic thoughts. Some may describe the beautiful mountains and rivers of ancient China, and there are also poems expressing personal experiences and emotions of poets such as joys and sorrows of life, the wanderer's longing for home, the frustration of unable to realize dreams (Owen, 1981). In this part, the motivation of this project and the objective will be introduced.

- **Motivation**

  If searching by keywords "Tang Poetry Learning Tools" in Baidu, Bing or Google, there is many software tools for Chinese children to learn Tang poetries whose interface uses Chinese. However, there is few existing software whose target users who are native English speakers. Therefore, the team decides to develop a Tang Poetry learning software which aims at helping people whose first language is not Chinese to acquire and deepen the understanding of Tang poetry; thus, to experience the charm of Chinese traditional culture.

- **Aims and objectives**

  Existing Tang Poetry learning tools have some characteristics in common. User-

friendly interface with an ink painting style background, which also serves as an element of Chinese culture, would correspond to the core of the software. Apart from this, other essential functions for a learning tool are also provided, such as some annotations and exercises for reviewing purposes. Nevertheless, for non-native speakers, interface with only Chinese could be beyond their abilities. There are also some other issues need to be considered. For instance, Chinese children start learning Tang Poetry from an early age, but this is not the same case for foreigners (Li, 2008). A basic and general introduction is necessary to help the target users develop a general understanding of Tang Poetry. Besides, how to motivate them to enjoy learning Chinese culture is also a question. Existing learning tools always have a comprehensive analysis of Tang Poetries but not showing them in an interesting way. Therefore, some gaming parts could be added as a solution to make the learning process more enjoyable.

The rest of the report is structured as follows: Chapter 2 describes the background information. Chapter 3 describes the marketing research. In Chapter 4, requirement specifications will be presented. Chapter 5 describes the design of this Tang poetry learning software. Chapter 6 is the implementation of the software and specific codes will be provided as well. For Chapter 7, an evaluation of the design will be illustrated from both application aspects and statistic aspects. Chapter 8 is about what was achieved according to the requirement specification. In Chapter 9, an overall reflation will be presented. Then it comes to the reference list. The bibliography of what is referenced during the whole developing process will be shown. Finally, an appendix including test cases, meeting minutes, a consent form and a user manual of this software.

Chapter 2

Background Information and Research

There are four sub-parts in the background information and research part. In the first part, the survey of existing software systems will be shown. The second part is the results of the interviews, then followed by marketing research. Last part is the survey of the tools used in the application.

2.1 Survey of the existing software systems

Because the aim of the program is for users to develop a basic understanding of the Tang poetry, the team decided to look for existing Tang poetry learning software on iOS, Android and Websites. The following contents are existing software systems found by the team. The following contents respectively show the detail of Xi Chuang Zhu, Three Hundred Tang Poems and Kai Shu Story. A summary of these software will be provided at the end of this section.

2.1.1 Xi Chuang Zhu

The first software is called Xi Chuang Zhu[1]. It is an application for both iOS and Android platforms. The major function is poetry reading. Its main page (Figure 2.1) is designed

[1]http://www.xcz.im/

in the format of poem cards. Users are able to swipe left or right to switch between each sentence (Figure 2.2).

Figure 2.1: Main page of the application

Figure 2.2: Switch contents

It also supports users to search poems by keywords (Figure 2.3) or categories (Figure 2.4).

Figure 2.3: Search by key words

Figure 2.4: Search by categories

The application supports online games as well, such as poetry solitaire (Figure 2.5) or coming up with the sentence with the specific word (Figure 2.6).

Figure 2.5: Poetry solitaire

Figure 2.6: Poetry sentence riddle

2.1.2 Three Hundred Tang Poems

The second software is Three Hundred Tang Poems[2]. In the reading part, when users choose to select a poem and enter the poem page, the software will show the translation from the ancient Chinese to the contemporary Chinese language, which is understandable by the modern Chinese (Figure 2.7). Also, Pinyin and recordings are provided for users to read after it. Learners can simply press the AI reading button in the rightmost of the bottom bar to realize this function (Figure 2.8). The font size customization button is provided in the up-right corner as well.

[2]http://www.appchina.com/app/org.vv.tang300

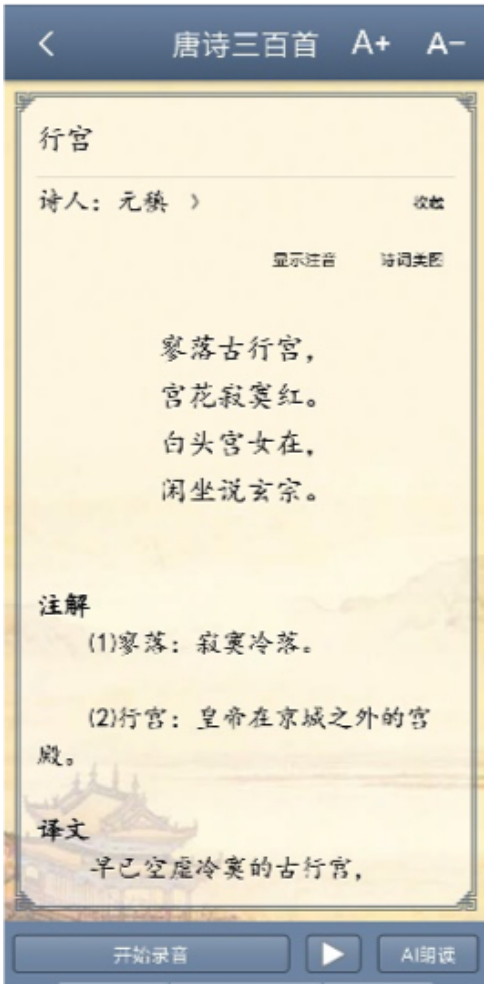


Figure 2.7: Translation

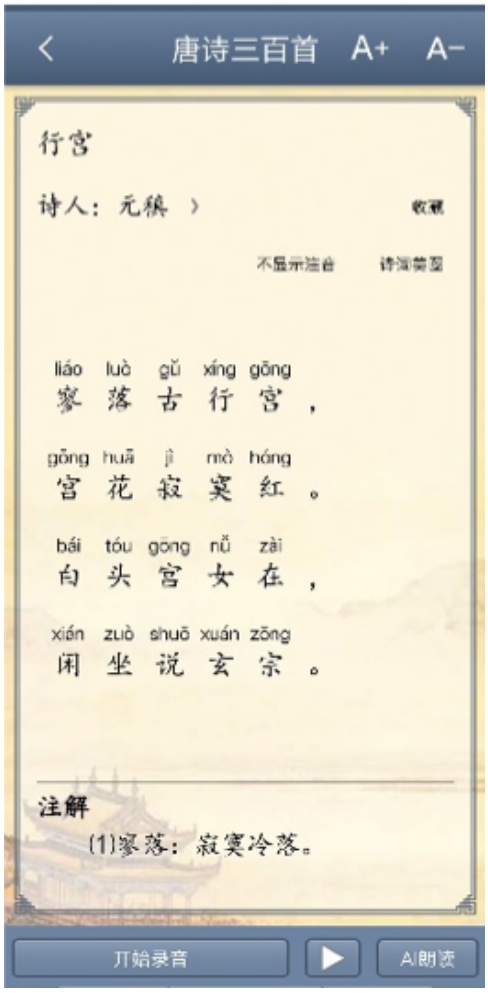


Figure 2.8: Pinyin and Buttons

Another attractive highlight in the software is the daily poetry function. Once the user opens the application, the software will recommend daily poetry randomly (Figure 2.9, 2.10).

Figure 2.9: Daily poetry on 10.19

Figure 2.10: Daily poetry on 10.20

Moreover, the software supports users to take tests themselves. It contains five games in various level and formats where users can select one and consolidate their mastery of previous knowledge (Figure 2.11).

Figure 2.11: Test for users

2.1.3 Kai Shu Story

Kai Shu Story[3] is a story-telling application for mobile devices. The main feature of the application is audio playing function. The graphic user interface is similar to the music player, which has functions including downloading, history, favorite and commenting module (Figure 2.12).

Figure 2.12: Story player interface

The commenting part is online, so users could leave the comment below and share the point.

2.1.4 Summary of the above software

The above three software are existing ones for learning Tang poetry, and the functions of those applications are worth developing. Although the software is nicely designed, none of the above provides both Chinese and English versions of Tang poems. Hence, they are

[3]http://www.kaishustory.com/

not suitable for the people whose first language is not Chinese. The software which we are developing contains some of the functions mentioned in the previous part, including daily poetry, keyword searching, translation, and games. Additionally, our software provides translation from Chinese to English, which is the function none of the above software has. Another innovation is that apart from self-tests, the application provides two own games for users to play, which are called “apple tree” and “tang story”. It is reported that people who are learning a new language need to be provided with a culturally relevant curriculum (Vega et al., 2019), which will be provided by our software.

2.2 Result of Interviews (consent form, question sheet, invitation)

As the target users of the software are those whose first language is not Chinese, to understand their preferences, the team has composed a list of interview questions. It is reported that interview studies could provide a better response rate (Tracy, 2019). The following are questions and feedback.

Before question: Brief introduction about the project and Tang poetry.
Feedback: This part is the introduction of our project and Tang poetry. The interviewees showed great interest in this project after the introduction part. Then the team showed a sample of the poem. Interviewees were amazed by the coincident rhyme of the poetry but had no great interest in the content.

Question 1: Do you know about Tang poetry?
Feedback: After the introduction, the team first chose to ask if the interviewees had learned about Tang poetry before. The feedback showed that the interviewees had completely no previous knowledge about Tang poetry.

Question 2: Have you ever studied Chinese? Please give us more details, such as how long have you studied? And your experience of studying Chinese.
Feedback: The interviewee claimed that their ability to use Chinese was very limited and had taken a Chinese class module for a semester. After then the interviewee was able to remember approximately 50 Chinese words. For the experience of studying Chinese, the interviewee explained that the most difficult part was the pronunciation part because of some specific pronunciations such as “zh” and “ch”.

Question 3: Are you interested in tang poetry? Which part do you interest in?
Feedback: The interviewee showed little interest in Tang poetry and the most interesting

part for him was the reading part, while writing would be more difficult and boring.

Question 4: If there is a software to learn tang poetry, do you want to use it on the phone or the computer?
Feedback: The mobile version was mostly preferred, i.e. using the software on mobile phones or tablets.

Question 5: By what means do you hope to learn Tang poetry? (video, audio, games, exercises, . . .)
Feedback: The video and audio parts were the patterns that mostly wanted. It was the best way for him to associate the words with visual contents.

Question 6: What functions do you expect the software have?
Feedback: The required function turned out to be the rewind function, which could be useful for the user to listen to the audio repeatedly. Besides, both the English and Chinese versions of the interface should be provided. Thus, the user could learn both using English and Chinese.

Question 7: What would you like to learn from the softwares? Such as knowing the meaning of some tang poems, understanding the history of tang dynasty, writing tang poems, reading tang poems, and so on.
Feedback: The interviewee hoped to attain the history about the dynasty and the background story behind each poetry.

Question 8: How much time would you like to spend on learning tang poetry?
Feedback: The time differed between each one, from 20 minutes to a few hours in the evening. It depended on the interviewees' interest in poetry.

Question 9: Do you have any questions/suggestions?

Feedback: The main point was to keep the software interacting with the user and interest users to the greatest extent.

Chapter 3

Marketing Research and Implementation Decisions

Market research is carried out from the perspectives of platform and development language to figure out which type of device is the most prevalent version among Tang poetry learners.

3.1 Platform

- **Device**

  The main devices that used by most users are mobile phone and personal computer (Watts et al., 2013). The target users are non-native Chinese speakers. According to the interview results, most interviewees prefer to use mobile phone applications to learn Tang poetry rather than use a computer. The reason is that using a mobile phone is more convenient and people can have better use of fragmentation time without the restriction of location. As a result, the application will be developed on a mobile phone.

- **Operating system**

  According to Chadha et al. (2017), dominant operating systems used on smartphones are Android and iOS. Market research was conducted to understand which

operating system has more users. According to Fund et al. (2017), the global market share of the Android phone was 85.1%, which means the Android user group was a much larger audience of mobile users. Based on the research, the application will be developed on the Android platform.

3.2 Programming language

Android applications can be developed using Java, C, and C++. C language is a process-oriented and abstract general programming, which is widely used in bottom-level development (Ritchie, 1993). C++ is the inheritance of C language. It can be used for procedural programming of C language, object-based programming characterized by abstract data types, and object-oriented programming characterized by inheritance and polymorphism (Stroustrup, 2000).

In this project, Java is used as the programming language. There are two main reasons why Java is the best choice. First, the characteristic of the Java platform allows it to run on any computer with JVM (Java Virtual Machine)(Liang, 2009). It allows the programmers to write codes based on the various underlying hardware. Compared with Java, C and C++ will compile each software into versions that are suitable for different models. Second, Java can realize modularization. The learning Tang poetry software will be implemented with several different functions, and each function can be divided into modules for development. Using Java not only increases the efficiency of development but also reduces the difficulty of software maintenance in the future as requirements changing and functions elaborating.

3.3 Existing Tools

To realize some of the main functions which require operations on remote server, third-party tools are essential. The following parts are related researches and the introductions of some exists tools including XAMPP, NETAPP, MySQL and Volley.

- **XAMPP**

  XAMPP was used to build a local website and storage of data sources for the media player (Dvorski, 2007). MP3 files are stored in the sub folder of 'htdocs' so that they could be accessed by the corresponding URL of the local website. Now the URL is "localhost:8081/music/Sol.mp3" (Figure 3.1).

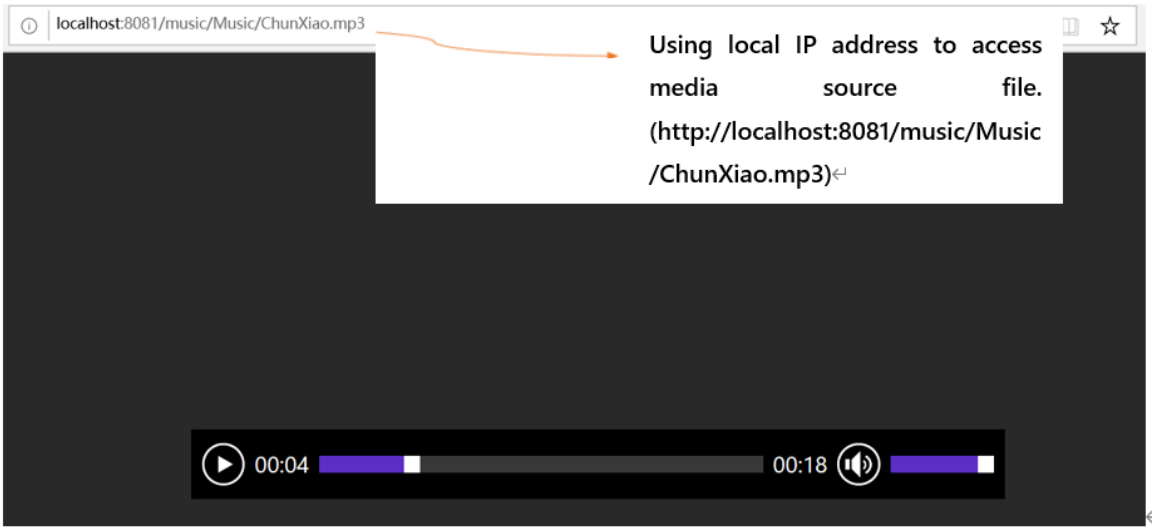


Figure 3.1: Accessing MP3 files through the local website

- **NATAPP**

  A third-party service named NATAPP was used when setting the data source for the Media player (Yue and Ping, 2017). Based on ngrok's reverse proxy software, it establishes a secure channel between the public network and the locally running Web server. In other words, it provides the mapping of the local network to the external network (Figure 3.2, 3.3).

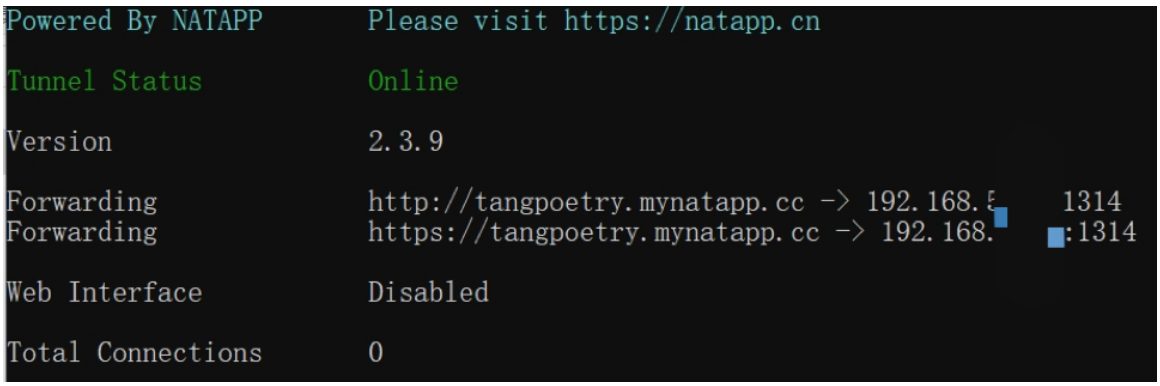


Figure 3.2: Mapping the local network to the external network

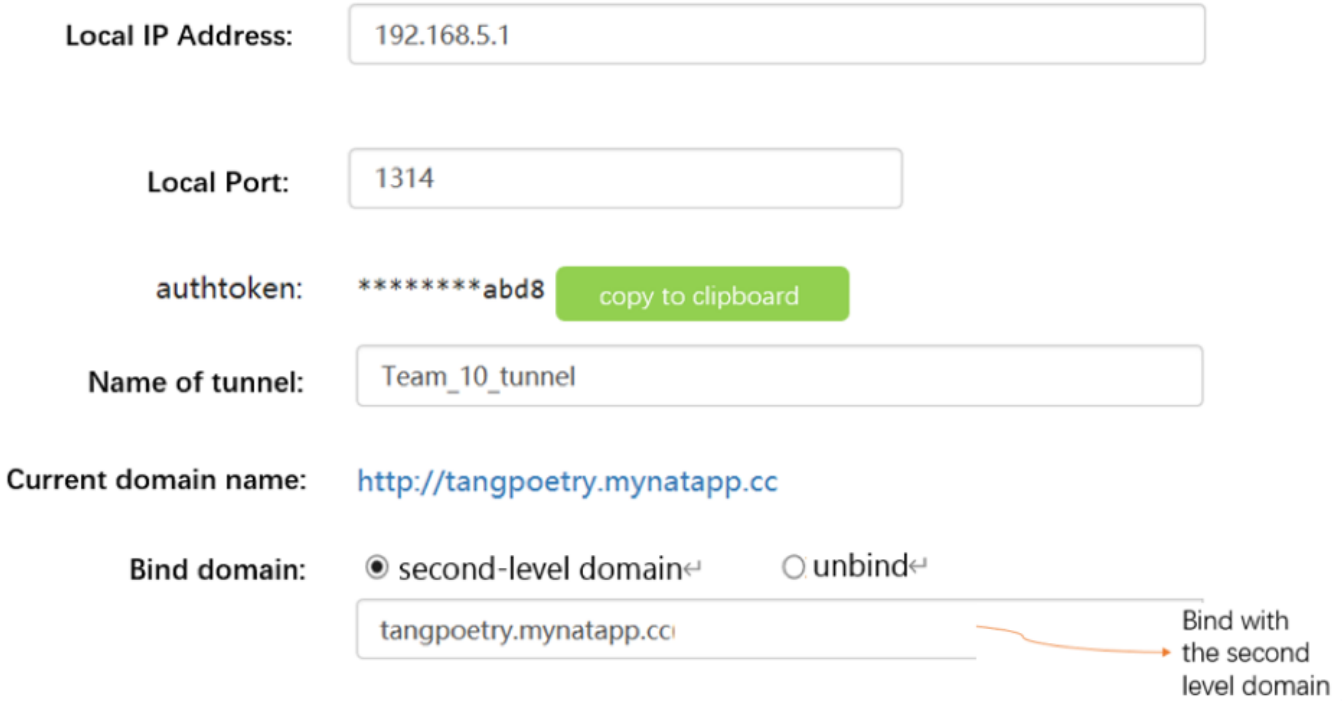


Figure 3.3: Setting second-level domain

After the mapping being established, the external network could access this MP3 file through the second-level domain that is set manually. The URL for the external network to access the file is "https://tangpoetry.mynatapp.cc/music/Sol.mp3" (Figure 3.4).

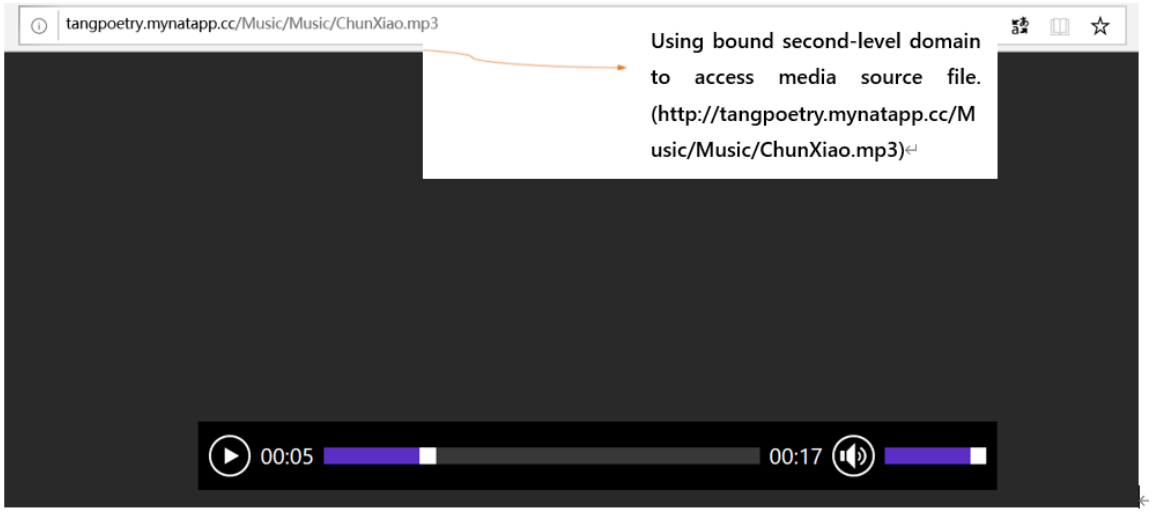


Figure 3.4: URL for external access

- **MySQL**

  In this project, a user system to be designed to realize the personalized needs of users. Therefore, a suitable database for storing user data (such as username, password, email, user image picture, et al.) is required. At present, there are many databases on the market including MySql, Oracle, Sql Server, SQLite and PostgreSQL, but eventually, MySQL was selected. The reasons are as follows:

  1) MySQL database has the characteristics of small in size, fast, low in the total cost of ownership, and providing open-source code (Luan et al., 2010). Generally, MySQL is chosen as the website database for the development of small and medium-sized websites and applications.

  2) Because of its excellent performance in the community version, it can form a good development environment with PHP and Apache server and there is many popular software in this case (Welling and Thomson, 2003). An integrated software called XAMPP, which contains the database visualization service of phpMyAdmin as well as the Apache server, is used for better management of database (Yue and Ping, 2017). It can bring us great convenience that we can use our own computer as a server.

  3) Some basic usages of MySQL were covered in the previous course in year 2, so it greatly reduces the time spent on learning.

- **Volley**

  A network request framework is a necessary tool in this project. It is used for connection and interaction between database and application through network. According to the design requirements of the software, the information interaction of the application will be small and frequency, because most of the interaction occurs when the application is uploading or downloading user's comments on poetry and the storage of the user's personal information. In this situation, compared to other common network frameworks such as OkHttp and Retrofit, Volley is selected as the network request framework for its great performance dealing with highly concurrent network requests. Additionally, Volley has a network request cancellation mechanism, making it easier to manage the request queue (Agarwal et al., 2010).

  There are also some other reasons for choosing Volley. First, when coding with Volley, the management of thread can be out of consideration because Volley has provided a multithreading mechanism when multiple requests need to be handled at the same time. This feature will greatly simplify the codes and make them more readable (Shulin and Jieping, 2014). Second, in the Android Developer official documentation, there is a very detailed guide for using Volley, which will make our learning more efficient and enable us to use the tool correctly.

Chapter 4

Requirement Specification

The reason for having a requirement specification part is that the software is designed for users. Before developing the software, the main functions need to be clarified. For example, in this project, one of the things that heard most from interviewee is to try to be fun. In this part, the requirement specification is split into two parts: the functional requirements and non-functional requirements. A Functional requirement is that developers must implement software features that enable users to complete their tasks, while a non-functional requirement is that quality standards to meet product business requirements.

4.1 Functional Requirements

1. The software system should allow users to register or login.
2. The software system should provide users with a guide on how to use the software system.
3. The software system should provide users with a list of all the Tang poems.
4. The software system should allow users to browse/search for Tang poems by themes' names or authors, et al.
5. For every poem selected by a user, the software system should allow the user to view the translation of it.
6. For every poem selected by a user, the software system should provide information (Pinyin/Romaji) of each character in the poem.
7. For every poem selected by a user, the software system should provide an audio of it which allows the user to listen to the pronunciation of every character.
8. For every poem selected by a user, the software system should provide the user with a video of it which allows the user to see the whole picture of the poem.
9. The software system should provide exercises or games for users to practice what they have learnt.
10. The software system should allow users to keep a record of their favorite poems.
11. The software system should be able to recommend poems to users based on their tastes or progress.
12. The software system should allow users to download poems and learn them.
13. The software system should allow the user to choose how to sort the list of poems, by poem name or author name.
14. The software system should allow the user to choose the type of poems.

15. The software system should allow the user to post own comments in the comments section.

16. The software system should allow the user to choose own font size and background.

17. For every poem selected by a user, the software system should provide the author‘s brief introduction and the background of poem.

18. The software system should allow the user to choose whether turn on eye protection mode.

4.2 Non-Functional Requirements

1. All users' information and study records should be stored in a separated database on the cloud, which should be encrypted to ensure information security and privacy.

2. The response time of the software system should be no more than 6 seconds.

3. The software system should be able to be adapted to different screen sizes and be presented with complete interfaces and functions.

4. The software system should support at least 100 users at the same time.

5. The software system should be available on the Android platform.

Chapter 5

Design of the system and user interface

This part states the design of the software from perspectives of system and interface. The first part describes the basic functions of the application and system structure with UML (Unified Modeling Language) diagrams (use case diagram and class diagram) (Dobing and Parsons, 2008). The second part explains how the system is designed from the aspects of function and data. In the aspect of interface design, the article elaborates on the aspects of design language, color, and animation.

5.1 UML Diagram

5.1.1 Use Case Diagram

In Figure 5.1 (use case diagram), A user can register and log in. The login function depends on the register function. Next, the home page, game page, and personal center page can be viewed by the user. All three use cases depend on the login use case.

The use case "view home page" includes two functions: view daily poetry and view poems. The difficulty can be chosen by the user and then the user can do searching and viewing. The searching poems function is inherited by three methods: searching by letter, searching by name and searching by categories. Besides, the use case "view poems" includes several functions: view original text, translation, video, detailed information, listen to audio, put it in favorite and move it from favorite.

Besides, the use case "view game page" includes three games: Apple Tree, Tang Dynasty Story and Exercise. After finishing the games, the user can check the mark of the game.

Finally, the use case "view personal center" includes the use cases of setting, change the head portrait, view download poems and view favorite poems. Besides, the use case "setting" including several different functions.

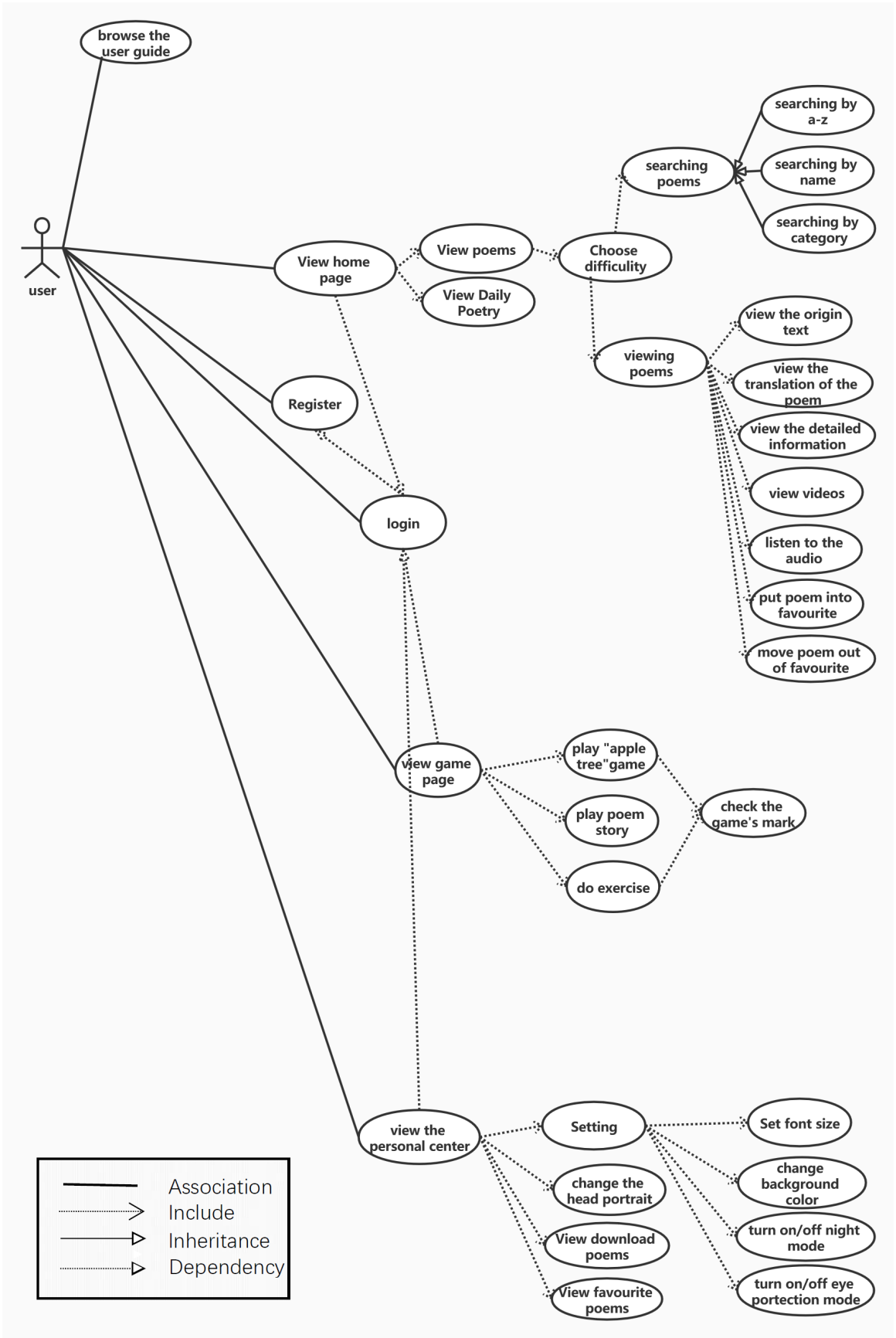


Figure 5.1: Use Case Diagram

5.1.2 Class Diagram

Figure 5.2 is the class diagram of the application, in this diagram, all the class of interfaces inherit the abstract class **BaseActivity**. The classes can be divided into four parts, the first part is the personal center, including **SettingPage**, **ChangeFont**, **BackgroundSetting**. the next part is the poem part, including **PoemPage**, **PoemsPage**, **ChooseDifficulty**, **CommentPage**, **DailyPoetryPage** and **FavouritePage**. The third part is the game part, including **AppleTree**, **GameStoryPage**, and **ExercisePage**. Register/login is the last part, which includes **LoginInterface** and **RegisterInterface**. Besides, some classes depend on the tool classes in other packages.

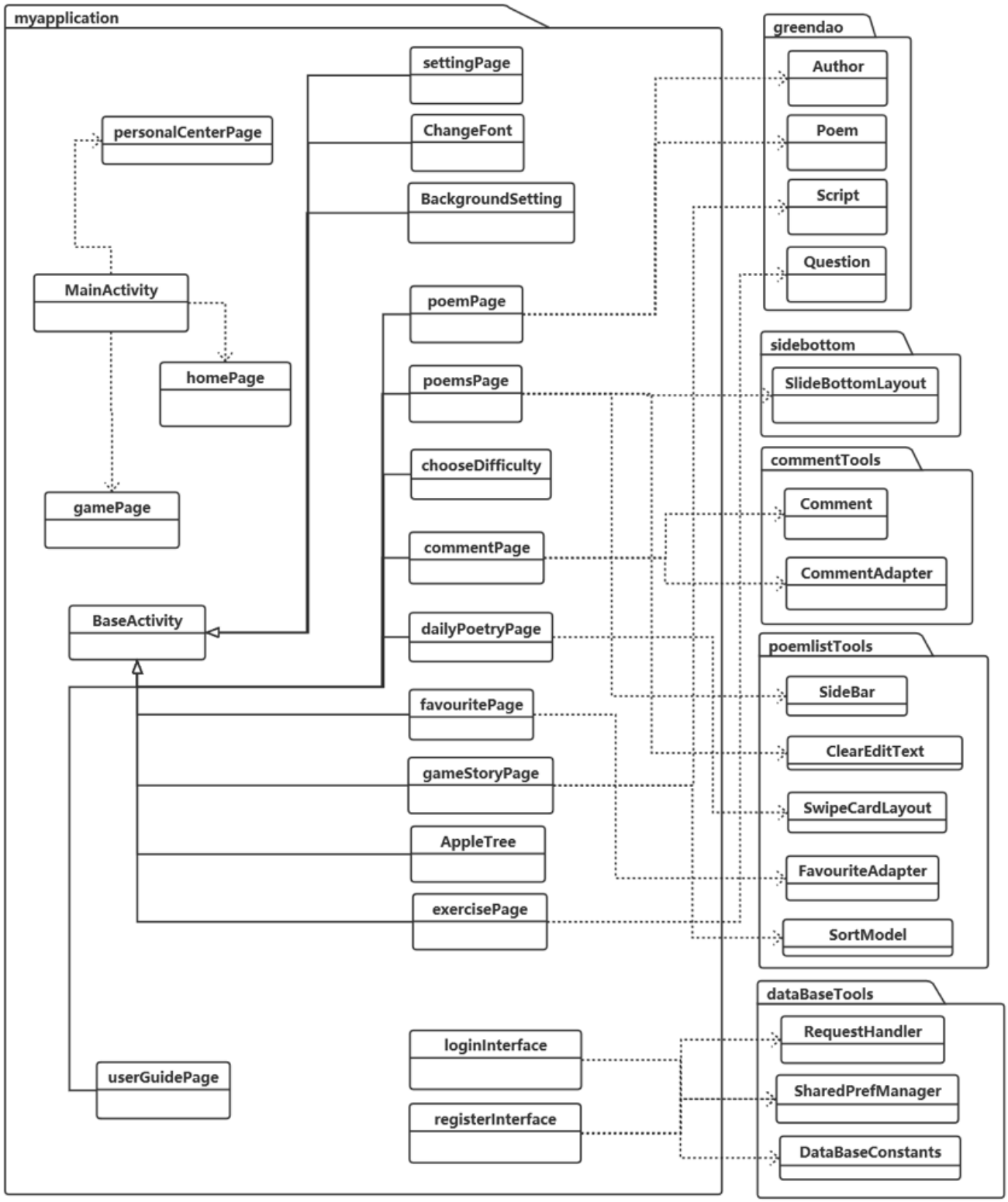


Figure 5.2: Class Diagram

5.2 System Architecture

5.2.1 High-Level Overview

To accomplish all functional and non-functional requirements, the system must be designed meticulously. Several changes may be made since the development used the Agile methodology (Maruping et al., 2009) to search for improvement.

The Tang poetry application mainly includes two main parts – the components of application and data. The two parts are separated by the relation with the user. The application component mainly meets the requirements of the user, while the data component is the connection between the backend and front end. Therefore, for the application component, it will be executed on the user's machines, as it is the user interfaces. As for the data component, it includes servers and databases, which run on either local or cloud-hosted machines.

The data component is mostly based on the server and database. The user could access the application with the purpose of either to learn tang poetry or to do the personalization. The application then retrieves the data from the server and database, which eventually will cause a high load rate. In order to meet the high load, the data component then needs to be designed with and extra mind into it. The system component can be shown in a way of a diagram, shown in the Figure 5.3 below.

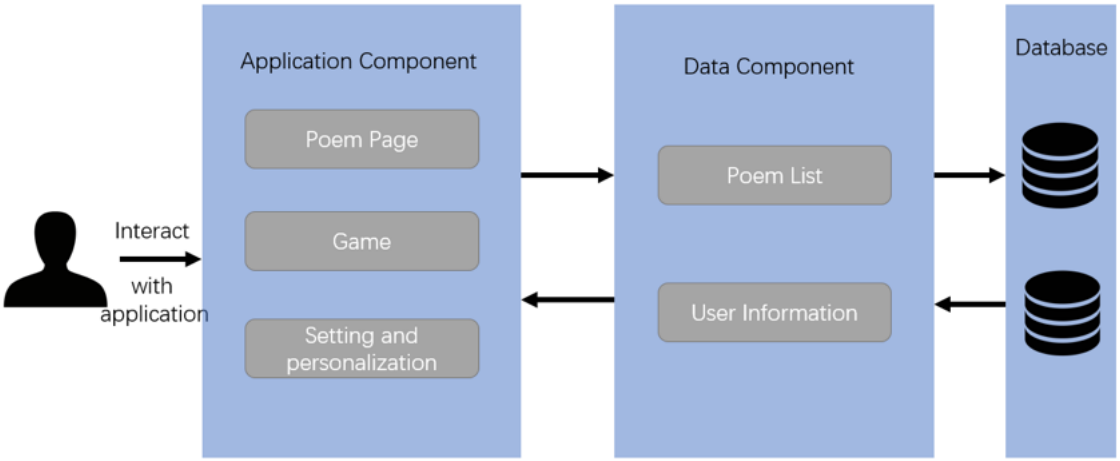


Figure 5.3: System Component

5.2.2 The design of the application component

Learning Tang poetry functions

The functions of learning Tang poetry will be realized from the sections of Tang poetry and Game, which enables users to learn poetries from pronunciation, translation, and emotion expressed by poets.

- **Poem page**
  - **Difficulty selection.** The system should allow users to choose the difficulty level of the poems. When the user selects the mode, Tang poems with the corresponding difficulty will be displayed.
  - **Search function.** One of the requirements is users can view the list of poems sorted according to the names of Tang poems or poets based on their preferences. Additionally, the system should allow users to view the poetry list according to the poetry type. Moreover, the system should provide the search function for users to discover the specific poem and offer two different methods to accomplish the search function. The first one is the user can find the poem or author corresponding to the first letter through the A-Z list on the right. The second one is the user can enter a string of any length in the input box at the top, then click on the search icon, the poem list will return all the lyrics which have this keyword and label the keyword red.
  - **Additional function.** The system should allow the user to translate the poem to English while the user clicks on the TRANSLATE button. Additionally, the system enables the user to watch the video by clicking on the VIDEO button. The system should also allow the user to scroll up the bottom menu to learn about the background of the poem and the author.

- **Comment.** Comment function is another part of the learning function, which provides users with space to exchange ideas with each other. Users can write down their understanding of poetry, exclamation about the life of the poet, questions about poetry or the reply to others' questions. This function can trigger discussion and cooperative learning through this platform, so as to improve the efficiency of learning.

- **Game part**

  The reason for designing the Game part is that the game part brings users disparate experience from rigid Tang poetry learning through graphics and animation (Gozcu and Caganaga, 2016). There are three distinct types of games, which allow users to learn Tang poetry in an interesting method of interacting with the game interface. The content of the study involves the poem, the emotion expressed by the poet, and the history of the Tang Dynasty.

  - **Apple Tree.** The Apple Tree is a gap-filling game. The game is designed to require users to drag the apple with a word to complete the given poem. Once the user answers correctly, the apple on the tree will grow up by animation. The questions of the game are famous poems that are worth learning. The way of dragging apples makes users deepen the impression of poems and better remember them.

  - **Tang Dynasty Story.** Tang Dynasty Story is a drama game. Users can decide the plot direction by selecting the options based on the story background and plots. This game aims to let users feel the Tang Dynasty culture better and understand the mood of poets when they write poetry by story plot description. In this game, the users learn not only from the Tang poetry itself but also from the perspectives of poets and Tang history.

- **Exercise.** The exercise part is different from the previous two games. This part seems to be a serious test, which examines the effect of learning Tang poetry by multiple-choice questions. Exercise questions mainly test whether the user holds the basic information of poetry and understand the emotion the poet expressed.

Basic software functions

This part aims to improve the user experience that every user can have a customizable, one-of-a-kind application, providing users with the opportunity to customize their own their unique programs with customized avatars and personal design.

- **Register and login**

  Consider the requirement of personal customization, an account should be provided to the user to store the personal setting information. Thus, the register and login functionality has been concerned. For the user, the register function can help save the user's behavior track of using the application. Besides, it synchronizes account data on multiple devices and platforms, which improves the effectiveness of this application (Reilly and Zieme, 2011).

- **Setting**

  - **Personalized operation.** This includes the avatar setting and background color setting. These two functions are the implementations of personalized customization. The avatar can be set using the local picture on the mobile phone, and it will be stored in the database for the online interactive function. And there are four background colors for users to choose and use on a global theme. All these operations help improve the user experience (Reilly and Zieme, 2011).

– **User-friendly operation.** This part includes Night mode and Eye protection mode. It is important to consider the user's humanized usage experience. For this application, most of the user's actions are viewing, which relates to the eye using. good design takes care, planning, thought. It takes conscious attention to the needs of the user (Norman, 2013). Therefore, protecting the user's eye while reading will significantly improve the user's reading experience and is good for health. The Night mode reduces the irritation of the phone's light to the eyes in low-light environments. In addition, the Eye protection mode adjusts the screen color temperature to reduce the blue light emitted by the screen, thus decreases irritation to the eyes and makes it less possible to visual fatigue.

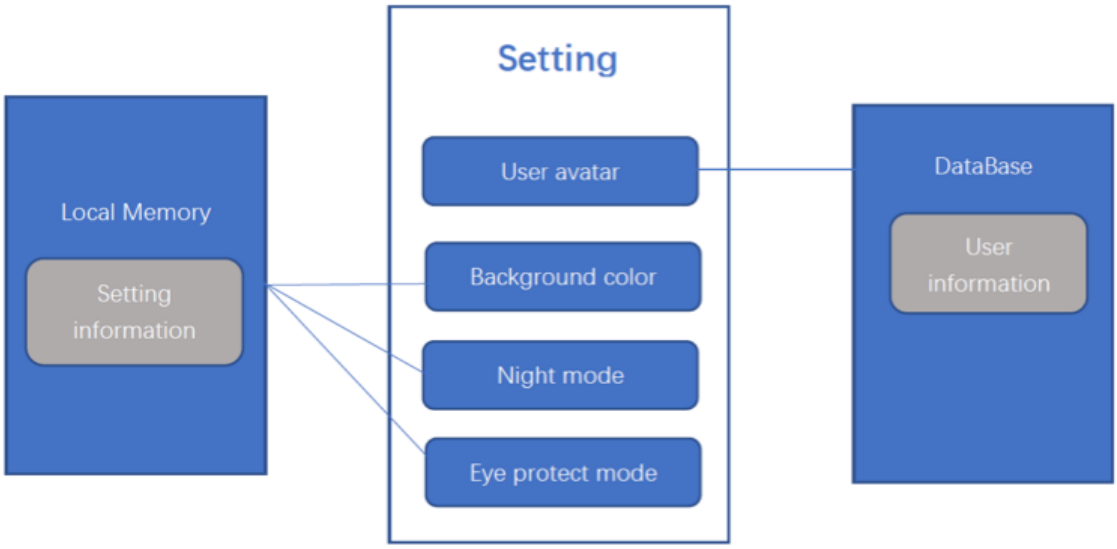


Figure 5.4: Setting Functions

5.2.3 The design of data storage and acquisition

How to interact with data is a problem that must be considered carefully in this project. In this application, data interaction will be applied in many situations, such as reading poetry content, obtaining user login information, obtaining comment content, et al. In this part, two interaction modes, local data interaction and remote data interaction will be introduced.

Local data interaction

Local data interaction is to save the data to be acquired in the form of a file in the application and at the same time obtain it through a certain mechanism. There are two ways to get local data: one is to store it in the local database and get it when necessary, and the other is to read it directly. These two mechanisms will be used in different scenarios.

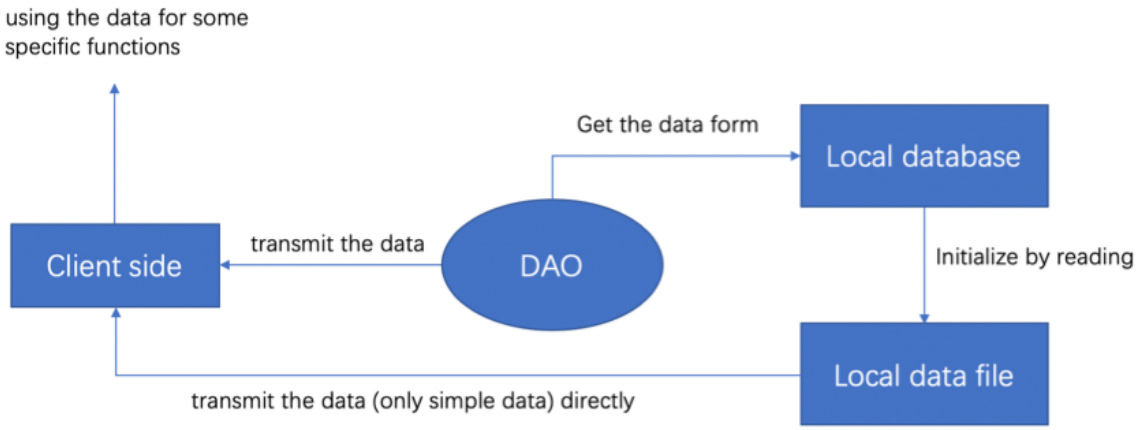


Figure 5.5: Mechanism of storing local data

When the acquired data is large and the structure is complex but requires a fast speed of the process (such as poetry data reading), the local data will be read through the local database. At the beginning of the application, an instance of DAO (data access object) will be created, and the required data will be created on the stack in a proper data structure. In this case, lightweight databases should be chosen to reduce run-time memory usage. At the same time, considering the stability of the program, this method can ensure that the program will not crash directly due to the sudden damage of the file

during the operation of the application.

However, when there is little data to be obtained or the data structure is very simple, directly accessing a local file to obtain data will improve efficiency. At the same time, in order to ensure that the local file is not intruded or maliciously damaged, the path of the file will be hidden.

Remote data interaction

Remote data interaction is to store the data to be acquired in the server and obtain it through the network. It is applied to a large number of data storage, such as all the comment contents or users' private information. The main way to connect the application to the server is to transmit the request through the network connection framework on the client-side and receive the result returned on sever-side. Especially, if the operation is related to the database on the server, the scripting language must be used. Considering the efficiency of the application, database operations should be completed in the script language of the server, and the design of data operation should be avoided on the client-side.

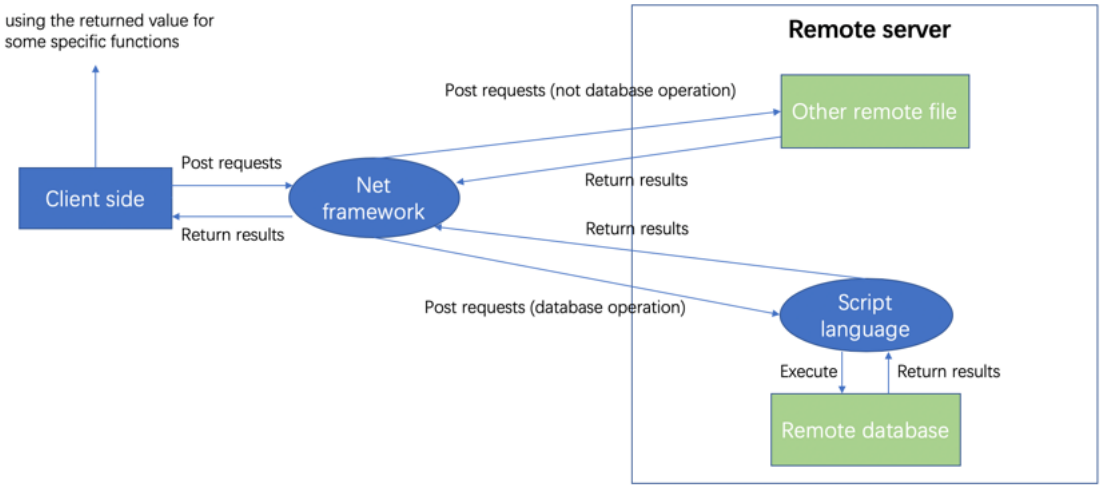


Figure 5.6: Interaction of Remote Data

Additionally, some data security issues need to be considered, such as how to store the user's password safely. Therefore, some basic encryption measures need to be implemented, which can be realized by using the MD5 (message-digest algorithm, a widely used cryptographic hash function) (Deepakumara et al., 2001) on the server-side through a scripting language.

5.3 Interface Design

There are three main aspects of the interface design: designing while previewing, building a feeling of Chinese traditional style and making the interface more user-friendly and interesting.

5.3.1 Design language and tool

Android studio has provided an advanced layout editor (Figure 5.7) that allows the programmer to drag widgets and preview them in real-time after editing the XML (eXtensible Markup Language)(Schwartz and Buttigieg, 2014). When the developer is designing the interfaces, layout editor helps make the design work more efficient.

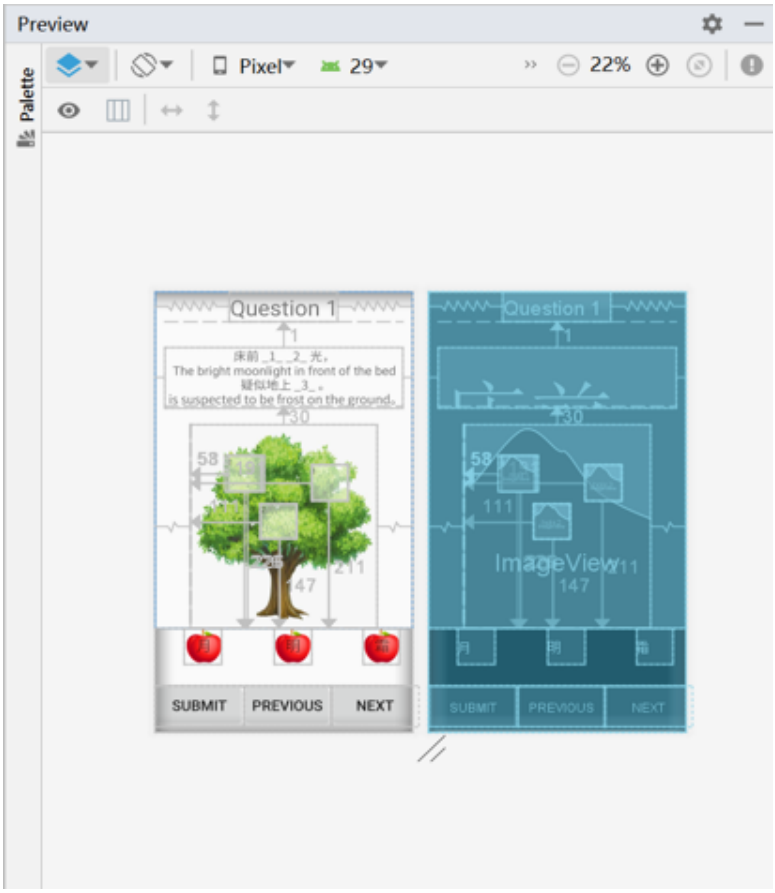


Figure 5.7: A Preview Window provided by Layout Editor

5.3.2 Theme color selection

The color chart is shown below (Figure 5.8) is chosen as the theme color of the Tang poetry learning software. In order to create interfaces of Chinese traditional style, on the whole, it has brown and black colors, together with dark green (Zuixiong, 2002). The brown color is associated with yellow old books and creates a sense of antiquity. For the black color, it might remind the users of Chinese ink painting, which is the treasure of ancient Chinese culture and they were imbued with times features. As for the dark green, it promotes the variety to the theme color and makes it less monotonous.

Figure 5.8: Color chart selected

5.3.3 Animation

The animations are used in the Game part. Since the aim of designing this software is to help users learn and realize the essence of Tang poetry, enhance learning interest by animation would be a good idea to achieve this goal rather than just displaying knowledge to the users. In the Apple Tree game, if the user answers the question correctly, an animation of apples growing from the tree will be shown to indicate that the user has done an excellent job. This is also the idea of instant rewards in the game design field to make users get more interested in the game.

5.3.4 Card

Instead of clicking buttons to shift from one poetry to another in a traditional way, a card interface is adapted in the design of the poetry reading part. From the user's point of view, shifting between poetries by simply sliding the screen with fingers would be more comfortable and interesting. This design conforms to Fitts' law (MacKenzie, 1992). In other words, target area for users' operation is large and easy to implement, which improves user's operation efficiency.

Chapter 6

Implementation

This section focuses on how the important functions of the software are implemented by code. The following statements will discuss code hierarchy and code implementation (Keuler et al., 2012). The code hierarchy part introduces the structure of the application from the code aspect, and the code implementation part will present the realization of main functionalities of the User system, Bottom Navigation bar, Tang poetry list, Game and Personal center.

6.1 Overview of code hierarchy

Explicit architecture engagement purposes enabled the organization in planning and tracking architecture as an activity within their software engineering. This system structure can be easily shown as a diagram, seen in the Figure 6.1 below.

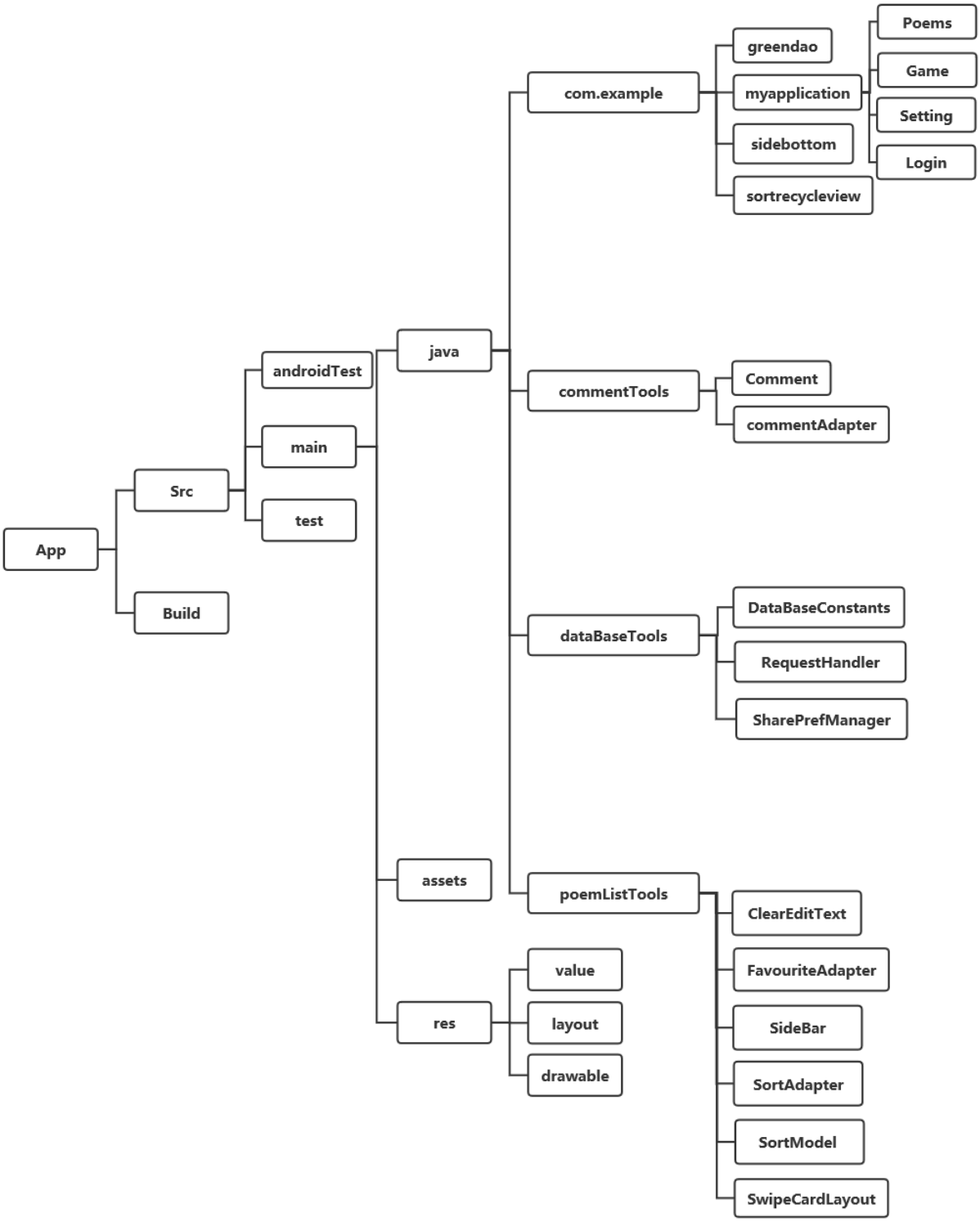


Figure 6.1: Hierarchy of the application

As the figure shows, the app package is composed of two parts, the build folder stores the files to build and the Src folder stores all the resource files. Test files in androidTest and test folder. In addition, all the resource files are in the main folder where code files are stored in the java folder, image resources of the interface are stored in the res folder and native resource files are stored in the assets folder. In the java folder, the code file is divided into four sections which are three tools and the executing body of a program. The body contains the GreenDao (a database), two view components and the main application.

The application consists of three main components – the resource, activities and controlling tools. The resource of application are stored in the assets packages and res packages, the former stores the font files and the poem's txt files while the latter stores the layout, Image resource files and values. Next, the activities can be divided into four integral parts: Poems, Game, Setting, and Login, corresponding to the main functions related to the program respectively. Finally, the controlling tool part includes comment tools, database tools, and poem list tools.

The application using GreenDao, volley, and butterknife as external dependencies:

- **GreenDao**

  GreenDao is a lightweight and fast open-source solution that maps objects to SQLite databases, it is appropriate to use it for lightweight data usage for this application (Bernard Che, 2019).

- **Volley**

  Volley is an HTTP library that makes networking for Android applications easier and, most importantly, faster for highly concurrent network requests. The network requests the cancel mechanism (Agarwal et al., 2010).

- **Butterknife**

  This frame provides powerful View binding and Click event handling to simplify code and improve development efficiency.

6.2 Code Implementation

6.2.1 User system

The user interaction system of this application is implemented by MySQL, PHP, and Volley network framework. In this part, how they are used to achieve the user system's functionalities of login, registration and uploading user's profile will be explained.

- **Using MySQL to store data**

  MySQL is mainly used to manage the data stored in the database by executing some SQL statements (Welling and Thomson, 2003). In this project, two separate databases were set for different purposes. The databases named **tang_application** are designed for storing users' personal information and comments are used for storing the contents commented by users in every poem.

  In **tang_application**, a table called **users_login** is designed, which contains five items: **id**, **username**, **password**, **email**, and **picture**. The **id**, the primary key of the table, is set to be **AUTO_INCREMENT**, which means **id** will be automatically initialized with an increased number every time a new user is signed up so that every user can get a unique id. **Username**, **password**, and **email** are the sections for storing the information registered in the database by users, and users can use their username and password to login and enter the application. The **picture** is used to store the absolute path of the user's profile picture in the server. When the application needs to obtain the users' profile picture, this path is required to initiate a request.

  For the **comments** database, a series of tables were designed because the comments for each poem corresponds to an independent table. The name of the table is **poem_x**, where **'x'** refers to the index of the poem. Each table contains eight items: **comment_id**, **content**, **poem_id**, **username**, **parent_id**, **date**, **like_number**,

picture.

The **comment_id** is the primary key of the table. **Content**, **username**, **date**, **like_number**, and **picture** are the sections for storing the information showing in one comment while **parent_id** refers to the **poem_id** that the comment reply to.

- **Using Volley to initialize requests and receive results on the client-side**

  Volley works whenever clients need to make requests to the database. Volley provides a class called **StringRequest**, which is designed to send the user's input data to the specified PHP file in the form of an object of **Map** and receive a **JSONObject** as the return value (Agarwal et al., 2010).

```
StringRequest stringRequest = new StringRequest(Request.Method.POST,
    DataBaseConstants.URL_LOGIN, new Response.Listener<String>() {
     @Override
     public void onResponse(String response) {
         try {
             JSONObject obj = new JSONObject(response);
             //If the key "error" of obj is false, the user
             //login successfully
             if(!obj.getBoolean("error")){
                 //Things to do if user logins successfully
             }else {
                 //Things to do if user fails to login
             }
         } catch (JSONException e){
             e.printStackTrace();
         }
     }
}, new Response.ErrorListener() {
     @Override
     public void onErrorResponse(VolleyError error) {
```

```
20          //Things to do because of the failure of connecting server
21      }
22  }){
23      @Override
24      protected Map<String, String> getParams() throws AuthFailureError
        {
25          //Get the response information and return it
26          Map<String, String> param = new HashMap<>();
27          param.put("username", username);
28          param.put("password", password);
29          return param;
30      }
31  };
```

Code 1: The usage of Volley

This framework will be applied every time a request for network data interaction with the database needs to be initialized, such as user registration, sending comments, user login, et al.

- **Using PHP as the script language executed on the server-side**

  When the user sends the request, the request will be received by specified PHP files on the server-side and the database operation will be realized by operating the file (Welling and Thomson, 2003). There is a series of functional PHP files as well as header PHP files in the server, which respectively implement the specific requests initiated by the user system in the client-side and provide database parameters for the database connection. All functional PHP files need to use the functions included in the header PHP files. Take user login as an example:

```
public function userLogin($username, $pass){
    $this->con = $this->db->connect();
    $password = md5($pass);
    $stmt = $this->con->prepare("SELECT id FROM user_login WHERE
    username = ? AND password = ?");
    $stmt->bind_param("ss", $username, $password);
    $stmt->execute();
    $stmt->store_result();
    return $stmt->num_rows() > 0;
}
```

Code 2: The implementation of user login

In the **DB_operation.php**, there is a function **userLogin()** declared, and this function is called in **userLogin.php**.

```
$db = new DB_operation();
if($db->userLogin($_POST['username'], $_POST['password'])){
    $user = $db->getUserByUsername($_POST['username']);
    $response['error'] = false;
    $response['id'] = $user['id'];
    $response['email'] = $user['email'];
    $response['username'] = $user['username'];
    $response['picture'] = $user['picture'];
}else{
    $response['error'] = true;
    $response['message'] = "Invalid username or password";
}
```

Code 3: userLogin() in DB_operation.php

In this way, PHP files can receive the data sent by the client and operate the database, as the bridge between the client and the server.

6.2.2 Bottom Navigation Bar

The application was separated into three sections: Home, Game and Personal center. The bottom navigation bar was firstly realized by applying three buttons at the bottom. There are two drawbacks to using buttons. Firstly, by using the button module, a new activity will be created each time the button is clicked, a mass of memory is used wastefully. Meanwhile, the speed of the system response is slowed down. Secondly, the button is not artistic to a certain degree. Currently, the majority of the applications are utilizing the **Fragment** in order to acquire the bottom navigation bar function, the team also implanted the **Fragment** class into the project.

To realize this function, these three sections are switched into the sub-class of the **Fragment** class. The next step is to create a **Fragment manager** and realize the fragment showing function.

```
1  private void showFragment(Fragment fragment){
2          if(currentFragment != fragment){
3              FragmentTransaction transaction = fManager.beginTransaction();
4              transaction.hide(currentFragment);
5              currentFragment = fragment;
6              if(!fragment.isAdded()) {
7                  transaction.add(R.id.ly_content, fragment).show(fragment).
       commit();
8              }else{
9                  transaction.show(fragment);
10                 transaction.commit();
11             }
12         }
13     }
```

Code 4: The implementation of showFragment()

Then, a switcher is needed. Here the **MainActivity** class serves as the switcher. The main function is implemented by clicking on the icon, the specific content is shown on the **Framelayout** by using **switch-statement**.

```
<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_above="@id/div_tab_bar"
    android:id="@+id/ly_content">
</FrameLayout>
```

Code 5: The container of the fragment

```
View.OnClickListener l = new View.OnClickListener(){
    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.txt_home:
                setSelected();
                txt_home.setSelected(true);
                showFragment(hp);
                break;
            case R.id.txt_game:
                setSelected();
                txt_game.setSelected(true);
                showFragment(gp);
                break;
            case R.id.txt_me:
                setSelected();
                txt_me.setSelected(true);
                showFragment(pcp);
                break;
        }
```

```
21          }
22      };
```

Code 6: Showing FrameLayout using switch-statement

6.2.3 Poem List

Model. As with any object-oriented programming language, objects are one of the fundamental building blocks behind any application. Therefore, to clarify what data the application is processing, a separate package (greendao) containing all the models is created. All the models are plain old Java objects with custom annotations from GreenDao that define the relationships between them and provide guidance for converting to relational database models.

In the Poem List part, two objects were created using this model. One is called **Author** and the other is called **Poem**. As the name shows, the class **Author** is used to store the author‘s information, the class **Poem** is used to store the poem‘s information.

```
1  @Entity
2  public class Author {
3      @Property(nameInDb = "authorNameEnglish")
4      private String authorNameEnglish;
5
6      @Property(nameInDb = "authorBriefIntroduction")
7      private String authorBriefIntroduction;
8
9      //getters, setters and constructors omit
10     //...
11 }
```

Code 7: Model Class **Author**

In Code 8, **@Entity** means that when the program is running, the class annotated by

@Entity is found automatically, generates the classes needed for GreenDao. **@Property** means the system will create a column named **nameInDb** (i.e. **authorNameEnglish** et al.). In **Author** Class, it is used to store author's name in English and the author's brief introduction.

```
@Entity
public class Poem {
    @Id
    private Long id;
    @Property(nameInDb = "difficulty")
    private String difficulty;
    @Property(nameInDb = "poemName")
    private String poemName;
    @Property(nameInDb = "poemNameEnglish")
    private String webLink;
    @Property(nameInDb = "webLink")
    private String poemNameEnglish;
    @Property(nameInDb = "authorName")
    private String authorName;
    @Property(nameInDb = "authorNameEnglish")
    private String authorNameEnglish;
    @Property(nameInDb = "kindOfPoem")
    private String kindOfPoem;
    @Property(nameInDb = "chineseVersion")
    private String chineseVersion;
    @Property(nameInDb = "EnglishVersion")
    private String EnglishVersion;
    @Property(nameInDb = "poemBackground")
    private String poemBackground;
    private String poemNameHtml;
    private String authorNameHtml;
    private String audioLink;

```

```
29      //getters, setters and constructors omit
30      //...
31  }
```

Code 8: Model Class **Poem**

The **Poem** class is used to store the following list of things:

- **difficulty:** The difficulty of this poem is based on personal experience.
- **difficulty:** The Chinese version of the poem's name.
- **poemNameEnglish:** The English version of the poem's name.
- **weblink:** The link of the video to this poem.
- **authorName:** The Chinese version of the author's name.
- **authorNameEnglish:** The English version of the author's name.
- **kindOfPoem:** The classification of poetry types, in this software, there are five kinds of poems: 1 for Frontier poems; 2 for farewell poems; 3 for lyrics; 4 for poetries chanting things; 0 for other poems.
- **chineseVersion:** The Chinese version of the whole poem.
- **EnglishVersion:** The English version of the whole poem.
- **poemBackground:** The background of this poem.
- **poemNameHtml and authorNameHtml:** These two things are used after the search. If there is the same keyword as the user input, then turn the string into HTML and display the string in HTML.

Function. After the login interface is initialized, the data of poems and authors will be imported into the database by **poemDao** and **authorDao**. There are two lists are defined, one called **Poems**, which is used for displaying what it contains in the **Recyclerview**, while another called **Poems_middle**, which is used for storing the middle

value of changing **orderWay** (order by poem name or author name), **kind** and **difficulty**.

Contacts a-z alphabetical sort using RecyclerView implementation. When the user selects a letter, the screen will show the selected letter at the center of the interface by **Sidebar** class. Code 9 shows how to draw an A-Z letter list. First, getting the height of the space that the interface remains. Then, calculating the height of each letter. Finally, the **painter** draws the letter on **canvas**.

```
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    int height = getHeight();
    int width = getWidth();
    //Get the height of each letter
    int singleHeight = height / b.length;

    for (int i = 0; i < b.length; i++) {
        paint.setColor(Color.rgb(33, 65, 98));
        paint.setTypeface(Typeface.DEFAULT_BOLD);
        paint.setAntiAlias(true);
        paint.setTextSize(30);
        if (i == choose) {
            paint.setColor(Color.parseColor("#3399ff"));
            paint.setFakeBoldText(true);
        }

        float xPos = width / 2 - paint.measureText(b[i]) / 2;
        float yPos = singleHeight * i + singleHeight;
        canvas.drawText(b[i], xPos, yPos, paint);
        paint.reset();
    }
}
```

Code 9: The implementation of **onDraw()**

Then, when importing the poems into the list, the system could get the starting position of each letter. When the user clicks on one letter, the position of this letter will be gotten by the function **getPositionForSection()**. If the returned position is within a reasonable range, the list of poems will present poems beginning with this letter.

```
public int getPositionForSection(int section) {
    for (int i = 0; i < getItemCount(); i++) {
        String sortStr;
        if(orderWay == 1){
            sortStr= Poems.get(i).getPoemNameEnglish();
        }else{
            sortStr= Poems.get(i).getAuthorNameEnglish();
        }
        char firstChar = sortStr.toUpperCase().charAt(0);
        if (firstChar == section) {
            return i;
        }
    }
    return -1;
}
```

Code 10: The implementation of **getPositionForSection()**

Filter searches. If the user enters a keyword in the search field, the string will be transferred to the function **afterTextChanged()**. Then, the function **filterData()** will be called to filter all poems containing the keyword and mark all keywords contained in the poem name or author name.

Code 11 shows that how do these poems return to the original state when the string in the search bar is empty. The **PoemNameHtml** and **AuthorNameHtml** of all the poems will reset to null, and all marks will disappear.

```
mClearEditText.addTextChangedListener(new TextWatcher() {
    @Override
    public void onTextChanged(CharSequence s, int start, int before, int
    count) {}

    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int
     after) {}

    @Override
    public void afterTextChanged(final Editable s) {
                    if(TextUtils.isEmpty(s.toString())) {
            for (Poem poem : Poems_middle) {
                if (poem.getAuthorNameHtml() != null || poem.
    getPoemNameHtml() != null) {
                    poem.setAuthorNameHtml(null);
                    poem.setPoemNameHtml(null);
                }
            }
        }
        mAdapter.updateList(Poems_middle);
        search.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                filterData(s.toString());
            }
        });
    }
});
```

Code 11: The implementation of **addTextChangedListener()**

Code 12 shows the search algorithm. It will check whether the author's name and poem name contain the string which is input by the user. If the author's name and poem name contain the string, it will change the string format to the HTML format. Otherwise, doing

nothing.

```
for(Poem poem:filter){
    String poemName = poem.getPoemNameEnglish();
    String authorName = poem.getAuthorNameEnglish();

    if(authorName.toLowerCase().contains(filterStr.toLowerCase()) ||
    poemName.toLowerCase().contains(filterStr.toLowerCase())){
        int length = filterStr.length();
        int index1 = poemName.toLowerCase().indexOf(filterStr.toLowerCase()
    );
        int index2 = authorName.toLowerCase().indexOf(filterStr.toLowerCase
    ());
        if(index1 != -1){
            poem.setPoemNameHtml(poemName.substring(0,index1) + "<font
    color='#f08519'><b>" + poemName.substring(index1,index1+length) + "</b
    ></font>" + poemName.substring(index1+length));
        }
        if(index2 != -1){
            poem.setAuthorNameHtml(authorName.substring(0,index2) + "<font
    color='#f08519'><b>" + authorName.substring(index2,index2+length) + "</b
    ></font>" + authorName.substring(index2+length));
        }
        filterDateList.add(poem);
    }
}
```

Code 12: The implementation of search algorithm

Media player. The media player function is used to play web audio(Code 13). The source of audio is stored on a local website using XAMPP (Apache+MySQL+PHP+Perl) and then given a second-level domain with the help of natapp so that extranet could also access this audio. The data source is surrounded by a try-catch block to ensure the source is valid. The function adapts asynchronous prepare method to implement playing while

buffering. Once the user clicked the "PLAY" button, a toast will be shown indicating that the audio is loading. After the audio source being prepared, it would start to play. Besides, if the audio is stopped, then the media player will be destroyed and release the resource that it occupies.

```
mediaPlayer = new MediaPlayer();
try {
    mediaPlayer.setDataSource(audio_pp);
} catch (IOException e) {
    e.printStackTrace();
}
mediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
mediaPlayer.prepareAsync();
mediaPlayer.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
    @Override
    public void onPrepared(MediaPlayer mp) {
        mediaPlayer.start();
    }
});
if (mediaPlayer != null && mediaPlayer.isPlaying()) {
    mediaPlayer.stop();
    mediaPlayer.release();
    mediaPlayer = null;
}
```

Code 13: The implementation of media player

6.2.4 Game – Apple Tree

As for the specific construction of this game, it can be divided into three parts: a welcome page, questions page, and a congratulation page. All the three parts use a xml file for a layout and a java file to add functions for widgets.

Long click event and drag event. For the question part, there are nine different questions with a similar structure. These apples with Chinese characters are set with an **OnLongClickListener** (Code 14) which enables the user to do the operation that after clicking the apple for about one second, a shadow of this apple will be created and dragged following the user's instruction.

```
1  private void dragSetting(View v){
2      ClipData.Item item = new ClipData.Item(v.getTag().toString());
3      ClipData dragData = new ClipData(v.getTag().toString(),new String[]{
       ClipDescription.MIMETYPE_TEXT_PLAIN},item);
4
5      // Instantiates the drag shadow builder.
6      View.DragShadowBuilder myShadow = new View.DragShadowBuilder(v);
7      v.startDrag(
8          dragData,    // the data to be dragged
9          myShadow,    // the drag shadow builder
10         null,        // no need to use local data
11         0            // flags (not currently used, set to 0)
12     );
13 }
```

Code 14: The implementation of **OnLongClickListener()**

For the three holes (translucent squares) on the apple tree, each of them has three layers with the same size: an **ImageView**, a **FrameLayout** and a **TextView**. The **FrameLayout** will implement an object of **OnDragListener**.

In the **OnDragListener** (Code 15), a switch-case structure is used and for each necessary case, **v.invalidate()** is called to force a redraw, in this way the shadow of the apple will always exist during the whole drag process. When the shadow is dropped in a hole, firstly the id of this hole needs to be checked. Then the text contained in the shadow will be extracted and filled to the corresponding place of the verses.

```
case DragEvent.ACTION_DROP:
    // Gets the item containing the dragged data
    ClipData.Item item = event.getClipData().getItemAt(0);
    if(v == hole1){
        s1 = item.getText().toString();
        setQuestionSelect(numOfQues);
        System.out.println("s1 is "+s1);
    } else if (v == hole2){
        s2 = item.getText().toString();
        System.out.println("s2 is "+s2);
        setQuestionSelect(numOfQues);
    } else if (v == hole3){
        s3 = item.getText().toString();
        System.out.println("s3 is "+s3);
        setQuestionSelect(numOfQues);
    }
    return true;
```

Code 15: The implementation of **OnDragListener()**

Judge the answer. "SUBMIT" button will enable them to check the answer by setting a **mySubmitListener** (Code 16). This function will check if all the choices mathches the right answers. An animation of apples growing on the tree will be shown if the answers are correct.

```
if(s1.equals(r1)&&s2.equals(r2)&&s3.equals(r3)){
```

```
    Toast.makeText(AppleTreeQuestion.this, "Correct!", Toast.LENGTH_LONG).
    show();

    tv1.setBackgroundColor(Color.parseColor("#00FFFFFF"));
    tv2.setBackgroundColor(Color.parseColor("#00FFFFFF"));
    tv3.setBackgroundColor(Color.parseColor("#00FFFFFF"));

    Animation scaleAnimation = new ScaleAnimation(0,1,0,1,Animation.
    RELATIVE_TO_SELF,0.5f,Animation.RELATIVE_TO_SELF,0.5f);
    scaleAnimation.setDuration(2000);

    //rest of code...
}
```

Code 16: The implementation of **mySubmitListener()**

6.2.5 Game – Tang Dynasty Story

Model of Script. The plots of each scene are stored in the GreenDao database, and the script mode is saved in greendao package. The entity of plot (**Script**) defines five chunks of substances. They are the main characters, the content of the plot, the selectable operations, the next plot that will be executed, and the use of animation. The system will present the scenarios in sequence. The **Script** model can be seen below.

```
@Entity
public class Script {
    @Property(nameInDb = "ID")
    private int ID;
    @Property(nameInDb = "Role")
    private String role;

    private String content;
    private String action1;
```

```
10      private String result1;
11      private String action2;
12      private String result2;
13      private String light;
14      //getters, setters and constructors omit
15      //...
16  }
```

Code 17: Model Class **Script**

Transform in the plot. When the system needs to display the next scenario on the screen, the **QueryBuilder** class will be used to query and filter out the target plot based on the ID of the next script. Using **QueryBuilder** class not only enables programmers to stop building custom queries with SQL but also help programmers find errors at compile time.

```
1  private Script getNewScript(int ID){
2      QueryBuilder qb = scriptDao.queryBuilder();
3      qb.where(ScriptDao.Properties.ID.eq(ID));
4      List<Script> scr = qb.list();
5      if(qb.list().size() > 0){
6          return scr.get(0);
7      }
8      return null;
9  }
```

Code 18: The implementation of getting a new script

Animation effects. On account of the requirement of the storyline, the game has added animation effects. The animation presents the form of a flash of white light that whitens the interface and then recovers two seconds later. This causes the effect of dizziness. The method of this animation is to place a white image (**light**) of interface size on the layout file. At the beginning of the interface loading, setting the image to be invisible. With the help of **AlphaAnimation** class, the image gradually appears and dribble away.

Due to the duration of the animation is two seconds, the software will execute the following code before the animation is finished. This will result in the next interface to emerge during the animation, and it creates animation meaningless. In order to solve this problem, the **postDelayed()** function in the **Handler** class is used to delay the main thread for two seconds. In this way, the unity of animation and interface can be maintained. The code of animation can be seen below.

```
1  private void lightEffect(){
2      light.setVisibility(View.VISIBLE);
3      AlphaAnimation alphaAnimation1 = new AlphaAnimation(0.1f, 1.0f);
4      alphaAnimation1.setDuration(2000);
5      light.setAnimation(alphaAnimation1);
6      alphaAnimation1.start();
7
8      //wait 2 second and show next page
9      Handler handler = new Handler();
10     handler.postDelayed(
11         new Runnable() {
12             @Override
13             public void run() {
14                 setContent();
15             }
16         }, 2000);   //after 2 seconds to run run()
17 }
```

Code 19: The implementation of animation effect

6.2.6 Game – Exercise

Model of Question. All exercise questions are stored in the database using GreenDao. Therefore, a separate package (greendao) containing all the models is created. The entity

of exercise question (**Question**) defines the information of questions in the table. The system will randomly gather out ten questions to consist of a question list and present the questions with corresponding three options of each question in sequence. The **Question** model can be seen below.

```
@Entity
public class Question {
    //@Id
    private int ID;
    @Property(nameInDb = "Question")
    private String question;
    @Property(nameInDb = "AnswerA")
    private String answerA;
    @Property(nameInDb = "AnswerB")
    private String answerB;
    @Property(nameInDb = "AnswerC")
    private String answerC;
    @Property(nameInDb = "Answer")
    private int answer;
    @Property(nameInDb = "Explanation")
    private String explanation;
    @Property(nameInDb = "Hint")
    private String hint;
    //user select answer
    private int selectedAnswer;

    //getters, setters and constructors omit
    //...
}
```

Code 20: Model Class **Question**

Hint. Every exercise question has a corresponding hint to help users answer the question. When users press the HINT button, the system will obtain hint information from

the **Question** table, and the hint with the translation of poem in question will be shown in the form of an alert dialog. For some questions, which aim at judging the emotions of the poet and the substances to be described according to the poem, the system will appear relevant figures to make the question easier. This operator will be realized by **showPicture()** function.

```
1  //get hint information of current question
2  btn_hint.setOnClickListener(new View.OnClickListener(){
3      @Override
4      public void onClick(View view){
5          String[] strArr = q.getHint().split("\\*");
6          new AlertDialog.Builder(exercisePage.this)
7              .setTitle("Hint")
8              .setMessage(strArr[0]+"\n"+strArr[1]+"\n"+strArr[2])
9              .setPositiveButton("Confirm", new DialogInterface.
       OnClickListener() {
10                 @Override
11                 public void onClick(DialogInterface dialogInterface, int i)
       {
12                     //do nothing
13                 }
14             }).show();
15         showPicture();
16     }
17 });
```

Code 21: The implementation of hint function

Review question pattern. There are two modes of reviewing pattern – review the wrong question and review all the questions. Just like the mode names, the system will emerge the incorrect questions on the screen in the mode of reviewing the wrong questions, and the other will show all the ten questions. Both modes will record the selected option for each question. Different from the interface of answering the question, these two

modes both present the problem explained in the form of the scroll view. In any mode interface, modes can be switched to each another through the **SetOnClickListener()** function (Code 22). The system uses **Intent** to assist in reloading the mode interface. For example, if the user decides to change the mode, the interface will change from one mode to the other mode. When the correct rate of the user is 100%, the review wrong question mode will be superseded by an alert dialog to tell the user there is no wrong question.

```
btn_changeModel.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(wrongList.size()==0){
            new AlertDialog.Builder(exerciseWrongPage.this)
                .setTitle("Reminder")
                .setMessage("There is no wrong question.")
                .setPositiveButton("Confirm", new DialogInterface.
    OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialogInterface,
    int i) {
                        //do nothing
                    }
                }).show();
        }else {
            if (exerciseResultPage.reviewMode == 0) {
                exerciseResultPage.reviewMode = 1;
                btn_changeModel.setText("REVIEW ALL QUESTIONS");
            }else {
                exerciseResultPage.reviewMode = 0;
                btn_changeModel.setText("REVIEW WRONG QUESTIONS");
            }
            Intent intent = new Intent(exerciseWrongPage.this,
    exerciseWrongPage.class);
            startActivity(intent);
```

```
24            }
25        }
26    });
```

Code 22: The implementation of changing review mode

6.2.7 Personal center

The key is to create the function modules users will use and combine them with the main function to implement global use. The functions are "changing font-size", "night mode", "setting background" and "eye protects mode". The activity has been created called **BaseActivity** to realize the global change.

- **Setting functions**

  Changing the font size. To implement the global change, in the **BaseActivity** class, the static float **fontsize** represents the value of the global font size. The font size can be got and used for every class inherits the **BaseActivity** class to set the new font size.

```
1  @Override
2  public Resources getResources() {
3      Resources res =super.getResources();
4      if (Build.VERSION.SDK_INT <= Build.VERSION_CODES.N) {
5          Configuration config = res.getConfiguration();
6          SharedPreferences sharedPreferences = getSharedPreferences("
       font_size", Context.MODE_PRIVATE);
7          SharedPreferences.Editor editor = sharedPreferences.edit();
8          config.fontScale = getFontSize();
9          res.updateConfiguration(config,res.getDisplayMetrics());
10     }
11     return res;
12 }
```

```
13
14 @Override
15 protected void attachBaseContext(Context newBase) {
16     if(Build.VERSION.SDK_INT>Build.VERSION_CODES.N){
17         final Resources res = newBase.getResources();
18         final Configuration config = res.getConfiguration();
19         config.fontScale = getFontSize();
20         final Context newContext = newBase.createConfigurationContext(
    config);
21         super.attachBaseContext(newContext);
22     }else{
23         super.attachBaseContext(newBase);
24     }
25 }
```

Code 23: The implementation of getting font size

Night mode. For this function, the night theme is added to implement the night effect. In **BaseActivity**, the on/off state of the mode can be get and changed using **isEnableNightMode()** and **changeNightMode()**.

```
1 public abstract class BaseActivity extends AppCompatActivity {
2     private static boolean enableNightMode = false;
3     private static boolean enableColorMode = false;
4     private static int numOfColor = 0;
5     private static boolean enableEyeProtectMode = false;
6     private static Context mContext;
7     private static float fontsize = 1;
8
9     @Override
10    protected void onCreate(@Nullable Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        if(!enableNightMode) {
```

```
13              AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.
    MODE_NIGHT_NO);
14          } else {
15              AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.
    MODE_NIGHT_YES);
16          }
17
18      }
19      /**
20       *If enabled night mode
21       * @return  true or false
22       */
23      public boolean isEnableNightMode() {
24          return enableNightMode;
25      }
26      public boolean isEnableColorMode() {
27          return enableColorMode;
28      }
29      public boolean isEnableEyeMode() {
30          return enableEyeProtectMode;
31      }
32      public int checkThemeColor() {
33          return numOfColor;
34      }
35
36      //rest of the code
37      /...
38  }
```

Code 24: The implementation of night mode

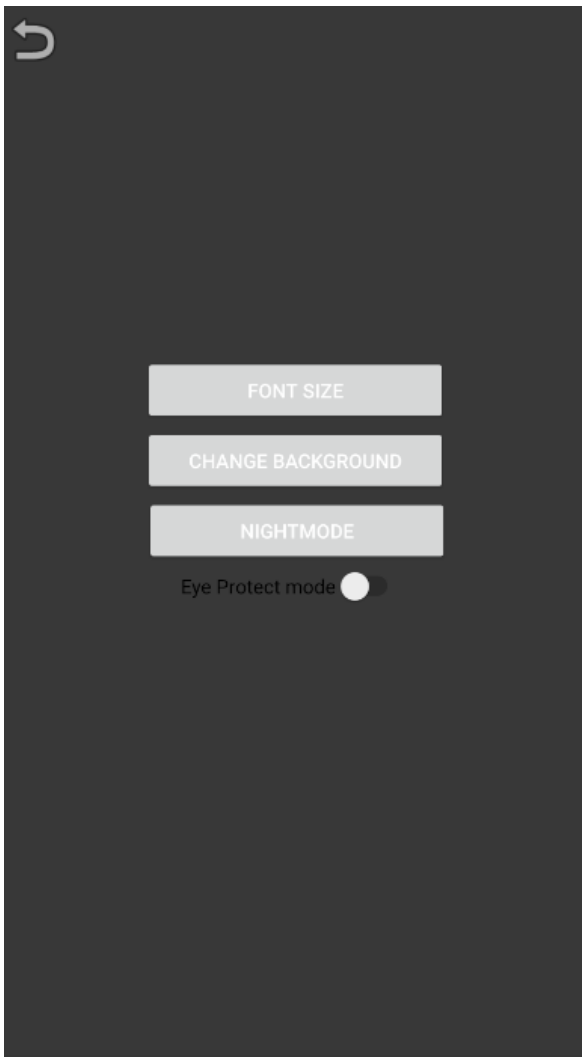


Figure 6.2: The Effect Picture of Night Mode

Background setting. In **BaseActivity**, the property **enableColorMode** represents the on/off state of the mode. The **numofColor** represents the type of background color, 1 represents Brown... and 4 represents Cyan. The function **isEnableColorState()** and **checkThemeColor()** can get the state and the color of the mode.

```
1    ...
2    private static boolean enableNightMode = false;
3    private static boolean enableColorMode = false;
4    private static int numOfColor = 0;
```

```
5     private static boolean enableEyeProtectMode = false;
6     ...
```

Code 25: The properties in **BaseActivity**

In the setting activity, the corresponding number of colors are set in the **BaseActivity** class. The next steps are recreating the activity, judging the state and getting the color number in **BaseActivity** and resetting the theme of the activity.

```
1 Button btn_theme = (Button) findViewById(R.id.night);
2 btn_theme.setOnClickListener(new View.OnClickListener() {
3     @Override public void onClick(View v) {
4         theme = (theme == R.style.AppTheme) ? R.style.NightAppTheme: R
    .style.AppTheme;
5         changeNightMode();
6         settingPage.this.recreate();
7     }
8 });
```

Code 26: The implementation of changing color mode

Eye protect mode. The method is to reduce the blue light of the screen. The key implementation is to realize an overlay window that can change the amount of blue light of the display layer. In this way, a class following the singleton pattern is created to generating the dialog which covers the whole screen and has an RGB model with less blue.

```
1 public class EyeDialog extends Dialog {
2     private static EyeDialog msEyeDialog;
3
4     protected EyeDialog(@NonNull Context context, boolean cancelable,
    @Nullable OnCancelListener cancelListener) {
```

```
5           super(context, cancelable, cancelListener);
6       }
7
8       public EyeDialog(@NonNull Context context, int themeResId) {
9           super(context, themeResId);
10      }
11
12      public static EyeDialog getInstance(Context context, int
    themeResId){
13          if (msEyeDialog == null){
14                  msEyeDialog = new EyeDialog(context,themeResId);
15          }
16          return  msEyeDialog;
17      }
18
19      public static EyeDialog getInstance2(){
20          return  msEyeDialog;
21      }
22
23  }
```

Code 27: The implementation of **EyeDialog**

In the setting page, the user can start the Mode using **EyeProtectMode()** in **BaseActivity**. Then it will create a dialog using **openAlertWindow()**, setting the color of the dialog to the color with less blue. A **WindowManger** tool is used to realize the effect of suspending on all the activities and manage the state of this window. When the mode is off, an instance of the dialog will be shown. The dialog will be removed using the **dismiss()** function.

```
1   mSwitch.setOnCheckedChangeListener(new CompoundButton.
        OnCheckedChangeListener() {
2       @Override
```

```
3      public void onCheckedChanged(CompoundButton buttonView, boolean
    isChecked) {
4          if (isChecked){
5              SharedPreferences preferences = getSharedPreferences("user
    ", Context.MODE_PRIVATE);
6              SharedPreferences.Editor editor = preferences.edit();
7              editor.putBoolean("flag", true);
8              editor.commit();
9              changeEyeProtectMode();
10             openAleterWindow();
11         }else {
12             SharedPreferences myPreference=getSharedPreferences("
    myPreference", Context.MODE_PRIVATE);
13             SharedPreferences.Editor editor = myPreference.edit();
14             editor.putBoolean("flag", false);
15             editor.commit();
16             changeEyeProtectMode();
17             if(ld>=0&&ld<=100&&red>=0&&red<=100&&blue>=0&blue<=100&&
    green>=0&&green<=100) {
18                 editor.putInt("ld", ld);
19                 editor.putInt("red", red);
20                 editor.putInt("blue", blue);
21                 editor.putInt("green", green);
22                 editor.commit();
23             }
24             Log.e("this","data"+ld+red+blue+green);
25             Dialog d =  EyeDialog.getInstance2();
26             d.dismiss();
27         }
28     }
29 });
```

Code 28: The implementation of opening the dialog

```
1 public void getData(){
2     SharedPreferences preferences=getSharedPreferences("myPreference",
```

```
        Context.MODE_PRIVATE);
3       iv_main.setBackgroundColor(Color.argb(alapha,200,200,150));
4       changeAppBrightness(ld);
5       Log.e("this","getR"+red);
6   }
```

Code 29: The implementation of setting the color and the lightness

Chapter 7

Evaluation

As the tasks are divided into several parts, each part has members in charge. During the development, numerous test cases are created. The performance then depends on the provided test cases. The following content is the evaluation of the performance on four perspectives: game, poem, database, and interface.

7.1 Database

In this project, two databases, GreenDao and MySQL, are used for different purposes. This part will focus on their performance and give an evaluation of them.

- **GreenDao**

  GreenDao is chosen as the database for storing and accessing local data. In general, GreenDao handles the local data with good efficiency. Because the data comes from the local files, the reading speed of GreenDao becomes a great advantage. At the same time, GreenDao provides the developer with good data management. When using GreenDao, a developer only needs to define the data model, for example, the Poem class in this project. The GreenDao framework will create data objects (entities) and Dao (data access objects), making the codes simpler and more readable. Besides, the small dependency library of it is also an advantage. The critical dependency library size of GreenDao is no more than 100kb.

However, from the perspective of building an application mechanism, GreenDao still has some potential disadvantages as a local database. First, if the poetry text file becomes too large to be accepted with the supplement further, the defects brought by GreenDao are irreversible, even though the GreenDao can store poetry text in a relatively light form locally. At the same time, the storage of local files means that if the poetry text library needs to be frequently updated, the client must also be forced to update frequently, even if other parts of the application do not need to be updated, and this may bring users a terrible experience if they are frequently informed that the application needs to be updated.

- **MySQL with Apache**

  MySQL is chosen as the remote database with the use of Apache service. On the whole, MySQL achieves the remote storage of data at good processing speed and provide efficient API for Java. With the help of PhpMyAdmin, the management of MySQL is convenient.

  On the other hand, for the evaluation for the usage of MySQL, the biggest defect comes from the fact that members have no access to a real formal server but choose to use our own computer to store the database using Apache instead. This leads to the result that currently, this application cannot afford too many registrations from users because of the limited computer memory space. In addition, using limited space database leads to the decision of giving up some functionalities, including users' personal collection and preference oriented recommendation, because it is not realistic to allocate enough storage space for each user in the database to collect and store data of each user's preference for poetry (this partly also stems from the failure to find appropriate and efficient algorithms and data structures to collect and analyze user data).

7.2 Interface

- **Perspective of application**

  In this project, as mentioned in the chapter of design, the design idea of the interface is to build a feeling of Chinese traditional style and make the interface more user-friendly and interesting. A color chart is chosen as the main theme of this software, and these colors do represent traditional Chinese culture to some extends. Besides, some custom widgets with ink painting styles are used in the interface design to enhance this effect. If more time was spent on the interface design, adding sufficient custom widgets may make it even better. Moreover, some animations and a card-like interface are used to improve the user experience. These designs are often used in existing software on the market, which demonstrates the users' acceptance of these modes of operation. Therefore, it could be considered as reasonable to implement these designs in this project.

- **Perspective of coding**

  The layout editor is used to realizing designing while previewing, which helps the design work more efficiently. Team members can have a more intuitive view of the design process and adjust the design in time.

7.3 Poem

The poem list part has realized the essential functions to complete the study effect, but there are still many deficiencies in the implementation.

- **Poems Page**

  The number of poems is limited. That is to say, the users can only learn some of the Tang poems from this software.

  The size of all controls in the interface is fixed. It cannot be changed according

to the size of the phone's screen, which means that some of the users who have a smaller screen than expected will not be allowed to view the inner poem page, because they cannot touch the button which is used for going into the internal poem page.

The history search function is not implemented, meaning that users cannot view what they have searched. This may lead to the result that users need to spend more time finding the same poem.

- **Poem Page**

  The number of links, authors‘ introductions, the background of poems and pictures is limited. That is to say, the users can only watch the video, learn the background of poems of a limited number of poems from this software.

  There is no corresponding pinyin on each character, which means that the users need to know Chinese characters before using the software. This will undoubtedly reduce the interest of many people in learning.

7.4 Game

This part enables users to learn the relevant knowledge of Tang poetry in the process of playing games. There are three different styles of games in the application. The game part has realized the basic function to complete the study effect, but there are still many deficiencies in the implementation.

- **Apple Tree**

  The drag apple operation of the game increases the interaction between the game and the user. This game is easy to understand and takes less time to operate. Users are able to use their spare time to play this game. However, the game has restrictions on users. The setting of the game requires that users are able to recognize

Chinese characters. Therefore, the game is only suitable for users with the basic knowledge of Chinese characters.

- **Tang Dynasty Story**

  The game enables users to remember the historical stories of the Tang Dynasty according to the aside of story and character dialogues. The plot of the story is adapted by the team members based on the history of the Tang dynasty. However, because of the uniqueness of the plots, it is difficult to find a background picture that is very consistent with the story. In addition, the interactive options of the game are less. Most of the plots are monologues, which may reduce the interest of the game. Therefore, more options should be added to make the game more amusing.

- **Exercise**

  Exercise enables users to examine the learning effect formally. The disadvantage of exercise is that in Hint function, the information presented may not be relevant to some types of questions. The prompting information should prompt the most effective information according to different question types. And the function of customizing exercise questions that meet the current learning progress of the user is not realized.

7.5 Personal Center

The personal center part consists of two main parts: personalized customization of the theme and the miscellaneous function.

- **Personalized customization**

  In the personal center part, the goal of personalized customization has been achieved. The head portrait can be set by the user freely, which also makes a contributes to user interaction in the comment page. The user can achieve the background and

font size setting as well. Whereas there are still some deficiencies. The background setting doesn't support using local images, which reduces the freedom of the personalized setting.

- **Miscellaneous function**

  The two new Miscellaneous functions take the user's humanized requirement into account. The night mode and eye protection mode make the user more comfortable with the eyes when using the application in the case of different light conditions. However, the night mode button is better to be implemented using a switch button to make the state of the switch more intuitive.

Chapter 8

Summary of what was achieved

In this chapter, first, the initial time plan will be displayed. Then, the process breakdown will be introduced, followed by a checklist of what has been implemented according to the requirement specification. Finally, a specific analysis of the implementations will be presented.

8.1 Initial time plan

ID	Task Name	Duration	Start	Finish	dependance
1	Project Analysing	6 days	Tue 10/8/19	Tue 10/15/19	
2	Market Research	13 days	Tue 10/15/19	Thu 10/31/19	
3	Preparing the interview	7 days	Fri 11/1/19	Sun 11/10/19	
4	Interview	1 day	Mon 11/11/19	Mon 11/11/19	Preparing the intervie
5	Requirement Specificatio	15 days	Tue 11/12/19	Sat 11/30/19	
6	Design the interface of the software	17 days	Fri 11/15/19	Sat 12/7/19	
7	Interim report	19 days	Sat 11/16/19	Wed 12/11/19	
8	Building the prototype	21 days	Wed 11/20/19	Wed 12/18/19	
9	Interface implement	12 days	Mon 12/23/19	Tue 1/7/20	prototype
10	Login and Register	6 days	Tue 1/7/20	Tue 1/14/20	
11	Database of the poetry established	5 days	Thu 1/9/20	Wed 1/15/20	
12	Randomly daily poem implementation	6 days	Thu 1/16/20	Thu 1/23/20	
13	Searching system implementation	11 days	Thu 1/16/20	Thu 1/30/20	
14	Game implementation	27 days	Mon 1/20/20	Tue 2/25/20	
15	Favourite implementatio	4 days	Sat 2/22/20	Wed 2/26/20	
16	Combinating all parts of software	9 days	Wed 2/26/20	Sat 3/7/20	
17	Testing	24 days	Sun 3/8/20	Wed 4/8/20	
18	Developing the software	23 days	Mon 3/9/20	Wed 4/8/20	testing
19	Final report	15 days	Fri 3/20/20	Thu 4/9/20	
20	Preparing for Open Day	8 days	Fri 4/10/20	Tue 4/21/20	
21	Final presentation	1 day	Wed 4/22/20	Wed 4/22/20	

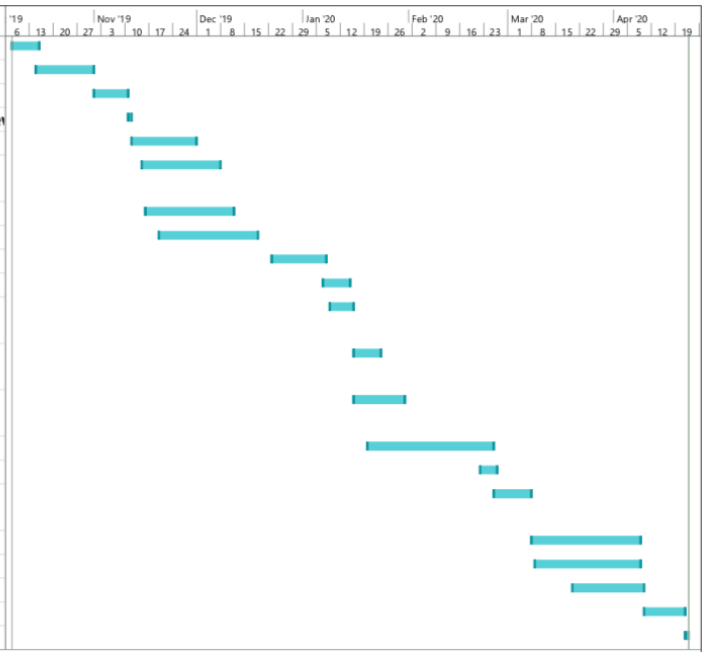


Figure 8.1: Gantt chart for initial plan

8.2 The progress breakdown

The project was divided into chunks of stages. Each stage has its planned tasks including core setup, requirements, design, implementation, and documentation work. Below are the working stages, their goals, and the results.

- **Sprint 1:** 10/8/2019 - 10/15/2019

  Goal: The first sprint will be focused on the analysis of this project. To be more specific, the analysis is about considering what preparation needs to be done before implementing this project.

  Result: A detailed plan for what needs to be done in the following three weeks is put forward according to the project description.

- **Sprint 2:** 10/15/2019 - 10/31/2019

  Goal: The second sprint will be focused on doing marketing research (development platform, interface design, specific content of similar software that related to this project).

  Result: Android is chosen as the development platform for this project, some interface design and specific content which are considered as useful are recorded for reference.

- **Sprint 3:** 11/1/2019 - 11/10/2019

  Goal: The third sprint will be focused on preparing interview questions. Based on the result of market research, some user requirements are put forward and their necessity needs to be checked with target users.

  Result: A question list for the interview is completed and an interview with the target user is arranged.

- **Sprint 4:** 11/11/2019 – 11/11/2019

  Goal: This sprint will be interviewing with the interviewee. The major objective is

to have a better understanding of the preferences of target users. In addition, the user model will be created based on the result of the interview.

Result: The interview went well. The information gathered was sufficient. The preference was collected, and the user model was created.

- **Sprint 5:** 11/12/2019 - 11/30/2019

  Goal: This sprint will be focusing on the preparation of requirement specifications. The requirements will be gathered from the task description, interview results, and analyzed by the team.

  Result: The requirement specification was successfully done, both functional requirements and nonfunctional requirements were created. The sprint went as planned.

- **Sprint 6:** 11/15/2019 - 12/7/2019

  Goal: Focus will be oriented towards software interface designing. The interface will be designed according to the requirement specification.

  Result: The interface design was mostly done, remaining some minor tasks. This led to less working time on prototype development.

- **Sprint 7:** 11/16/2019 - 12/11/2019

  Goal: This sprint will have a focus on writing the interim report according to what we have done in the previous race.

  Result: The interim report was going very well, as expected.

- **Sprint 8:** 11/20/2019 - 12/18/2019

  Goal: This sprint will have a focus on building the prototype by JavaFX while writing the interim report. The prototype should show what functions had designed in the interface.

  Result: The prototype went as planned.

- **Sprint 9:** 12/23/2019 - 1/7/2020

  Goal: This sprint will have a focus on implementing the designed prototype on to the Android Studio.

  Result: This sprint was a bit late than expected because our team spent more time on learning the knowledge about Android development and preparing for the exams.

- **Sprint 10:** 1/8/2020 - 1/23/2020

  Goal: This sprint will have a focus on implementing some main functions of application – login and register, Tang poetry list, and randomly generated daily poem. Before these functions will be realized, two databases for storing user information and Tang poetry should be established.

  Result: The two databases were successfully established, and hundreds of Tang poems were added to the Tang poetry database. The user login and registration function could be used smoothly. The functions of presenting the Tang poetry list and generating daily poems had been implemented.

- **Sprint 11:** 1/16/2020 - 2/22/2020

  Goal: This sprint will have a focus on the implementation of the searching function, the games and the favorite page.

  Result: The searching function has been successfully implemented in the poem part. And three games have also been developed for the user to play, including apple tree, poem story and exercise. There are some difficulties in the implementation of the favorite page and this part wasn't been developed successfully.

- **Sprint 12:** 2/26/2020 - 4/8/2020

  Goal: This sprint will be focused on the integration of all code parts completed by each team member and the program test according the test cases (see in Appendix

A). Furthermore, on the basis of the result of the test, it carries out the repair of programming error, code optimization and some interface improvements.

Result: Team encountered some difficulties in the process of integration. Because some members could not connect git, and the coupling between modules was very high, it took a lot of time to integrate because members had to integrate the code manually. The test part was done with little obstacle, and on the basis of the result of the test, members gave suggestions to each other to optimize the code and the interface. Overall, this sprint is three to four days later than planned due to difficulties in integration.

8.3 Functional requirements check list

Table 8.1: Functional requirements check list

NO.	Name of functional requirement	Having been implemented or not? (Yes/No)
1	The software system should allow users to register or login.	Y
2	The software system should provide users with a guide on how to use the software system.	Y
3	The software system should provide users with a list of all the Tang poems.	Y
4	The software system should allow users to browse/search for Tang poems by themes' names or authors, etc.	Y
5	For every poem selected by a user, the software system should allow the user to view the translation of it.	Y
6	For every poem selected by a user, the software system should provide information (Pinyin/Romaji) of each character in the poem.	N
7	For every poem selected by a user, the software system should provide an audio of it which allows the user to listen to the pronunciation of every character.	N (not for every poem, but the function is implemented)
8	For every poem selected by a user, the software system should provide the user with a video of it which allows the user to see the whole picture of the poem.	N (not for every poem, but the function is implemented)
9	The software system should provide exercises or games for users to practice what they have learnt.	Y
10	The software system should allow users to keep a record of their favorite poems.	N
11	The software system should be able to recommend poems to users based on their tastes or progresses.	N
12	The software system should allow users to download poems and learn them.	N
13	The software system should allow the user to choose how to sort the list of poems, by poem name or author name.	Y
14	The software system should allow the user to choose the type of poems.	Y
15	The software system should allow the user to post own comments in the comments section.	Y
16	The software system should allow the user to choose own font size and background.	Y
17	For every poem selected by a user, the software system should provide the author‘s brief introduction and the background of poem.	Y
18	The software system should allow the user to choose whether turn on eye protection mode.	Y

8.4 Non-functional requirement check list

Table 8.2: Non-functional requirements check list

NO.	Name of non-functional requirement	Having been implemented or not? (Yes/No)
1	All users' information and study records should be stored in a separated database on the cloud, which should be encrypted to ensure information security and privacy.	Y
2	The response time of the software system should be no more than 6 seconds.	Y
3	The software system should be able to be adapted to different screen sizes and be presented with complete interfaces and functions.	N
4	The software system should support at least 100 users at the same time.	Y
5	The software system should be available on the Android platform.	Y

8.5 Analysis

The analysis will be presented from two aspects: the requirements that have been implemented, whereas others are unconsummated.

Specifically, most requirements are implemented according to the requirement specification. For instance, according to the functional requirements, basic functions of poetry reading have been implemented. Uses can also play audios or videos of some specific poems, or view background information of the poem to help them get a deeper understanding. Besides, a searching function and a login system with comment function are also realized. In addition, a font size changing function and an eye protection mode are implemented to provide better user experience. Moreover, a gaming part of three different games is also implemented to make the learning process more interesting.

There are still some functions that are unaccomplished, such as keeping a record of users' favorite poems, recommending poems to users based on their tastes or progresses, providing information (Pinyin/Romaji) of each character in the poem and enabling users to download resources of poems. For the first two requirements mentioned above, they are abandoned mainly because there is not enough variety of poems in database, therefore, the recommendation might not be accurate to meet user's interests. Additionally, for the function of adding Pinyin/Romaji of characters, users might be unfamiliar with them and prefer to use phonetic symbol, which could result in these notes becoming meaningless. As for the downloading function, copyright issues should be considered, thus, this function is also removed in the end.

For the non-functional requirements, a database is set up to store all the user information separately and the whole system responds quickly on the Android platform. What has not been done yet is adapting the system to different screen sizes. This function is related to model adaption issues. For example, different phones have different length-width ratios. Some widgets suitable on a phone whose aspect ratio is 3:2 may get overlapped after the adaption to a phone with aspect ratio of 20:5.9. Therefore, it may cause unpredictable bugs which is not good for the stability of the system.

Chapter 9

Reflective comments on the success of the project

In this part, the reflective comments on the success of the project from a technical and a project management perspective will be given. Besides, the illustration of some unsolved technical issues and the discussion of some work issues occurred during the process of the project will be included.

9.1 The reflective comments on the technical part of the project

9.1.1 Achievement

- **Adaptability to unfamiliar development environment**

  During the project preparation, team members did not reach a consensus on the development platform, because different members have their areas of expertise. In the end, however, the members decided to develop the project on the Android platform that they were completely unfamiliar with. This is a great challenge because members have to search for literature materials and access to websites to obtain knowledge of Android programming development. This not only improves the programming skills of members but also improves the ability to learn quickly in an

unfamiliar development environment.

- **Ability of using open source distributed version control system**

  In the development of this project, GitHub was selected as the repository of the codes and used to perform version control. Team members created branches for task assignment and were required to upload their parts to Git so that everyone could keep track of the progress of the project. Besides, because of the large scale of the project, the conflicts of merging frequently occurred, but members still manually solved these problems properly.

9.1.2 Unsolved technical issues

- **Some unsolved programming problems still exist**

  Although the application of the project can run normally in the end, some small errors or warnings still exist and cannot be solved temporarily:

  - When implementing the Media player function, an error message (1, -2147483648) occurred which could be resulted from several reasons: wrong file path, implementing Mediaplayer.prepare() after implementing Mediaplayer.create() (this could lead to duplicate use of prepare() method), internet permission not being included in the manifest.xml et al. This made it difficult to fix the error. After trying all possible solutions to solve this problem, the error message still existed, and it failed to play the music when running the program on a simulator (Pixel 2 API 29) provided by Android Studio. However, the program worked well on real phones and could play music normally. It can be inferred that this error message might come from a model adaption problem, and more relative knowledge should be gathered to solve it.

  - When using the simulator to debug, the problem of unable to obtain the network permission has always occurred on the official simulator. This failed to debug the connection of the database on the simulator. Although it works nor-

mally on another third-party simulator called NetEase-Mumu and real device, it may mean that the way of getting access in our design is not suitable for every device

- **Interface is not delicate enough**

  In the limited project time, team members have spent most of the time on the functional part, resulting in insufficient design capacity of the interface. Some Tang style elements on widgets were not implemented. Besides, there is always a problem with the resolution mismatch in the application on the device. Members adopt a fixed resolution (1080 * 1920) in the design without realizing the adaptation function of different resolutions.

- **Some members met the Git connection problem**

  During the project, several team members could not successfully link to Git due to the network limitation. While integrating codes, members who could not connect to Git had difficulty accurately locating which files have been changed. In this case, contrasting the previous versions caused much more expenditure on time.

9.2 The reflective comments on the management of the project

9.2.1 Achievement

- **Communication without meeting**

  Since February, due to the epidemic situation, all members of the team could not return to school for project development. This brought great difficulties to the communication among members. However, through weekly frequent online meetings, members could still deal with these issues well by arranging a time to attend meet-

ings and actively expressing their ideas. After each online meeting, one member will take charge of the minutes of the meeting and send them to each member immediately, so that each team member can make their task assignment clear.

- **Reasonable division of tasks**

  Before designing software, members can communicate about their areas of expertise and make reasonable task allocation. This greatly improves the efficiency of programming, which is also the main reason why the project has made rapid progress in function realization.

9.2.2 The discussion of the working issues

- **The issues of time management**

  It is a common problem mentioned by most members in the team. The reasons are listed as follow:

  - The team has never learned how to develop on the platform of Android before. Therefore, members have to learn the development environment in a short period. In this case, when bugs appear, it is difficult to solve the problem expeditiously. This drawback has made the project more time-consuming.

  - In addition to the time to deal with the team project, the coursework and the arrangement of lectures of other subjects make team members feel stressed. Sometimes the pre-set schedule cannot be completed on time, resulting in the delay of some work.

  - On account of the epidemic situation in the second semester, the members cannot have meetings face to face, which decreases the efficiency of communi-

cation. In this case, the team sometimes failed to follow the timetable.

- **The issues of doing the interview**

  Before doing the requirement specification, all the members agreed on organizing an investigation for a better pattern design as well as the function design. The investigation is for the students and teachers who are native English speakers and willing to have experience of Tang poetry and aiming to collect some suggestions and perspectives from them. However, some difficulties were encountered during the investigation:

  - Most foreign students or teachers seem to have no idea about Tang poetry, even be indifferent when the invitations of the interview were sent. This phenomenon led to the result that fewer people were willing to participate in the interview, thus, little information was collected.

  - Although some people were willing to accept the survey, the quality of the survey is not as good as expected. The main reason is that they have no interest in poetry learning, so they only provided some valuable opinions on the interface design and some interactive learning methods. However, they did not provide any suggestions for poetry content learning.

  In addition, because of the delay of returning school due to the coronavirus, the interview that should have been in continuous progress could not be realized. In this case, members cannot update the requirement specifications and have to use the information acquired from the autumn semester to develop the software.

- **The merging issues during the management of modules**

  The merging problems during the management of software come from the fact that

the modules of the software are not separated independently (for example, the same variables are used in different modules), which causes the high coupling of them, resulting in the difficulty of merging different parts of the code. Additionally, during the development of the software, different databases were used (not communicated well). Thus, members suffered from the lack of understanding of the theory of database of each other, which caused a great expense of time on learning.

Bibliography

Sharad Agarwal, John Dunagan, Navendu Jain, Stefan Saroiu, Alec Wolman, and Habinder Bhogan. Volley: Automated data placement for geo-distributed cloud services. 2010.

Longho Bernard Che. Room vs GreenDAO for android: A comparative analysis of performance of room and GreenDAO, 2019.

Sanchit Chadha, Antuan Byalik, Eli Tilevich, and Alla Rozovskaya. Facilitating the development of cross-platform software via automated code synthesis from web-based programming resources. *Computer Languages, Systems & Structures*, 48:3–19, 2017.

Janaka Deepakumara, Howard M. Heys, and Ramachandran Venkatesan. FPGA implementation of MD5 hash algorithm. In *Canadian Conference on Electrical and Computer Engineering 2001. Conference Proceedings (Cat. No. 01TH8555)*, volume 2, pages 919–924. IEEE, 2001.

Brian Dobing and Jeffrey Parsons. Dimensions of UML diagram use: a survey of practitioners. *Journal of Database Management (JDM)*, 19(1):1–18, 2008.

Dalibor D. Dvorski. Installing, configuring, and developing with xampp. *Skills Canada*, 2007.

Fraida Fund, Shahram Shahsavari, Shivendra Panwar, Elza Erkip, and Sundeep Rangan. Resource sharing among mmWave cellular service providers in a vertically differentiated duopoly. In *2017 IEEE international conference on communications (ICC)*, pages 1–7. IEEE, 2017.

Emine Gozcu and Cagda Kivanc Caganaga. The Importance of Using Games in EFL Classrooms. *Cypriot Journal of Educational Sciences*, 11(3):126–135, 2016.

Yufang Hou and Anette Frank. Analyzing sentiment in classical chinese poetry. In *Proceedings of the 9th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 15–24, 2015.

Thorsten Keuler, Jens Knodel, Matthias Naab, and Dominik Rost. Architecture Engagement Purposes: Towards a Framework for Planning" Just Enough"-Architecting in Software Engineering. In *2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture*, pages 234–238. IEEE, 2012.

Daniel Liang. *Introduction To Java Programming, Comprehensive Version, 7/E*. Pearson Education India, 2009.

Shuang Luan, Ling Gao, and Jing Li. In Commercial Operation Mode MySQL Advantage. *Computer Knowledge and Technology*, (11):13, 2010.

Scott MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Human-computer interaction*, 7(1):91–139, 1992.

Likoebe M. Maruping, Viswanath Venkatesh, and Ritu Agarwal. A control theory perspective on agile methodology use and changing user requirements. *Information Systems Research*, 20(3):377–399, 2009.

Don Norman. *The design of everyday things: Revised and expanded edition*. Basic books, 2013.

Stephen Owen. *The great age of Chinese poetry: the High T'ang*. Yale University Press New Haven, CT, 1981.

Janis Reilly and Nicole Zieme. Interaction design: beyond human-computer interaction. 2011.

Dennis M. Ritchie. The development of the C language. *ACM Sigplan Notices*, 28(3): 201–208, 1993.

Marco Schwartz and Stefan Buttigieg. *Arduino Android Blueprints*. Packt Publishing Ltd, 2014.

Yang Shulin and Hu Jieping. Research and implementation of Web Services in Android network communication framework Volley. In *2014 11th International Conference on Service Systems and Service Management (ICSSSM)*, pages 1–3. IEEE, 2014.

Bjarne Stroustrup. *The C++ programming language*. Pearson Education India, 2000.

Sarah J. Tracy. *Qualitative research methods: Collecting evidence, crafting analysis, communicating impact*. John Wiley & Sons, 2019.

Desireé Vega, Jaclyn N. Wolf, Dylan O. Barton, Michele Stathatos, Charlotte Iurino, and Lily Hammer. Identifying the training experiences and needs of bilingual school psychologists. *Psychology in the Schools*, 56(10):1687–1699, 2019.

Sarah Watts, Anna Mackenzie, Cherian Thomas, Al Griskaitis, Louise Mewton, Alishia Williams, and Gavin Andrews. CBT for depression: a pilot RCT comparing mobile phone vs. computer. *BMC psychiatry*, 13(1):49, 2013.

Luke Welling and Laura Thomson. *PHP and MySQL Web development*. Sams Publishing, 2003.

Chan Zhen Yue and Shum Ping. Voice activated smart home design and implementation. In *2017 2nd International Conference on Frontiers of Sensors Technologies (ICFST)*, pages 489–492. IEEE, 2017.

Li Zuixiong. Pigment Analysis On Tang Dynasty Murals at The Mogao Grottes. *Dunhuang research*, 4, 2002.

Appendix A

Testing

A.1 Login and registration

Table A.1: Test cases of login and registration

Requirement to test	**Action**	**Expected Result**	**(Y/N)**
The system should allow users to enter the login page as the application starts	Launch the application	Display a login interface	Y
The system should allow users to jump to the registration page	Click the registration button	Display the registration page	Y
The system should allow users to go back to the login page	Click the back button	Go back to the login interface	Y
The system should allow a user to enter a username to the required field	Click the text box for username	Allow the user to enter a username.	Y
The system should allow a user to enter a password to the required field	Click the text box for password	Allow the user to enter a password	Y

When user wants to login, the system should be able to check whether a user's information matches the information in the database and then enter the home page if it is matched	Click the login button	Show an error message if the user's information is incorrect or enter the home page if correct	Y
When user wants to register, the system should be able to check if a username has been registered and shows a message if the registration succeeds	Click the registration button	Show an error message if the username or the email has been registered or the prompt that the registration succeeds.	Y

A.2 Personal Center and Home

A.2.1 Home - Bottom navigation bar

Table A.2: Test cases of Home - Bottom navigation bar

Requirement to test	**Action**	**Expected Result**	**(Y/N)**
The system should allow user to switch between home, game, personal center	Click the icon in the navigation bar	Display the corresponding page, change the color of the clicked icon.	Y

A.2.2 Personal Center - main page

Table A.3: Test cases of Personal Center - main page

Requirement to test	**Action**	**Expected Result**	**(Y/N)**
The system should allow users to open the Personal Center	click the Me button	Display the personal center interface	Y
The system should allow users to choose their head portrait	Click the head portrait	Open the album to choose a portrait	Y
The system should allow users to open the setting page	Click the SETTING button	Display the setting interface	Y
The system should allow user to view the favorite page	Click the FAVOURITE button	Display the favorite page	Y
The system should allow users to view the download page	Click the DOWNLOAD button	Display the download page	Y
The system should allow users to return to Personal Center page	Press the arrow icon	Display the Personal Center page	Y

A.2.3 Personal Center - setting page

Table A.4: Test cases of Personal Center - setting page

Requirement to test	**Action**	**Expected Result**	**(Y/N)**
The system should allow users to return to Personal Center page	Press the arrow icon on the left top corner	Display the Personal Center page	Y
The system should allow users to view font size page	Click the FONT SIZE button	Display the font size page	Y

The system should allow users to view the background page	Click the CHANGE BACK-GROUND button	Display the back-ground page	Y
The system should allow users to open the night mode	Click the NIGHTMODE button	The application's theme will be changed into dark theme	Y
The system should allow users to turn off the night mode	Click the NIGHTMODE button again when the night mode has al-ready been opened	The application's theme will be changed back to white theme	Y
The system should allow users to open Eye Protection mode	Switch on the Eye Protection Mode button	The Eye Protection Mode will be turned on	Y
The system should allow users to turn off Eye Protection mode	Switch off the Eye Protection Mode button	The Eye Protection Mode will be turned off	Y
The system should allow users to change font size	Dragging the slider to change the font size and click Confirm button	The font size of the application will be changed globally	Y

The system should allow users to return to the setting page in the font size page	Press the arrow icon on the left top corner	Display the setting page	Y
The system should allow users to change background color	Click the background images to preview	The background color of this page will be change to the color chosen by the user	Y
The system should allow users to cancel the change of background color	Click the cancel button	The background color will be reset to white	Y
The system should allow users to return to the setting page in the background page	Press the arrow icon on the left top corner	Display the setting page	Y

A.3 Poem

A.3.1 Poem - poem page

Table A.5: Test cases of Poem - poem page

Requirement to test	**Action**	**Expected Result**	**(Y/N)**
The system should allow users to choose the difficulty level of poems	Click on "POEM LIST" button	Display the four difficulty levels for users to choose from	Y
The system should be able to display the poem-list page	Select one of the four difficulty levels	Display all poems belongs to the difficulty level that a user has chosen	Y

The system should allow users to search for Tang poems by themes or author names	Enter a key word in the search bar	Display all the poems contains the key word, or display nothing if all poems do not contain the key word	Y
The system should allow users to sort the list of poems by poem name	Click on the "POEM" button	Display all poems sorted by poem name	Y
The system should allow users to sort the list of poems by author name	Click on the "AUTHOR" button	Display all poems sorted by author name	Y
The system should allow users to choose the type of poems	Select one of the five kinds of poems	Display all poems belonging to the type chosen by the user	Y
The system should allow users to get into the poem page	Click the arrow to the special poem	Skip to the poem that corresponds to the beginning of the selected letter if exists or do nothing if not	Y
The system should allow users to view the translation of the poem	Click on the "TRANSLATE" button	Display all the translation	Y
The system should allow users to view the video of the poem	Click on the "VIDEO" button	1. Open the browser if such a video exists or pop up a window shows "there is no such a video"	Y

For a selected poem, the system should allow users to view the author‘s brief introduction and the background of the poem	Press on the bottom button and pull it up	Display the author‘s brief introduction and the background of the poem	Y

A.3.2 Poem - comment page

Table A.6: Test cases of Poem - comment page

Requirement to test	**Action**	**Expected Result**	**(Y/N)**
The system should allow users to go back to the poem page	Click the back button	Display the poem page	Y
The system should allow users to view all the comments	Slide the comment list	Display the all the comments	Y
The system should allow users to reply to a comment	Click the reply button in a comment	The keyboard should pop up and allow users to reply	Y
The system should allow users to enter comments to a poem	Press the text editor and enter the content	The keyboard should pop up and users are able to enter content	Y
The system should allow users to send the comment	Press the send button	Send the comment to the remote database, refresh the comment list and show the response message	Y

The system should allow users to delete his own comment	Press the delete button	Delete the comment in the remote database, refresh the comment list and show the response message	Y

A.4 Game

A.4.1 Game - Apple tree

Table A.7: Test cases of Game - Apple tree

Requirement to test	**Action**	**Expected Result**	**(Y/N)**
The system should allow users to start the Apple tree game	Click the button "APPLE TREE"	Display the homepage of apple tree game	Y
The system should allow users to return to Game page	Click return icon	Display the Game page	Y
The system should allow users to return to homepage of apple tree game	Click the "HOMEPAGE" button	Display the homepage of apple tree game	Y
The system should allow users to jump to next question	Click the "NEXT" button	Display the next question page	Y
The system should allow users to check their answers, if the answer is true, an animation will be shown; else a prompt message will be shown	Click the "SUBMIT" button	Display the corresponding animation or toast	Y

The system should allow users to drag an apple into a hole, and the Chinese character will be entered into the corresponding gap after this operation	Long click the apple and drag it to a specific place	After a long click, a shadow of an apple is shown. After the drag, the corresponding gap is filled	Y

A.4.2 Game - Tang Dynasty Story

Table A.8: Test cases of Game - Tang Dynasty Story

Requirement to test	**Action**	**Expected Result**	**(Y/N)**
The system should allow users to start the Tang Dynasty Story game	Click the button "TANG DYNASTY STORY"	Display the story game interface	Y
The system should allow users to return to Game page	Click return icon	Display the Game page	Y
The system should allow users to view the content of the last sentence or the next sentence	Click the up arrow or the down arrow	Display the corresponding sentence	Y
The system should allow users to select the options that determine the behavior of a role, thus starting a new plot	Select one of the options	Display a new corresponding plot	Y
The system should allow users to view the ending of the story	Select a option from the last question	Display the corresponding ending of the story	Y
The system should allow users to restart the game	Press the button "TRY AGAIN"	Display the story game interface	Y

The system should allow users to return to Game page	Press the button "BACK TO GAME"	Display the Game page	Y
The system should allow users to view the whole story	Press the button "CHECK WHOLE STORY"	Display the whole story script in a scroll view	Y

A.4.3 Game - Exercise

Table A.9: Test cases of Game - Exercise

Requirement to test	**Action**	**Expected Result**	**(Y/N)**
The system should allow users to start the exercise	Click the button "EXERCISE"	Display the exercise interface	Y
The system should allow users to return to Game page	Click return icon	Display the Game page	Y
The system should allow users to view the last and next question	Click the "PREVIOUS QUESTION" BUTTON or the "NEXT QUESTION" BUTTON	Display the corresponding question	Y
The system should allow users to select the answer from three options that below the question	Select one of the options	The selected option turns green	Y

The system should allow users to view hint information	Press the button "HINT"	If the question do not have a hint picture, display the hint information with an alert dialog, or if the question has a hint picture, display the hint information with an alert dialog and a picture next to the options	Y
The system should allow users to finish the exercise	Press the button "FINISH"	Display the accuracy of the exercise	Y
The system should allow users to review the wrong questions	Press the button "REVIEW WRONG QUESTIONS"	Display the review questions mode interface and just display the wrong questions. The selected options will be marked, if a question is answered correctly, the selected option will be green; otherwise, the selected option will be red, and the correct answer will be green. Display the explanations of questions	Y

The system should allow users to review the all questions	Press the button "REVIEW ALL QUESTIONS"	Display the review questions mode interface and display the all exercise questions. The selected options will be marked, if a question is answered correctly, the selected option will be green; otherwise, the selected option will be red, and the correct answer will be green. Display the explanations of the questions	Y

Appendix B

Minutes

B.1 Minutes 2019.10.15

Meeting:

Date: 2019.10.15

Time: 15:00-16:00

Place: DB C09

Chair: Zhiyu ZHAO

Secretary: Xinyang LIU

Present: Zhiyu ZHAO, Mingjia ZHANG, Shengyi ZHOU, Yunpeng GUO, Zhongyi WANG, Xinyang LIU

Main points:

1. The time for formal meeting: Friday afternoon: 3:00 to 3:30

Template: Aim and points

2. Agenda: the point you want to discuss, should be sent early before the meeting

3. Minutes: Based on the agenda. Laying the action points (for the coming week)

4. Detail of the project:

A. Learning tang poetry, giving foreigners chances to learn this, functions: on how you teach others.

ex: features, voice, songs, games, exercises

B. Focus on how to teach others.

C. Basic interface like other software.

D. Choosing: select which function the user can choose, default or random With an introduction page for selecting.

E. Survey and research: the existing software for reference

Which function you think should be add.

F. Website: brief introduction, team name, agenda, plans.

G. Report: introduction, motivation, the requirement specification, design, implementation, tasking evaluation and summary (what good and what not good, issues, future work, conclusion).

H. Handbook: Interim report, marking details.

I. Roles in the team: report, interface. . . so on.

Need to know responsible for what thing.

J. Importance of teamwork: group members (advantages, cooperation), the experience is important as well. Link to others' work: divide task into different persons.

K. The software type: desktop or mobile is both OK.

5. About the preparation:

A. Email: send copy to other group members.

B. If want more meeting in one week: send the email.

C. Agenda: everyone should give their points to it.

Action points:

1. First 2 weeks: look at existing tools (maybe 6). Providing: name, link, one paragraph to introduce.

2. Need: why a good function? How to improve it. Think as a user first. (What the main feature and attraction).

3. Writing down what you really want to do: (requirement).

4. Next week tell the experience about the tools.

Next meeting:

Date: 2019.10.25

Time: 3.pm

Place: SEB 449

Chair: Xinyang LIU

Secretary: Shengyi ZHOU

B.2 Minutes 2019.10.25

Meeting:

Date: 2019.10.25

Time: 15:00-16:00

Place: PMB 449

Chair: Xinyang LIU

Secretary: Zhongyi Wang

Present: Zhiyu ZHAO, Mingjia ZHANG, Shengyi ZHOU, Yunpeng GUO, Zhongyi WANG, Xinyang LIU

Main points:

1. Report the content of Wednesday informal meeting

1.1 It is necessary to have a questionnaire, the content of the questionnaire should be considered, such as the learning time and the client... etc.

1.2 show some picture of the function to the supervisor

1.3 Some ideas about the application

A. The user can design their own learning cycles, like you can learn the Tang poem for one year or more. (In this way, we should take how the foreign learn about the poem into account)

B. We found the translated text of three hundred tang poem, we can use it as our translation.

C. We think that teaching the foreign how to pronounce the Chinese character is necessary. However, how to do this algorithm is difficult, so this function is optional.

D. We talked about the interface we image, and we can have the main topic to let the user choose what they want. We have the poem‘s the background, exercise for the poem and random choice for the answer.

E. What if the foreigners cannot understand the verse? We can have the translation of the word one by one.

F. We consider about the video of the whole poems, but we think this idea is impossible to achieve, so it is an optional function

G. We shall add PinYin to the sentence, it is optional, we shall pay more attention to the translation.

H. We shall add some fundamental Chinese into the application.

2. This week, we shall think about the learning outcome by the questionnaire, how to get the performance.

3. About the website, before tomorrow evening, submit the personal state to the wechat. About the project, we can use the brief introduction on the website.

Action points:

1. Questionnaire: learning stylelearning outcome, ask some foreigners

2. Get the user requirement after get learning outcome.

Next meeting:

Date: 2019.11.1

Time: 3.pm

Place: SEB 449

Chair: Zhongyi Wang

Secretary: Mingjia Zhang

B.3 Minutes 2019.11.01

Meeting:

Date: 2019.11.01

Time: 15:00-16:00

Place: SEB 449

Supervisor: Heshan DU

Chair: Zhongyi WANG

Secretary: Mingjia ZHANG

Present: Zhiyu ZHAO, Mingjia ZHANG, Yunpeng GUO, Zhongyi WANG, Xinyang LIU

Absence: Shengyi ZHOU (GRE Test)

Main points:

1. Show the website that we have already set up.

1.1 Check the copyright of pictures. (should put link)

2. Preparation work of interview:

2.1 Brief introduction about the project and Tang poetry

2.2 Prepare some slides, pictures, videos, audios and games about Tang poetry to show what is Tang poetry.

2.3 Learn how to interview.

2.4 Consent form.

3. Identify interview questions.

3.1 A brief introduction to the project.

3.2 Add “Do you know about Tang poetry?” at the beginning of interview and show some slides, videos or other thing.

3.3 Modify No1. “More detail (time and experience)”.

3.4 Modify No4. “If there is a software to learn tang poetry, do you want to use it on the phone or the computer?”.

3.5 Modify No5. “Through which way do you hope to learn tang poetry? (video, audio,

games, exercises, . . .)".

3.6 Modify No8. "How much time would you like to spend on learning tang poetry?"

3.7 Add at the end of interview "Do you have any question/suggestion?".

4. Detail of interview.

4.1 All interviewees should sign the consent form before interview.

4.2 The number of interviewees: 5-10 (can be teachers and students).

4.3 All group members should participate in the first interview to be familiar with all steps of interview.

5. User and system requirements.

5.1 User and system requirements can be modified base on the result of interview.

Action points:

1. Check the copyright of pictures that in our website. Put the link on the website.

2. Interview:

2.1 Learn how to do interview (books in library, videos on Internet).

2.2 Prepare slides, pictures, videos, audios and games about Tang poetry. (Not need all forms)

2.3 Prepare consent form.

3. Show user and system requirements in next meeting.

Next meeting:

Date: 2019.11.08

Time: 3.pm

Place: SEB 449

Chair: Mingjia ZHANG

Secretary: Yunpeng GUO

B.4 Minutes 2019.11.08

Meeting:

Date: 2019.11.08

Time: 15:00-16:00

Place: SEB 449

Supervisor: Heshan DU

Chair: Mingjia ZHANG

Secretary: Yunpeng GUO

Present: Zhiyu ZHAO, Mingjia ZHANG, Yunpeng GUO, Zhongyi WANG, Xinyang LIU, Shengyi ZHOU

Main points:

1. Show the work that we have done this week: e-mail, ppt slides, user/system requirements.

2. Functional requirement implement (divide into necessary parts or optional parts, core functions, show priority).

Non-functional requirement implement (deadline of specific tasks, programming language, respond time, i.e. when user send a command, how quick the sys would respond).

(About functional and non-functional requirement, refer to SE reference book and get some basic understanding, could be 2-3 pages in A4, can be included in interm and final report)

3. Check with interviewee whether 10 poems for gaming part is necessary.

4. Category poems into different themes for searching. (add this function)

5. Prototype: implement some main functions and interface.

6. User requirements modification:

a. shall -¿ system should allow user to... b. to be more general (logic: define the requirement-¿ build the software rather than define the requirement based on an imaginary existing software.

System requirement modification: (to be more general, summary each function into one or two sentences. e.g.)

b. system requires to keep record of history -¿ sys provide favorites module.

c. system allows the user to enter username, password -¿ the system allows the user to login.

d. reading module: the system allows the user to select a page; select a specific poem to read.

7. Record the functions in existing version of user/system requirement, use them in the interm/final report and the following development of software.

8. Try one or two interviews firstly and find if it is needed to modify consent form/questionare.

Action points:

1. Modify user and system requirements.
2. Interview and record.
3. Get start with developing the prototype.
4. Get start with writing the interm report.
5. If it is needed to book a room, write an e-mail to inform our supervisor in advance.

Deadline:

1. Send an e-mail containing the draft of interm report by 4th Dec (should include: introduction of this project and initial design (interface, main function).
2. User/system/functional/non-functional requirement modification before next meeting

Next meeting:

Date: 2019.11.15

Time: 3.pm

Place: SEB 449

Chair: Yunpeng GUO

Secretary: Zhiyu ZHAO

B.5 Minutes 2019.11.15

Meeting:

Date: 2019.11.15

Time: 15:00-16:00

Place: SEB 449

Supervisor: Heshan DU

Chair: Mingjia ZHANG

Secretary: Yunpeng GUO

Present: Zhiyu ZHAO, Mingjia ZHANG, Yunpeng GUO, Zhongyi WANG, Xinyang LIU, Shengyi ZHOU

Main points:

1. Finished modifying system requirements

1.1 The second one: haven't introduced user guide (start with "system should allow user to . . . " (register login) (provide user a guidance))

1.2 The third one: The system should provide users all tang poems and allow them to select from them to read.

1.3 Combine 3.2 3.3

1.4 Rephrase point 6

1.5 Point 4 can be: For the selected poem, system should allow user to switch translation

1.6 Rephrase point 7 (introduce daily poetry) Revise the requirement and send the email.

2. Allocated the tasks of interim report.

2.1 Time plan: draw the Gantt chart

3. Prototype design: use Android Studio. Learn the use of the Android Studio.

4. Interview more people

5. What is the meaning of prototyping: implement the user interface and show the code, make simple functions work but don't need all function.

6. Design: what the GUI should look like, separate the design with the prototype

Also: put on the diagrams

7. Put on one Screenshot and one paragraph to introduce the system.

8. For existing system, provide the reference link of the existing system.

9. Use Latex to write the report

Search for the templates of report

10. Requirement: Compulsory; Desired function: talk about desirable functions

11. Put the interview in background information and research, technical research

12. Introduction: why the project is interesting. Two pages. Talk about background of tang poetry. Motivation: how this would interest the users.

Report DDL: 11/28

Next meeting:

Date: 2019.11.22

Time: 3.pm

Place: SEB 449

Chair: Xinyang LIU

Secretary: Shengyi ZHOU

B.6 Minutes 2019.11.22

Meeting:

Date: 2019.11.22

Time: 15:00-16:00

Place: SEB 449

Supervisor: Heshan DU

Chair: Zhiyu ZHAO

Secretary: Xinyang LIU

Present: Zhiyu ZHAO, Mingjia ZHANG, Yunpeng GUO, Zhongyi WANG, Xinyang LIU, Shengyi ZHOU

Main points:

1. The use case diagram

1.1 The composing of the diagram should be easier for looking, the start point should be set on the top left or top right corner.

1.2 The arrows are not easy to understand, so we should add cutline to explain these different arrows.

1.3 Put on one paragraph to introduce the diagram.

2. Setting

2.1 The setting content can include eye protecting mode

3. Interface:

3.1 Add the reading poem part

3.2 using the JavaFX to implement the design.

DDL of this is 11.29 (Next Friday).

4. Game: the game should be designed more interesting and should has scores as well.

5. Interview more people. (Long-term)

6. The time to provide interim report to Heshan should be as soon as possible.

7. Interim report's separated work: DDL of this is 11.27 (Next Wednesday).

Report DDL: 11/28 (Last week decided).

Next meeting:

Date: 2019.11.29

Time: 3.pm

B.7 Minutes 2019.11.29

Meeting:

Date: 2019.11.22

Time: 15:00-16:00

Place: SEB 449

Supervisor: Heshan DU

Chair: Xinyang LIU

Secretary: Shengyi ZHOU

Present: Zhiyu ZHAO, Mingjia ZHANG, Yunpeng GUO, Zhongyi WANG, Xinyang LIU, Shengyi ZHOU

Main points:

1. The aspects that need to be improved in introduction

1.1 Need to make the content more related to the design.

1.2 More general ideas need to be discussed, such as background story, the motivation, and potential users.

1.3 Do not contain too much technical details and problems and try to give the reader ideas about what will be discussed next.

2. The aspects that need to be improved in background

2.1 Change the application's name.

2.2 Citations need work.

3. The aspects that need to be improved in marketing research

3.1 Interview report needs to be included.

4. The aspects that need to be improved in requirement part

4.1 More technical tools need to be introduced, such as scenes builder.

4.2 Show the use case diagram.

4.3 The interface of the design needs some improvements.

5. The aspects that need to be improved in timetable

5.1 Move the timeline to the end.

Next meeting:

Date: 2019.11.29

Time: 3.pm

Place: SEB 449

Chair: Shengyi ZHOU

Secretary: Zhongyi WANG

B.8 Minutes 2020.02.18

Meeting:

Date: 2020.02.18

Time: 10:30-11:00

Place: Online

Supervisor: Heshan DU

Chair: Zhiyu ZHAO

Secretary: Zhiyu ZHAO

Present: Zhiyu ZHAO, Mingjia ZHANG, Yunpeng GUO, Zhongyi WANG, Xinyang LIU, Shengyi ZHOU

Summary of the meeting:

In this meeting we will mainly introduce what we have done since the winter break started.

Main points:

The Tang Poetry App is developed by Android Studio, and the team is divided into four parts: Game part, Personal center part, Database building part.

1. Game part:

Question sheet: Improve the questions for more English-friendly. Think about the English speakers. For questions: Consider what you are testing; Who are the main users. Not meeting Chinese speakers but English speakers. Both game and question sheet add more English.

2. Personal center part:

Link this part with other part. E.g. downloading part with database. Collaborate with database.

3. Database part:

Format of poem: In 1st video: in time 4 second, 7 second, translation is overlapping with the text.

Provide the video and audio.

One of requirement: add the background information.

Tang poems:

How many poems are we going to add?

Focus on some of the poems, focus more on the mechanisms.

Focus on some famous poems and provide background information and more details.

Database:

Searching part: Connect it with real poems. Combine it with Zhongyi.

Next thing to do: connect different part and more functions.

Can it run on real phone.

Action Points:

There are some remaining tasks now for the team members, and here is the arrangement for the remaining tasks:

Zhao Zhiyu: Realizing the table switching function for each part (Home, Game, Me); Come up with new idea of game.

Guo Yunpeng: Fixing some bugs on apple tree, beautify the interface; Add English version of question; Provide the example code of long click function.

Zhang Mingjia: Connecting the game questions to the database; Fixing bugs on the exercise; Change into English version for user.

Liu Xinyang: Realizing the function of changing text style, background and changing the head profile; Connect to the database.

Wang Zhongyi: Make the poetry list better and classify the poetry in to separate classes.

Zhou Shengyi: Research on how to add profile pictures to the database; create the page of daily poetry.

Next meeting:

Date: 2020.02.25

Time: 10.30 am

Place: Online

Chair: Zhiyu ZHAO

Secretary: Zhiyu ZHAO

B.9 Minutes 2020.02.25

Meeting:

Date: 2020.02.25

Time: 10:30-11:00

Place: Online

Supervisor: Heshan DU

Chair: Zhiyu ZHAO

Secretary: Zhiyu ZHAO

Present: Zhiyu ZHAO, Mingjia ZHANG, Yunpeng GUO, Zhongyi WANG, Xinyang LIU, Shengyi ZHOU

Summary of the meeting:

In this meeting we will mainly introduce the work we have finished during last week and discuss the new idea for the app.

Main points:

1. Add to favorite function: How to add? By modifying the function and storing them in the string file and save it in the database.

2. Random background/fixed background. If make use of the background, then design the background, otherwise, background can be random.

3. Question sheet: Hint? E.g. figures? Add some hints.

4. Change the font size, background. Add a preview function. Night mode: Still in progress. Add another function to add: Eye protection mode.

5. Daily poetry: needs database, connect to the database. Not clear about how to show like or dislike. This week it will connect to the database.

6. Searching bar: problem: only can search the first time of the key word, it will not show the second time. If user type in two words, what will happen. Trying to add search button. Another thing: what if I want to search the keyword in the different themes?

(Try to search in separate area) Another: keyword is only related to the title? Title and author name. Keyword should also be searched in content.

In the algorithm, using a loop to get all the keywords.

Action Points:

1. Double check with the function requirement.
2. Add some videos/ audios.
3. Talk about background, provide good view.
4. Add more games.
5. Try to connect different parts together.

Next meeting:

Date: 2020.03.02

Time: 10.30 am

Place: Online

Chair: Zhiyu ZHAO

Secretary: Zhiyu ZHAO

B.10 Minutes 2020.03.02

Meeting:

Date: 2020.03.02

Time: 10:30-11:00

Place: Online

Supervisor: Heshan DU

Chair: Zhiyu ZHAO

Secretary: Zhiyu ZHAO

Present: Zhiyu ZHAO, Mingjia ZHANG, Yunpeng GUO, Zhongyi WANG, Xinyang LIU, Shengyi ZHOU

Summary of the meeting: In this meeting we will mainly introduce the work we have finished during last week and introduce the switching of our focus of work.

Main points:

1. Liu Xinyang: Connect the work with the base.

2. Zhang Mingjia: One small thing: Q1 provided 3 choices, different choices are written in Chinese. Should add the translation. Involve some foreigner. Hold some interviews. Not a good type of question to ask. Think about designing questions for foreigners. Think about what you really want to evaluate. Software to help learn tang poetry. What are main learning objectives? If learning Chinese words is an objective.

3. Wang Zhongyi: Additional function looks nice. Small point: Put some citation.

4. Zhou Shengyi: Connect with Zhongyi's database. Successfully run the app in real mobile phone.

Action Points:

1. Good to start focus on the poem page, try to think about how to teach tang poem. Focus on some part of poems. Think about how to teach it to foreigners. How to use

different figures, songs, games. It will be good if focus on some 3-5 poems to start with.

2. For some of the poem it is easier. Classify the tang poem based on the difficulty levels.

3. Design this as a game. If going well, users can choose some more difficulty. Marking system. If it is difficult, gain more marks. Play with the software. Design some game/-maze. Limitation of the time.

Next meeting:

Date: 2020.03.10

Time: 10.30 am

Place: Online

Chair: Zhiyu ZHAO

Secretary: Zhiyu ZHAO

B.11 Minutes 2020.03.10

Meeting:

Date: 2020.03.10

Time: 10:30-11:00

Place: Online

Supervisor: Heshan DU

Chair: Zhiyu ZHAO

Secretary: Zhiyu ZHAO

Present: Zhiyu ZHAO, Mingjia ZHANG, Yunpeng GUO, Zhongyi WANG, Xinyang LIU, Shengyi ZHOU

Summary of the meeting:

In this meeting we will mainly introduce the work we have finished during last week and talk about the learning outcome we want the software to be, also next step of the project.

Main points:

1. Game part: Finished the game. Try the demo later.

2. Commenting part: Working on the reply function. Many tables to edit.

3. Poem part: Optimize the code to improve the efficiency.

4. Setting page: eye protection part is finished.

5. Audio part: How to get mp3 file. Get copyright, download the mp3. Found the way to add the audio.

6. Story: When do the conversation, is there any audio (music).

7. Another thing: Provide the link of the video, click to jump to the video.

Features: e.g. music software, first use it, ask which kind of song they like. Ask the level. What do they expect to learn? (Level system)

Action Points:

1. Combine all parts.

2. Have some demonstrations. Demonstration using the app. User guide. Provide 5 poems. Show how the software works. Preparation for the open day.

3. Unify the theme of the application.

Next meeting:

Date: 2020.03.17

Time: 10.30 am

Place: Online

Chair: Zhiyu ZHAO

Secretary: Zhiyu ZHAO

B.12 Minutes 2020.03.17

Meeting:

Date: 2020.03.17

Time: 10:30-11:00

Place: Online

Supervisor: Heshan DU

Chair: Zhiyu ZHAO

Secretary: Zhiyu ZHAO

Present: Zhiyu ZHAO, Mingjia ZHANG, Yunpeng GUO, Zhongyi WANG, Xinyang LIU, Shengyi ZHOU

Summary of the meeting:

In this meeting we will deliver the work finished last week, and discuss about the final report, also some questions about the interim report.

Main points:

1. Next to create database to each poem. The database name is based on the name of the poem. First thing is to make sure that the order is matched.

2. Test with users, see how they will use it. When university opens, it's good to have some interviews to test whether they like the functions, and how much they like them.

3. Can users add video link?

Provide users some root to provide useful information etc.

4. Open source database, e.g. map app is created by normal user, openStreetMap.

5. Report: Divide different part to different teammates. Set the framework. If want to get some early feedback, could send to email.

Action Points:

1. Combine all parts of the application.

2. Get started with the final group report.

3. Unify the theme of the app.

Next meeting:

Date: 2020.03.24

Time: 10.30 am

Place: Online

Chair: Zhiyu ZHAO

Secretary: Zhiyu ZHAO

B.13 Minutes 2020.03.24

Meeting:

Date: 2020.03.24

Time: 10:30-11:00

Place: Online

Supervisor: Heshan DU

Chair: Zhiyu ZHAO

Secretary: Zhiyu ZHAO

Present: Zhiyu ZHAO, Mingjia ZHANG, Yunpeng GUO, Zhongyi WANG, Xinyang LIU, Shengyi ZHOU

Summary of the meeting:

In this meeting we will deliver the work finished last week and discuss about some points of final report as well as individual report.

Main points:

Work finished last week:

1. Combined all the individual part of application into a whole project. Some bugs exist, will be fixed soon.

2. Each team member has written the implementation part of the final report.

Advice and tasks:

1. Need to write motivation and objective in the introduction part: tell the readers why it is an interesting problem. Talk about aim. It is important to make aim and motivation very clear.

2. Focus on new parts. Finish new parts first, for example, implementation and evaluation.

3. Individual report: add citation, reference list. If used third party platform, should add the reference.

Action Points:

1. Integrate the implementation part as soon as possible.
2. Revise the interim report.
3. Send the final report before 2nd April.

Next meeting:

Date: 2020.04.07

Time: 10.30 am

Place: Online

Chair: Zhiyu ZHAO

Secretary: Zhiyu ZHAO

B.14 Minutes 2020.04.07

Meeting:

Date: 2020.03.24

Time: 10:30-11:00

Place: Online

Supervisor: Heshan DU

Chair: Zhiyu ZHAO

Secretary: Zhiyu ZHAO

Present: Zhiyu ZHAO, Mingjia ZHANG, Yunpeng GUO, Zhongyi WANG, Xinyang LIU, Shengyi ZHOU

Summary of the meeting:
In this meeting we will update the progress we have made (mostly on the final report). Also, we will rise some new questions for answers.

Main points:

1. User manual: The User manual will be helpful. Readme file for the software.
2. When submitting the software: Source code software are required.
3. In report: provide the hierarchy of the software. Provide some description to explain the structure.
4. In diagram, levels were not deepest, not need to add all. Is it needed to list all component: Just show the structure is fine. Provide more detailed explanation i.e. description about the figure.
5. In the implementation part, show a few lines of src code, start to explain the meaning of each line. Think about which part is most important. Think about the src code, some of the code are from the api. Talk about which is written by you, which is from the api.
6. Once finished writing, send via email.
7. For testing, can do black box testing, also junit, or refer to software engineering.

Action Points:

1. Finish revising the separate parts of final report.
2. Start writing test cases send to Wang Zhongyi.
3. Write the user manual and Readme file for the application.

Next meeting:

Date: 2020.04.14

Time: 10.30 am

Place: Online

Chair: Zhiyu ZHAO

Secretary: Zhiyu ZHAO

B.15 Minutes 2020.04.14

Meeting:

Date: 2020.04.14

Time: 10:30-11:00

Place: Online

Supervisor: Heshan DU

Chair: Zhiyu ZHAO

Secretary: Zhiyu ZHAO

Present: Zhiyu ZHAO, Mingjia ZHANG, Yunpeng GUO, Zhongyi WANG, Xinyang LIU, Shengyi ZHOU

Summary of the meeting:

In this meeting we will mainly discuss about the revise suggestions of the report. Also, we will talk about things left to do.

Main points:

1. Discussion during the meeting:

(a) Format: using Latex. Be careful about the reference list. For bib format, edit using JabRef. In the text, use ‘

cite’.

(b) Reflection part: Do we need to specify/write the solutions? About the issues, write about the issues already solved.

(c) Background part: About programming language: provide a survey to mention some more languages that are available to us.

(d) Design part: when talk about design, don’t have system yet. For the design: provide some use case diagrams. Design is more general, does not contain the implementation. Move the gui into the implementation part. Talk about structure of the system. Talk about software architecture based on the initial plan. How plan to realize the function/in-

terface. Talk about the algorithm. Talk about main concepts. Implementation: how to actually realize it. Refer some textbook about software design. UML to design the structure. Use case diagram. Provide something you have before coding. Provide high level description.

(e) On latex, for code, not use figure, but use verbatim. So, show the actual code, use verbatim (or upgrade the resolution of the figure).

(f) Currently, the code in the implementation: in this section, the code is a bit too much. Think about in which order shall we provide and why do I need to present this part of the code. Talk about the most important content.

(g) About the source code: provide a readme file.

(h) Presentation: how do we show the application? Prepare a ten minutes presentation. Need to record the small video about how to run the software.

(i) Presentation:

Description of the problem.

Overview of the application.

Reflective remarks on the project.

At least 3 members to do the presentation.

Action Points:

1. Complete the final report.
2. Prepare presentation.

Next meeting:

Date: 2020.04.21

Time: 10.30 am

Place: Online

Chair: Zhiyu ZHAO

Secretary: Zhiyu ZHAO

B.16 Minutes 2020.04.21

Meeting:

Date: 2020.04.21

Time: 10:30-11:00

Place: Online

Supervisor: Heshan DU

Chair: Zhiyu ZHAO

Secretary: Zhiyu ZHAO

Present: Zhiyu ZHAO, Mingjia ZHANG, Yunpeng GUO, Zhongyi WANG, Xinyang LIU, Shengyi ZHOU

Summary of the meeting:

In this meeting, we mainly discussed the plan and some questions for the coming deadlines. Also, we updated the progress we have made so far. The upcoming work is covered in the meeting as well.

Main points:

Discussion during the meeting:

(a) For the consent form, don't include the personal information.

(b) Format needs to be improved.

(c) Code 12. Implementation of the search method.

(d) Reference: uniform the format.

(e) Presentation: follow the handbook, divide into three part, include the demonstration.

1. Problem description, explain background, clear the object.

2. Describe the software, do a demonstration.

3. Reflect upon the management and technical issues.

4. Follow Dave's email.

(f) At least 3 speakers.

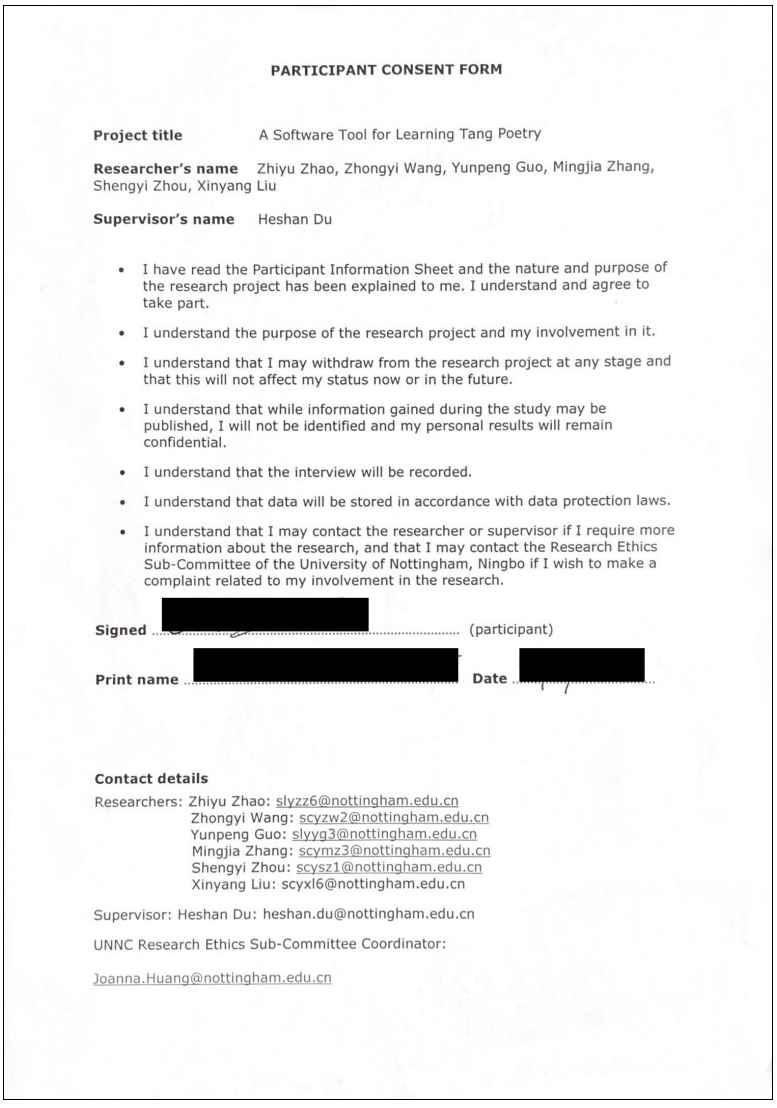
Action Points:

1. Complete the final report.
2. Prepare presentation.

Appendix C

Interview Materials

C.1 Participant Consent Form



PARTICIPANT CONSENT FORM

Project title A Software Tool for Learning Tang Poetry

Researcher's name Zhiyu Zhao, Zhongyi Wang, Yunpeng Guo, Mingjia Zhang, Shengyi Zhou, Xinyang Liu

Supervisor's name Heshan Du

- I have read the Participant Information Sheet and the nature and purpose of the research project has been explained to me. I understand and agree to take part.
- I understand the purpose of the research project and my involvement in it.
- I understand that I may withdraw from the research project at any stage and that this will not affect my status now or in the future.
- I understand that while information gained during the study may be published, I will not be identified and my personal results will remain confidential.
- I understand that the interview will be recorded.
- I understand that data will be stored in accordance with data protection laws.
- I understand that I may contact the researcher or supervisor if I require more information about the research, and that I may contact the Research Ethics Sub-Committee of the University of Nottingham, Ningbo if I wish to make a complaint related to my involvement in the research.

Signed [redacted] (participant)

Print name [redacted] **Date** [redacted]

Contact details

Researchers: Zhiyu Zhao: slyzz6@nottingham.edu.cn
Zhongyi Wang: scyzw2@nottingham.edu.cn
Yunpeng Guo: slyyg3@nottingham.edu.cn
Mingjia Zhang: scymz3@nottingham.edu.cn
Shengyi Zhou: scysz1@nottingham.edu.cn
Xinyang Liu: scyxl6@nottingham.edu.cn

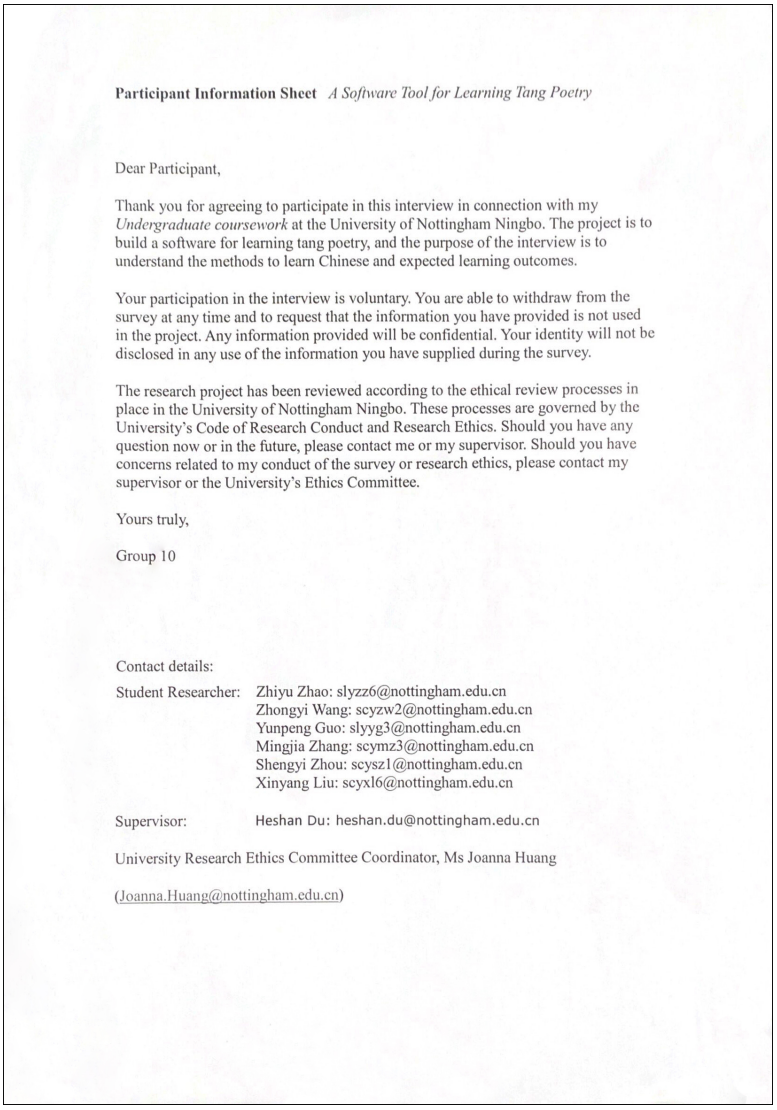
Supervisor: Heshan Du: heshan.du@nottingham.edu.cn

UNNC Research Ethics Sub-Committee Coordinator:

Joanna.Huang@nottingham.edu.cn

Figure C.1: Participant Consent Form

C.2 Participant Information Sheet



Participant Information Sheet *A Software Tool for Learning Tang Poetry*

Dear Participant,

Thank you for agreeing to participate in this interview in connection with my *Undergraduate coursework* at the University of Nottingham Ningbo. The project is to build a software for learning tang poetry, and the purpose of the interview is to understand the methods to learn Chinese and expected learning outcomes.

Your participation in the interview is voluntary. You are able to withdraw from the survey at any time and to request that the information you have provided is not used in the project. Any information provided will be confidential. Your identity will not be disclosed in any use of the information you have supplied during the survey.

The research project has been reviewed according to the ethical review processes in place in the University of Nottingham Ningbo. These processes are governed by the University's Code of Research Conduct and Research Ethics. Should you have any question now or in the future, please contact me or my supervisor. Should you have concerns related to my conduct of the survey or research ethics, please contact my supervisor or the University's Ethics Committee.

Yours truly,

Group 10

Contact details:

Student Researcher: Zhiyu Zhao: slyzz6@nottingham.edu.cn
Zhongyi Wang: scyzw2@nottingham.edu.cn
Yunpeng Guo: slyyg3@nottingham.edu.cn
Mingjia Zhang: scymz3@nottingham.edu.cn
Shengyi Zhou: scysz1@nottingham.edu.cn
Xinyang Liu: scyxl6@nottingham.edu.cn

Supervisor: Heshan Du: heshan.du@nottingham.edu.cn

University Research Ethics Committee Coordinator, Ms Joanna Huang

(Joanna.Huang@nottingham.edu.cn)

Figure C.2: Participant Information Sheet

Appendix D

User Manual

D.1 Homepage

Enter the Poem Page

1. Touch the view poems button to enter the poem page.

2. Touch the back button  to go back to the homepage.

Enter the Daily Poetry Page

1. Touch the Daily poetry button to enter the poem page.

2. Touch the back button  to go back to the homepage.

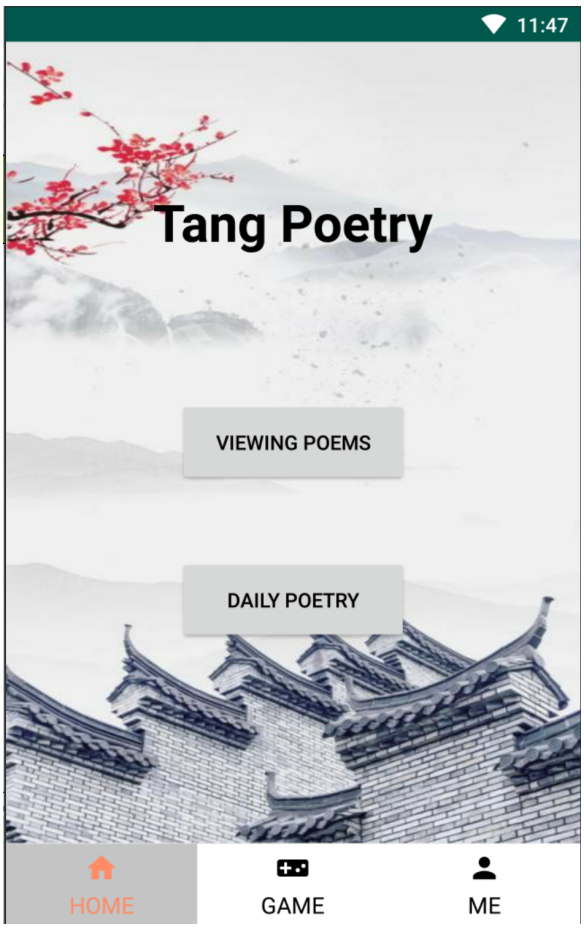


Figure D.1: Homepage

D.2 Difficulty Selecting Page

Choose Difficulty

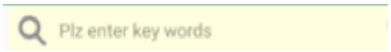
1. Touch the button to choose the difficulty according to the label.

2. Touch the back button  to go back to the homepage.

Figure D.2: Selecting Difficulty page

D.3 Poem List Page

Search for Poem

1. Touch the search bar  to pop out the keyboard.

2. Input the keyword.

3. Touch the search button  to start searching.

Change the Order Method for Poem

1. Touch the Author button  to reorder by the author's name from A to Z.

2. Touch the Poem button 

to reorder by the poem name from A to Z.

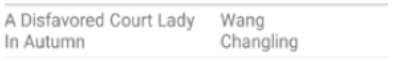
3. Touch the Back button  to return to difficulty selecting the page.

Change the category for Poem

Touch the button in the top right section 

to show the corresponding theme of poems.

Go to the inner poem page

Touch the poem name bar (e.g. 

)to enter the poem page.

Go to the Corresponding Letter Beginning Poem

Touch the letter on the rightmost bar to jump to the poem begin with the corresponding letter.

Figure D.3: Poem List page

D.4 Poem Page

Watch video

Touch the video button 

to watch the poem video.

Audio

Touch the audio button 

to listen to the poem audio.

Comment

Touch the comment button 

to enter the comment page.

Translate

Touch the translate button 

to translate it into the English version.

Learn More about Poem

Drag the Learn More bar up 

to show the background information of the poem and introduction of the author.

Figure D.4: Learn More in the Poem page

D.5 Daily poetry

1. Touch the daily poetry button.
2. Swipe to the left or right to switch the poetry.

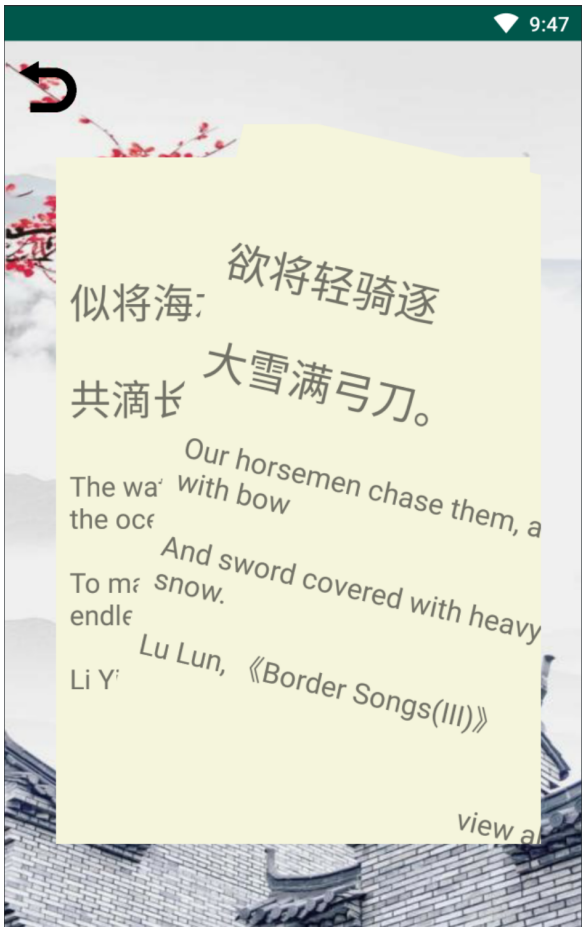


Figure D.5: Swipe the card to switch poetry

D.6 Game

Apple Tree

1. Touch the Apple Tree button APPLE TREE on the Game page.

2. Touch he Game Start button GAME START! to start the game.

3. Touch the Return icon to return to the Game page.

4. Based on the question that is given on the top of the screen, drag an apple with a character (e.g. ) onto the space of tree (e.g. ) where you want to add it.

   The character of the selected apple will be displayed in the corresponding position of the question.

   The same apple can be selected repeatedly.

5. Touch the Submit button SUBMIT to check the answer.

   If the answer is wrong, the prompt "Wrong answer, please try again." will be shown at the bottom of the page;

   If the answer is correct, the prompt "Correct!" will be displayed at the same place and the apples on the tree will grow bigger.

6. Touch the Previous button PREVIOUS to return to the last question.

   If you press the Previous button on the interface of the first question, the page will not have any response.

7. Touch the Next button NEXT to jump to the next question.

   If you press the Next button on the page of the last question, the Ending page (Figure D.6) will be displayed.

8. Touch the Back To Game button to return to the Game page.

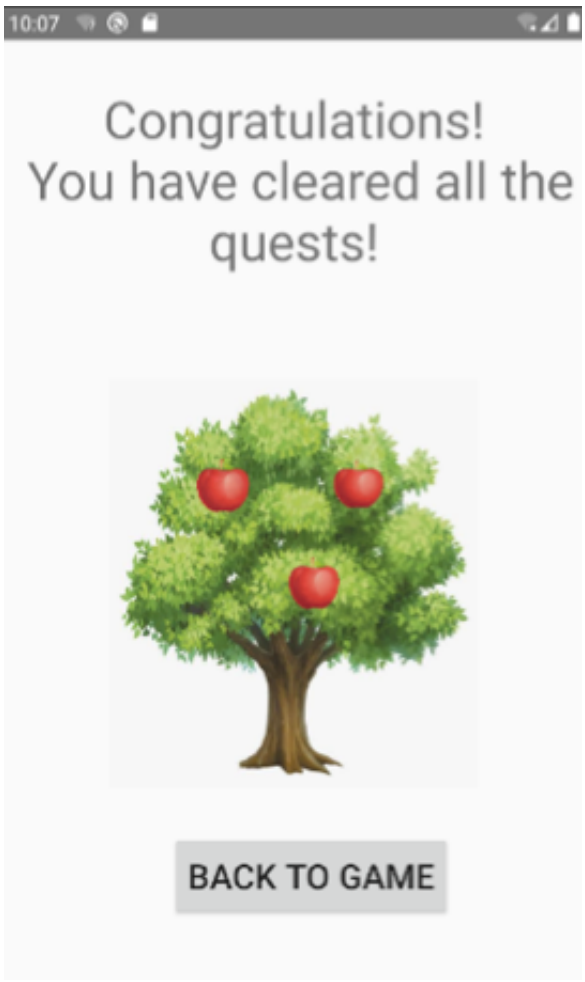


Figure D.6: Apple Tree Ending Page

Tang Dynasty Story

1. Touch the Tang Dynasty Story button 

on the Game page.

2. Touch the Return icon  to return to the Game page.

3. Touch the icon  to view the previous sentence.

4. Touch the icon  to view the next sentence.

5. Select one of the options (e.g. 

)to continue the game.

6. At the end of the game, the ending will be displayed at the top of the page (Figure D.7).
7. Touch the Try Again button to restart the game.
8. Touch the Back To Game button to return to the Game page.
9. Touch the Check Whole Story button to view the complete story.

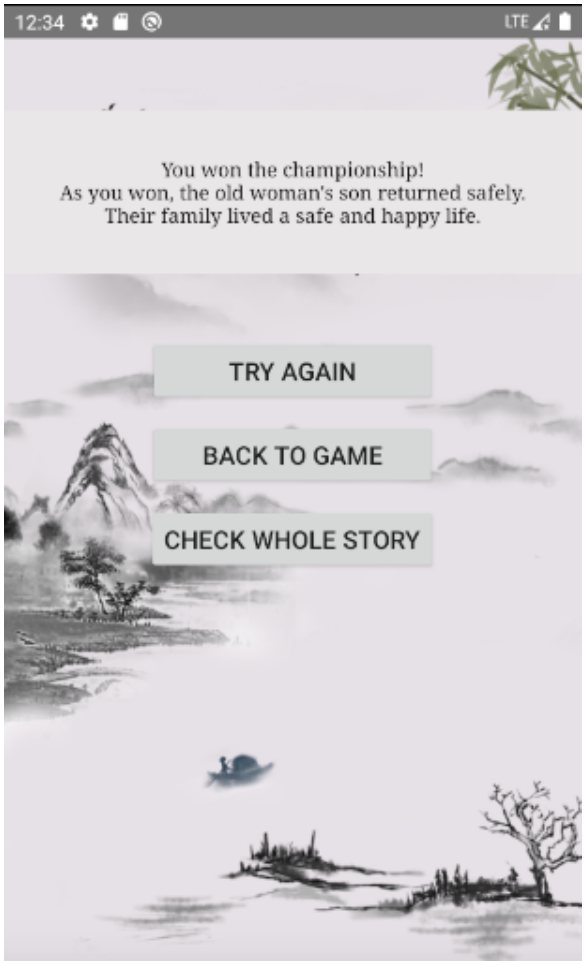


Figure D.7: Tang Dynasty Story Ending Page

Exercise

1. Touch the Exercise button EXERCISE on the Game page.

2. Touch the Game button GAME to return to the Game page.

3. Touch the Hint button HINT to view the hint information.

   The hint information will be displaced by the alert dialog (Figure D.8). Touch the Confirm button to close the dialog.

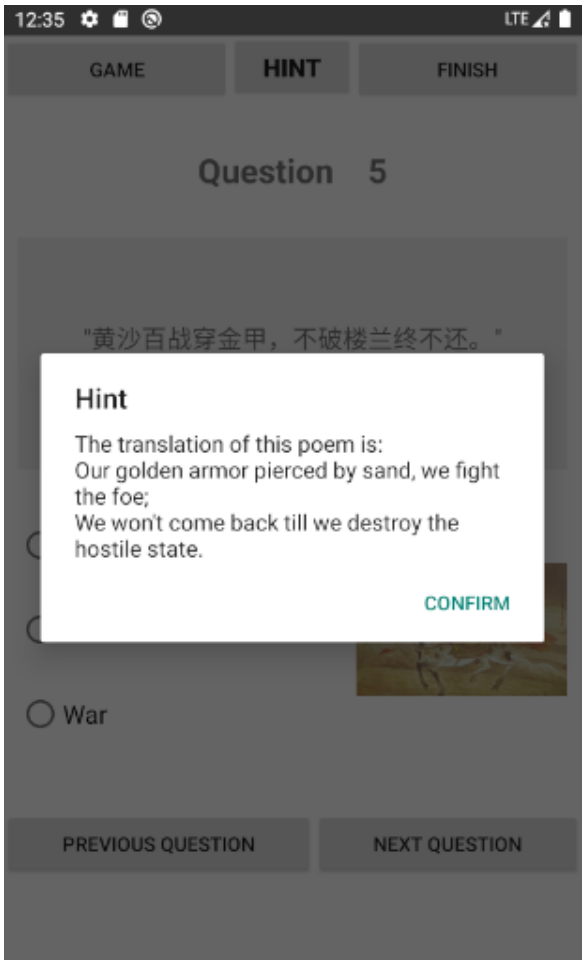


Figure D.8: Hint Information with an Alert dialog

4. Select one of the options to answer the question.

5. Touch the Previous Question button PREVIOUS QUESTION to view the previous question.

   If the question is already the first question, the reminder information "This is the first question." will be presented.

6. Touch the Next Question button to jump to the next question.

   If you press the Next Question button PREVIOUS QUESTION on the final question, the reminder information will ask you "Are you sure you want to submit your answers?" (Figure D.9).

   If you press the Confirm button, the page will jump to the Result page (Figure D.10);

   If you touch the Cancel button, the page will stop on the interface of the last question.

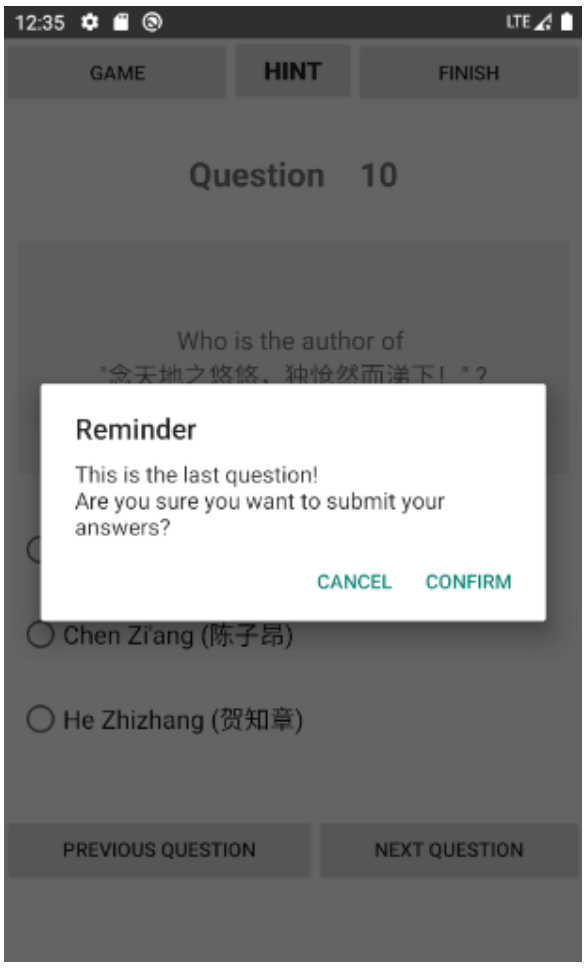


Figure D.9: Reminder information on the page of the last question

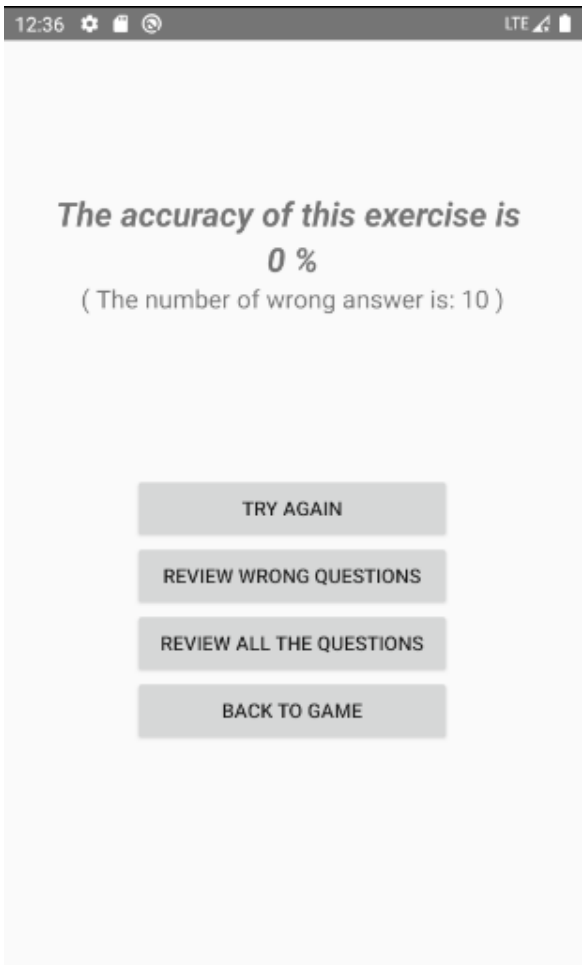


Figure D.10: Exercise Result page

7. Touch the Finish button FINISH to finish the exercise and jump to the Result page.

8. Touch the Try Again button TRY AGAIN to restart the exercise.

9. Touch the Review Wrong Questions button REVIEW WRONG QUESTIONS to review the wrong questions.

10. Touch the Review All The Questions button REVIEW ALL THE QUESTIONS to review all the exercise questions.

11. View the explanation (Figure D.11) of each question by the review modes (review the wrong questions and review all questions).

    Touch the Review Wrong Questions button or Review All The Questions button in each review mode to jump to the corresponding mode.

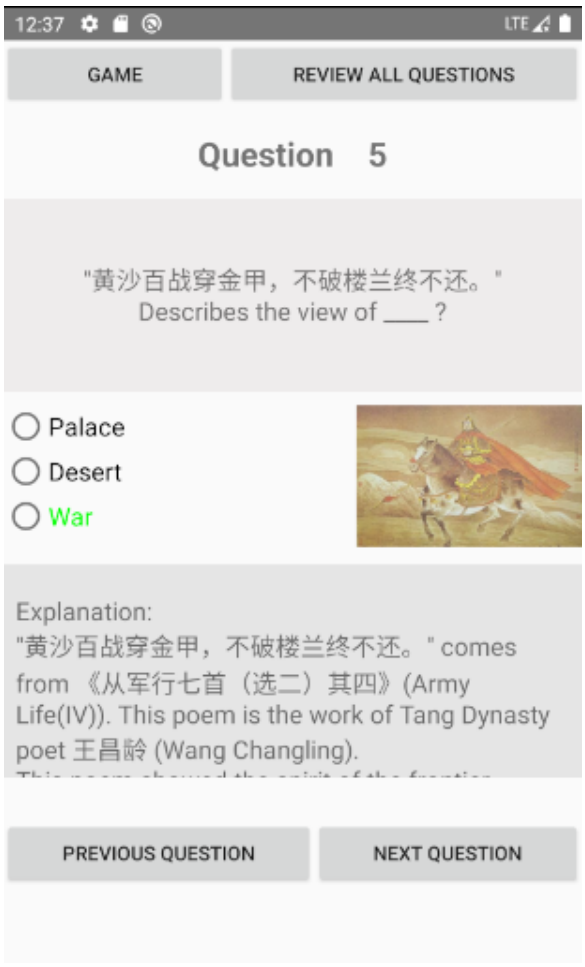


Figure D.11: The Mode of Reviewing the Wrong Questions

12. Touch the Back To Game button BACK TO GAME to return to the Game page.

D.7 Settings

Font size

1. Touch the font size button in the setting page.
2. Drag the slider to switch the font size.

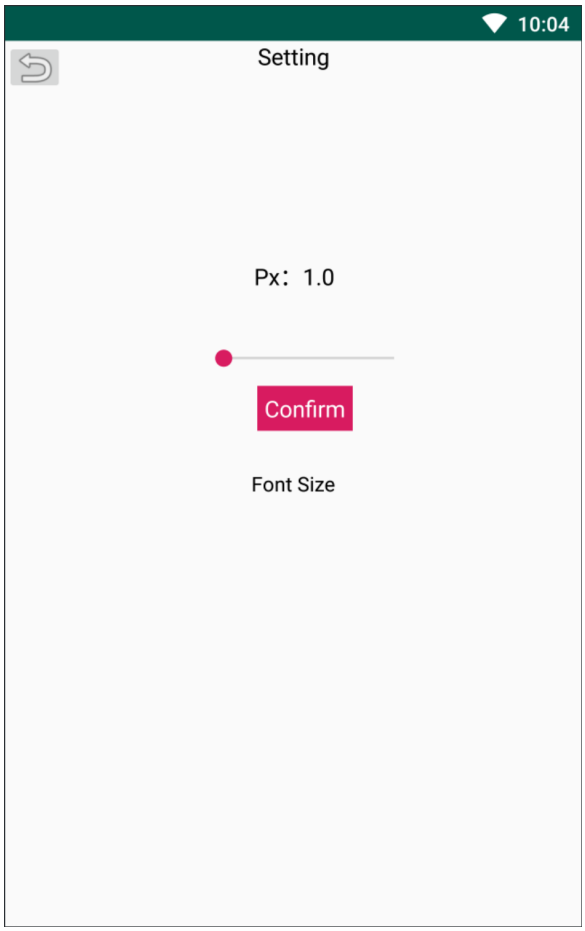


Figure D.12: Change font size

Background

1. Touch the change the background button on the setting page.
2. Touch the chosen background to change the background.

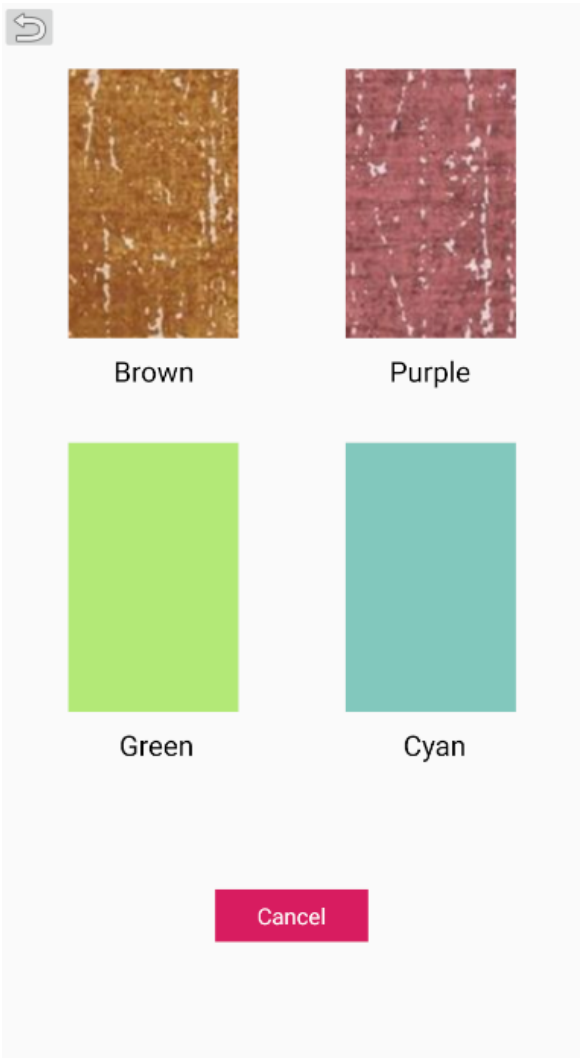


Figure D.13: Change background

Night mode

Touch the night mode button to switch to night mode.

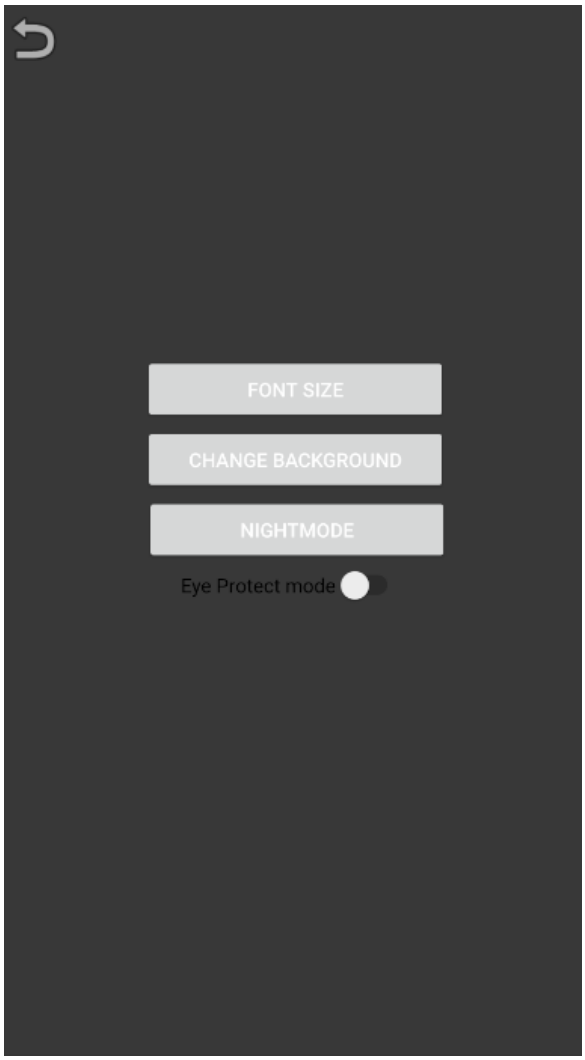


Figure D.14: Night mode

Eye protect mode

Touch the switcher next to the eye protect mode label in the setting page.

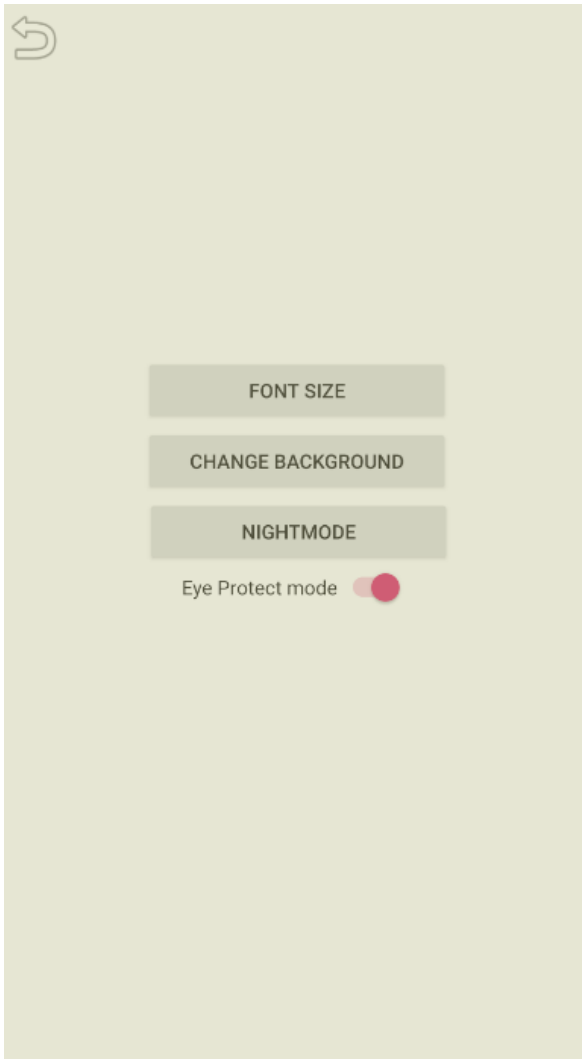


Figure D.15: Eye protect mode