How Knowledgeable Are You ?

System Design Document
V1.0

28th of April 2023

Kaan Eker

Prepared for
COMP4902 Graduation Design Project

Table of Contents

SYSTEM DESIGN DOCUMENT [1]

1. Introduction

How knowledgeable are you is a mobile game application. In this game application, you can measure your own knowledge, play for fun in your spare time, play to learn new information. In the game, you are asked questions on the topic you choose and your score increases in line with the questions you know. You can use different one-right jokers for questions you do not know or have guesses. The game is an application that is very easy-to-use and provides competition , knowledge and entertainment to the user.

1.1. Purpose of the System

The use of phone games has increased considerably nowadays. People waste a lot of time these days. While waiting for something somewhere, using public transport on the road, during work breaks, when there is nothing to do at home, and in many other places. How knowledgeable are you here for them. It allows people to spend time without getting bored. It provides entertainment to people with its constantly different and new questions, questions on different subjects, and its point system, which intensifies the competition among people. Even if they have a job, they can take a break from the game and then continue.

1.2. Design Goals

Performance

Responses to requests are given between one and six seconds after they are accepted.
System has a minutely capacity of ten tasks.
1 Gigabytes of memory are used by the system.
System should control internet connection every 5 sec.

Dependability

When a user submits incorrect data, the system displays errors.
To ensure that the system is bug-free, unit testing is used.
System can achieve common tasks below 30 seconds.
When system crashes , a report sent to the developers.

Cost

The cost of developing a system is free. Every part of system was created using open-source programs like Unity and MySQL Firebase.

Deployment cost for the system is free. It is available for free download from the Android Google Play Store.

Maintenance

The system created by using the MVC Architecture , making modifications simple and error-free.

Considering that the system was created for Android Devices so adaptation is simple. Since my deployment target is Android , my application can use by only android or in windows with emulator. But it is not available for other environments such as Ios and Windows itself.

CamelsCase is the format used for system design and coding. Codes also include remark explanations. Because the system is programmed in the C# IDE with Unity, it is simple to map the code of the specific requirements.

End User Criteria

The system is easy to use for users. The user interface has explanations and the buttons are easy to access. The system's font has been adjusted for easy readability. The questions in the system are arranged in a way that everyone can understand.

1.3. Definitions, Acronyms, and Abbreviations

SDD: System Design Document

App: Application

MVC: Model View Controller

AI: Artificial Intelligence

Android: Android Operating System

UI: User Interface

2. Proposed Software Architecture

2.1. Overview

This system is developed on Unity , C#. C# is a very effective and innovative paradigm, according to Android Game operating system. The UI is created using Visual Studio IDE. For database, system uses MySQL. Database and the system communicate with HTTP requests with PHP XAMPP. Since the system is android mobile game application, it uses MVC pattern as architecture. The three main layers of this architecture are as follows. There is UI in the View layer. The Controller layer contains key operations. Data storage is the model layer's responsibility.

2.2. System Decomposition

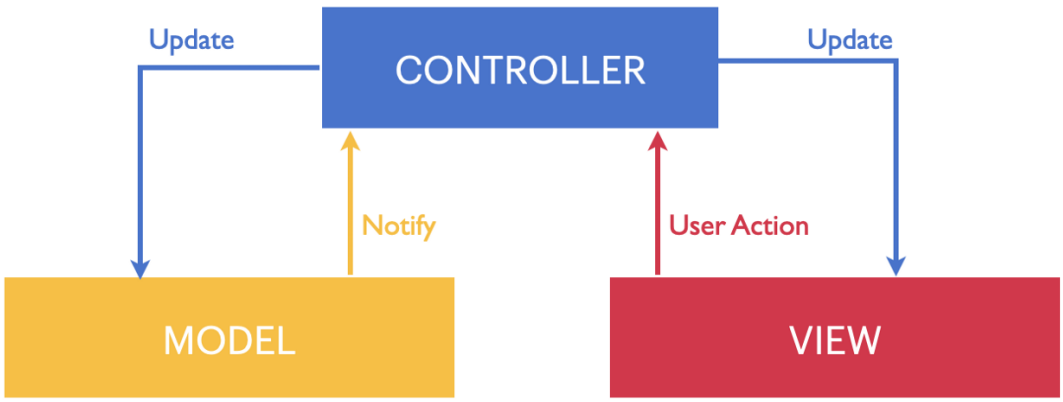
A typical software design paradigm for implementing user interfaces, data, and controlling logic is MVC (Model-View-Controller). It emphasizes a division between the business logic and appearance of the software. This "separation of concerns" allows for a better labor-force distribution and better upkeep.

The Model is the area where the data to be used in the developed project is kept. Model, in short, is the modeling of concrete objects that we use in our daily life in the software industry.

The View is what user can see on data. The view is aware of how to obtain model data, but it is unaware of the meaning of that data or how a user might be able to alter it.

The Controller provides coordination between Model and View in the game application. It is the control section where all operations (various calculations, data exchange, database operations, etc.) in the developed projects are made.

On my system the user selects an action from the UI in the View layer. The selected action that calls the interfaces for the selected action is sent to the Controller layer. The Controller layer updates the Model layer via interfaces. The Model layer notifies the Controller layer whether the update was successful or not. If the update is successful, the Controller layer updates the view shown to the user via the UI.



[2]

2.3. Hardware/Software Mapping

This diagram shows deployment of How Knowledgeable Are You ? game application. The Application runs only on android and windows with emulators. The How Knowledgeable Are You ? application may run different many devies. Application communicates MySQL database with HTTP requests via PHP and XAMPP.

2.4. Persistent Data Management

Persistent data management refers to the storage and retrieval of data that persists beyond the lifetime of a game session or application. In the case of a Unity game that uses MySQL and PHP, the game need to store information such as player profiles, game progress, or in-game purchases. Using MySQL and PHP for persistent data management in a Unity game can provide a scalable and reliable way to store and retrieve game data.
There are various advantages of MySQL database, such as;

Scalability : Can handle large datasets , scalable to meet the needs of growing applications.
Speed : Fast performance , can handle high speed transactions
Flexibility : Can be used in various environments
Reliability : Can ensure data integrity and availability.
Cost-Effective: Its free to use so I don’t need any money to deployment.

In our system every static data will be stored in the database.

2.5. Access Control and Security

To use current system , Visitors must set their nickname to system. All users can play the game in all features.

Access Control

	Settings	PlayerClass
Player	<<create>> ChangeNickname() seeLeaderboard() adjustAudio() adjustTheme() deleteAccount()	Login() QuitGame() stopGame() restartGame() displayScore() useJokers() setAudioIngame() pickGenre() selectAnswer()

Security

How Knowledgeable Are You ? Game Mobile Application , since our program does not store any private information of the user other than nickname, I do not need great protection, but certain protections are provided in the database to protect the source code of the program.

Since the applications needs a database, I am using MySQL. MySQL offers us range of built-in security features that can help protect your data from unauthorized access. These include user authentication and access control, encryption, and secure connections.

2.6. Global Software Control

Global software control service for system is that it only works on Android and Windows with Android Emulators running devices such as Samsung , Huawei phones , so there won't be any synchronization or concurrency issue. If visitors want to use our system, they need to have internet connection and an Android running device.

Users send requests to the system database and system replies to related questions to access or store static data to database. As replies to the requests from users, the system replies to it with corresponding services or continues listening to the next requests.

Global Access Table

- Player,Settings <<create>>
- Player, Settings, ChangeNickname()
- Player, Settings, seeLeaderBoard()
- Player, Settings, adjustAudio()
- Player, Settings, adjustTheme()
- Player, Settings, deleteAccount()
-Player, PlayerClass, Login()
-Player, PlayerClass, QuitGame()
-Player, PlayerClass, stopGame()
-Player, PlayerClass, restartGame()
-Player, PlayerClass, displayScore()
-Player, PlayerClass, useJokers()
-Player, PlayerClass, setAudioIngame()
-Player, PlayerClass, pickGenre()
-Player, PlayerClass, selectAnswer()

2.7. Boundary Conditions

Initialization:
How Knowledgeable Are You ? is an android mobile application. It needs to be downloaded and installed from the Android Store on an Android running device. An internet connection is required for the download.

An internet connection is also necessary to use the application. To access the program, users must give it a nickname; otherwise, they will be unable to.

The user will be sent to the home page and given full access after logging in.

Termination:
Users can delete their accounts and quit games using buttons on the settings page of the application. Use the "Change Nickname" button if the user wants to alter their nickname.

If the user wants to stay signed in but wants to close the application, they will simply exit to the main page of their device and they can terminate the app running session.

Failure:

If there are any crashes or issues, a pop-up message will appear for the user. If the connection between MySQL and the system is broken, the user will be informed, asked to restart the application, and their session will be terminated. The system makes an effort to fix the problem on its own. If the program crashes, the user will need to restart the session.

3. Subsystem Services

The MVC model has three subsystems. Model, View, and Controller are them.

The Model part consists of two components. And they are Settings and Player Models. Settings Model holds Settings data and Player Model holds Player's data. The Model part comunicates controller with Settings and Player Manager.

The Controller part consists of six components. They are Login, Profile, Game, Leaderboard, Settings Controller. Login, Game and Settings Controller connected with Player Manager to reach Player Model.
Settings , Main and Leaderboard Controller connected with Settings Manager to reach Settings Model. Login Controller updates Login View with loginViaNickname. Game Controller updates gameQuestion , pickGenre , selectAnswer , stopGame , quitGame , restartGame , useJokers. Main Controller updates Main View with leaderboard , settings and playGame. Controller updates profile view with profile.

User Interface is a part of the View portion. Six additional components make up this component. Login, Leaderboard, Profile, Game, and Settings View are the options. These are the views of the application itself, How Knowledgeable Are You? The View Layer allows the application to receive user input.

4. References

1. https://www.lucidchart.com/pages/uml-class-diagram
2. https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm
3. **https://sam-16930.medium.com/unity-game-architectures-part-1-dc53b3c7307d**
4. **https://app.assembla.com/wiki/show/Melange/Global_software_control**