

**香港中文大學**

**計算機科學及工程學系**

**Department of Computer Science and Engineering,**

**The Chinese University of Hong Kong**

**DevTour**

version no.: 1.1.0

date: Feb. 14, 2021

**Group ID: A5**

**CHANG Chirui 1155124553**

**DING Baizeng 1155124617**

**LYU An 1155124488**

**YU Yue 1155124490**

Supervised By

**Prof. LYU Rung Tsong Michael**

©2020 The Chinese University of Hong Kong

The Chinese University of Hong Kong holds the copyright of this proposal.

Any person(s) intending to use a part or whole of the materials in the thesis

in a proposed publication must seek copyright release from the University.

Contents

[1. INTRODUCTION 3](#_Toc64413770)

[1.1. Project Overview 3](#_Toc64413771)

[1.2. Objective 3](#_Toc64413772)

[1.3. Expected Customers and Market 4](#_Toc64413773)

[1.4. System Features 4](#_Toc64413774)

[2. BACKGROUD 6](#_Toc64413775)

[3. SPECIFICATION 6](#_Toc64413776)

[3.1. Registration and login system 7](#_Toc64413777)

[3.2. Game subsystem 7](#_Toc64413778)

[3.3. Chat board subsystem 9](#_Toc64413779)

[3.4. Administrator System 10](#_Toc64413780)

[3.5. Overall view 10](#_Toc64413781)

[4. SYSTEM ARCHITECTURE 12](#_Toc64413782)

[4.1. Architecture 12](#_Toc64413783)

[4.2. System Components 12](#_Toc64413784)

[4.3. Description of Major System Components by UML 12](#_Toc64413785)

1. INTRODUCTION
   1. Project Overview

The project, named “DevTour”, aims to build a platform for students, especially programming beginners, to play mini games to experience as a developer and to communicate with each other. Users can log in their accounts and continue to play in their original game archive, and also leave their own comments and walkthroughs in the chat board. Our vision is to let users, especially novices in programming, understand the experience of being a software developer and engineer through the story of the game, and to create a platform for beginners and masters to communicate and discuss.

The development cycle of the whole project is about three months, and the developers include CHANG, Chirui, DING Baizeng, LYU An and YU Yue. We will use web-based access client-server architecture to implement the project, which may involve front and end technologies, such as HTML5, CSS3 jQuery, Ajax, Node.js, ReactJS, MongoDB and so on.

This report provides high-level descriptions on some functionalities, features and architectural designs. It also introduces the project background, architecture diagrams and brief descriptions of some key components of the system.

* 1. Objective

The overall goal of the project is to build a multi-functional network platform for the majority of programming developers, which integrates game playing, entertainment, career planning and social interaction. We find that most computer science students or beginner programmers are unable to map out their career plans clearly, and it is difficult to find mentoring opportunities and guidance with experienced developers, therefore, “DevTour” aims to provide such a platform.

First, our game mechanic is that the player has to answer a series of questions about the developer's career path, including any dilemmas they might encounter along the way. Due to the different choices, the player's health will be changed accordingly. The player's health is represented by four values: mental health, physical health, money, and academic status. The player's goal is to try to keep these values as balanced as possible through different choices, neither too high nor too low. When the four values are above or below a certain range, the round ends. Thus, the number of choices successfully completed is the player's score for the round. In this game, the player has to balance the four values by considering the possible consequences before making each choice, so the process of choosing is a process of experiencing a career as a developer. The first objective of the project is to give novice programmers the experience of being a developer through games.

Second, players can exchange ideas and share game tips with other players in the chat board. Users can post their reviews, thoughts on the game, and playthroughs to a chat board where other users can read and comment. Therefore, the second objective of the platform is to create a common discussion platform for developers – beginners or masters.

Overall, the objectives of this project are to provide a platform for developers, especially beginners, to explore their career and meet new friends.

* 1. Expected Customers and Market

The main target customers of “DevTour” are the programming beginners, especially for students whose major is related to computer science and engineering. Of course, for some experienced software developers, they can also know more beginners through this platform and give their own help. The most important thing is to let beginners learn more about the profession through the story of the game and make more friends on the way of the career.

For the market on this platform, we will first promote the software in colleges and universities, so that students learning programming will first join this platform. Next, according to the feedback of students, we will gradually improve some functions and gradually push it into the market.

The number of potential users is huge due to the large number of new programmers entering the developer profession. At the same time, as the popularity of programming technology is increasing year by year, almost every professional will be exposed to programming and software development at some point in the future, so there is a huge potential market to be explored. Of course, this project can continue to update and add the story in the game after launch to attract more players to join, and at the same time, VIP mechanism and hidden story can be added to make profits.

* 1. System Features

We divide the whole system into two subsystems, and users can click different buttons to enter the two subsystems. The first subsystem is the game experience system, and the other is the chat board system. Of course, the two systems share an account login mechanism and an account shares information in the two systems. In order to better understand the characteristics and operation mode of the whole system, we will introduce it in sections latter.

First, the user registration and login mechanism. New users can customize the username and password for registration, and the registration names of different users should be different. After registration, the user can use the registered account to log in. Each account independently stores the progress of the game and other data information.

Second, the game subsystem. After logging in, users can click into the game interface to play the game. The system automatically saves the game progress after each round of the game. The content of the game is to tell the story to the player and allow the player to make choices. Different choices have different consequences, so different choices affect the player's health. Player health value includes physical health, mental health, money and academic progress four values. Players need to balance their health by making choices that fit their current health. At the end of each round of the game, players break the balance of health, the system will record the number of questions successfully answered as the total score of the round of the game. At the end of each round, players are given feedback on the top 5 scores of the overall leaderboard. Players can also view their total progress, which is the total number of successful answers.

Third, the chat board system. After finishing a round of the game, or logging in again, users can click the corresponding button to enter the chat board system. The chat board feeds users comments from other users as well as comments from users to each other. Users can click the "Post" button to create their own comments and thoughts about the game, and they can also click the "Comment" button to reply to others' comments after reading them. Users can also delete comments and replies freely. The content of the chat board exists on the server side, so it ensures that the previous message is displayed every time the Web page is opened.

Finally, system maintenance and operation. System administrators have their own special account, login this account can modify the server database. The system administrator can create, traverse, modify and delete the user accounts to ensure the security. Administrators can also make changes to the database that stores stories and questions in the game, by creating new questions and answers, changing the consequences of each answer, and deleting outdated questions and answers. Finally, the administrator can also modify and delete the comments and messages in the chat board.

Our system will be designed to maximize user friendliness, robustness and reliability for all above features.

1. BACKGROUD

“DevTour” was originally designed to provide a platform for programming beginners to plan their own career by playing education simulation games, and to communicate with each other in a relatively relaxed environment. The ultimate purpose of this product is to provide a platform for beginners to discuss career planning issues and to find like-minded friends and common topics among developers. We hope that it can let science and engineering students escape the stereotype of “wearing a plaid shirt and boring” and become more socially active.

We found that most of the programming beginners, computer majors in the face of their coming into the society to seek work will enter a very confused state, sometimes cannot handle the work and study pressure. We hope that through the plot setting in the game, players can understand their love for programming learning and work, understand their strengths and weaknesses, and even understand the influence of their personality on future career planning. Because in the workplace, you may experience the annoyance of 996 work schedule, may experience the workplace PUA and so on, so our story design can start from there and give the player an early taste of what it's like to be a developer. Of course, not only the game, but we also hope that the programming developers can become a big collective, we can exchange thoughts and feelings in our platform, or find like-minded friends, or help other beginners to get started and so on.

Therefore, the most attractive feature of our product is the game that simulates real world problems, and the player can choose the answer freely to get different consequences. By answering real world questions, players can feel their own inner answers as well. In order to get higher score, the player is also forced to reply answers that are not what they really think, taking into account their current health. In fact, in real life, there are all kinds of dilemmas, and sometimes you need to consider the current situation and have to make a choice against your mind: this is the most attractive part of the game.

Of course, in addition to games, this product also has a social function, so it will bring more programmers and developers into the community to discuss career planning and share personal experiences with each other.

1. SPECIFICATION

In order to better illustrate the working mechanism of the various subsystems of the project, we introduce the subsystems with several data flow diagrams as follows:

* 1. Registration and login system

After opening the website, the page will first display the login interface. Users can select the registration or login button to register a new account or log in an existing account. If the user chooses to “register”, you need to first enter new account username and password, the system compares with the account number in the player database and administrator database to check whether the username already exists, if this user account has been registered, it returns an error message, if the username is available, it updates player database and shows registration information. If the user selects “login”, the user first need to enter account and password, the system will first check from the administrator database, if successful, returns the administrator interface and shows “successful login” message. Otherwise, the system will check from the player database, if successful, returns the player interface and shows “successful login” message, otherwise it returns an error message. The following diagram is the DFD of the registration and login system:

图片包含 图示

描述已自动生成

* 1. Game subsystem

When the user successfully logs in, the user enters the player interface. If the user chooses to start the game at this time, the user enters the game interface. When starting the game, the system will first read the user's save from the game archive database, if no save or finished recovering the save, it will enter a round of game. In each round of the game, the system checks four health values: physical health, mental health, money and academic status are within a certain range. If the health value exceeds a certain range, the game round ends, the system records the number of questions answered in this game round and records it into the total score of the user and updates the game progress of the player in the game archive database. If the health is within a certain range, the game continues, the player will be asked one question and make a choice, the system updates the health based on the player's choice, and again checks to see if the game is over. Based on the DFD of the user registration and login system, we can draw the following game DFD:

图形用户界面

描述已自动生成

As the following FSM shows, our system mainly has 4 states: start, wait, check and game over. When the game start, it firstly in the start state, and then asks the player a question and goes to the wait state, after receiving the answer of the player, it goes to check state to check whether the game is over. If yes, it goes to the game over state, an accepted state and game over; otherwise, it goes to the start state and continue the game:

图形用户界面, 文本, 应用程序

描述已自动生成

* 1. Chat board subsystem

After logging in, the user can click the specific button to send a request to enter the chat board. After sending the request, the web page displays the interface of the chat board, and the system reads all the comments stored in the comment database from the back end and displays them to the user. At this point, the user can send the request of POST comment and input the comment he wants to send. The system will update the content in the comment database and update the display. Based on the DFD drawn above, the following diagram adds the DFD of the chat board system:

有遥控器和文字

中度可信度描述已自动生成

* 1. Administrator System

The system keeps several accounts as administrator accounts. If you successfully log in an administrator account, you can enter the administrator interface for back-end database management. After the user enters the account password, the system will confirm whether it is an administrator. If it's an administrator, it goes into management mode. Administrators can click different buttons to send different administrative requests. The administrator needs to input the changed data information at the same time after sending the management data request. The system will change the relevant data content in the corresponding database according to the request in the back end. The following is the DFD of the administrator system (For clarity, extraneous elements have been removed from this diagram):

图表, 图示

中度可信度描述已自动生成

* 1. Overall view

The following FSM diagram describes the general structure and flow of the whole system, different circles represent different states of the system.

First, the system is in the main interface waiting for the user to register or log in. When the user chooses to register, the system enters the registration interface. After the user enters the registration account and password, the system will check availability. If successful, it will return to the login interface, otherwise, it will return to the registration interface.

After the user logs in, the system will check whether it is a player account or an administrator account. If the account is invalid, return to the login interface. If it is an administrator account, the system will enter the administrator interface, and the administrator can request management data. After entering the update interface, the system will update data and return to the administrator interface after updating. Administrators can also log out.

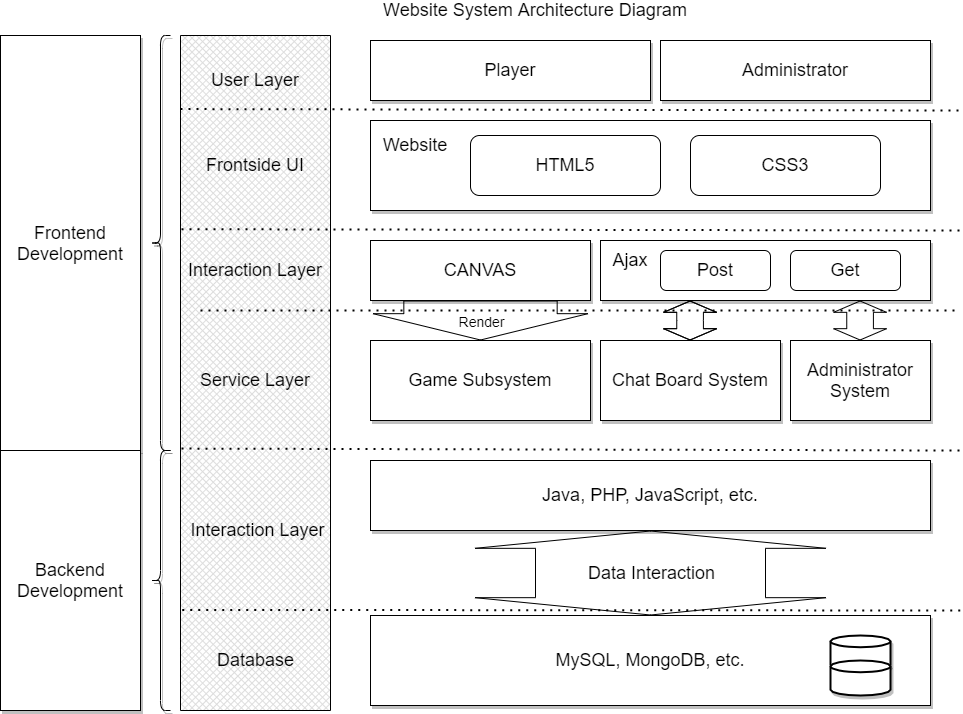
If the account is a valid player account, enter the player interface. The player can choose to enter the chat board or the game interface, which is similar to the process described in 3.2. If the player chooses to enter the chat board, the system can receive comments from the player and update the chat board and backend database. Users can also back to the player interface or log out directly.

背景图案

描述已自动生成

1. SYSTEM ARCHITECTURE
   1. Architecture
      1. Website System Architecture Diagram

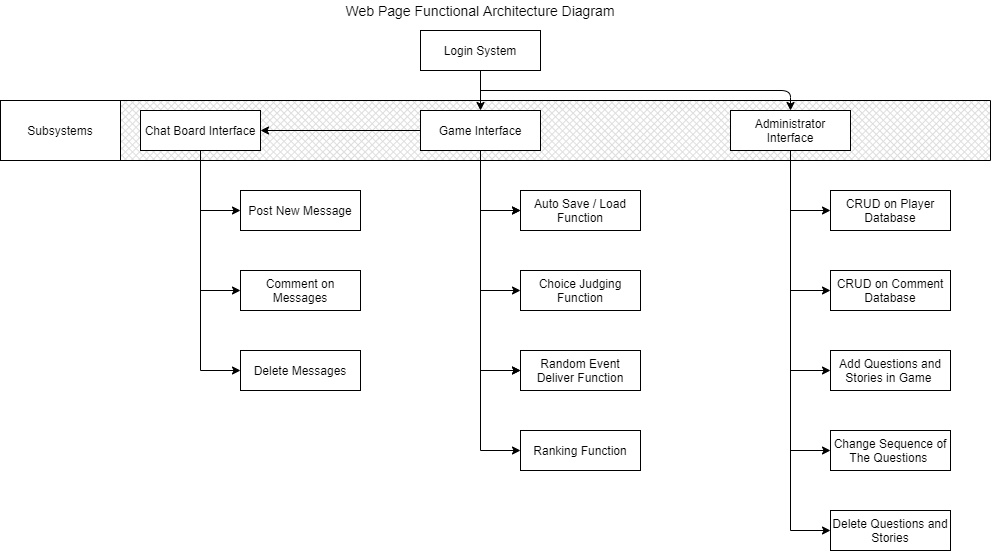
The system diagram is drawn intended to demonstrate the overall structure of the application. As our program is shown in form of web application, some general web development tools are used in the program. In front-end development, HTML5 and CSS3 are needed to construct the web page and JavaScript is used to build the logic of game and other subsystems. In back-end development, tools like Java and PHP are used to realize the interaction with database. The overall framework is shown in the following figure.



System Architecture Diagram

* + 1. Website Function Architecture Diagram

The diagram indicates the function each module shall consist of in high level description.



* 1. System Components
  2. Description of Major System Components by UML
     1. Use-case Diagram

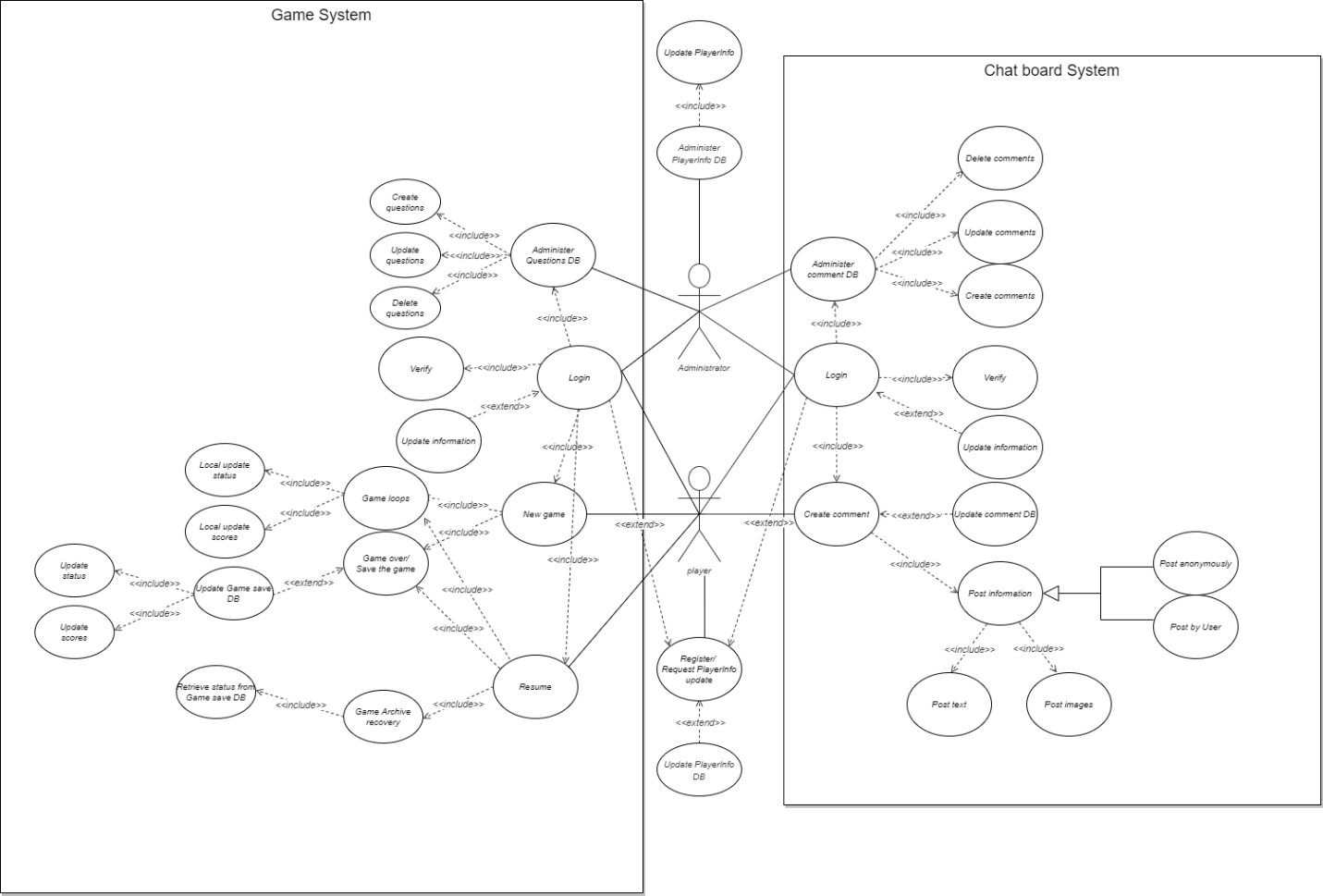
In this section, we will use a use-case UML diagram first to give a high-level overview of the relationships in our system. For better description of major system components, it is necessary to demonstrate the interactions between users and the system. Furthermore, to better illustrate the specification requirements of this system, there is no doubt that a use-case UML diagram is a fantastic start.

In this diagram, we combine the two subsystems into one whole picture to give a complete overview. The diagram consists of three parts: game subsystem, chat board subsystem and the operation about player information. There are two types of actors: administrator and player.

For the operation about player information, player can register or request updating his or her information such as ID and password. Administrator can do some operation on player information such as updating password, etc.

For the game subsystem, administrator can interactive with the question database including updating questions and so on after login. For player, her or she can start a new game or load the game after login. A few details about more specified use case and extension system behavior can be found in the diagram.

For the chat board subsystem, a verified administrator can do some operations on the comment database including deleting comments, etc. For the player, a verified player can post comments on the chat board. More details can be found in the diagram.



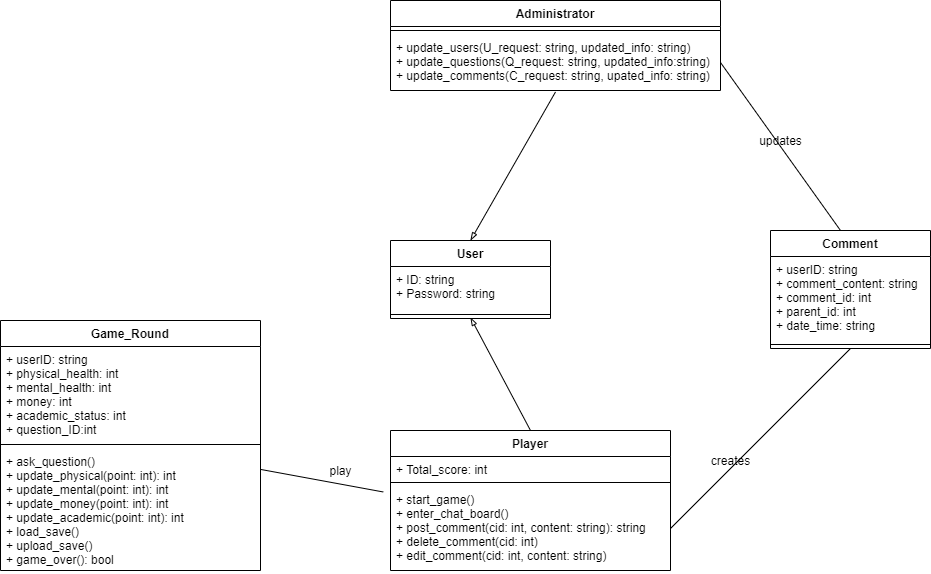
* + 1. Class Diagram

In our design, there are five main classes: User, Administrator, Player, Comment and Game\_round. Specially, class Administrator and Player are inherited from class User. Comment and Game\_round are the main classes for two subsystem respectively.

For User, the basic properties are ID and password. The functional components of administrator user are under the Administrator class. For player, total\_score is counted and stored in class Player. Under Player class, user can use the function start game, enter chat board and their child functions. More details can be found in the diagram.

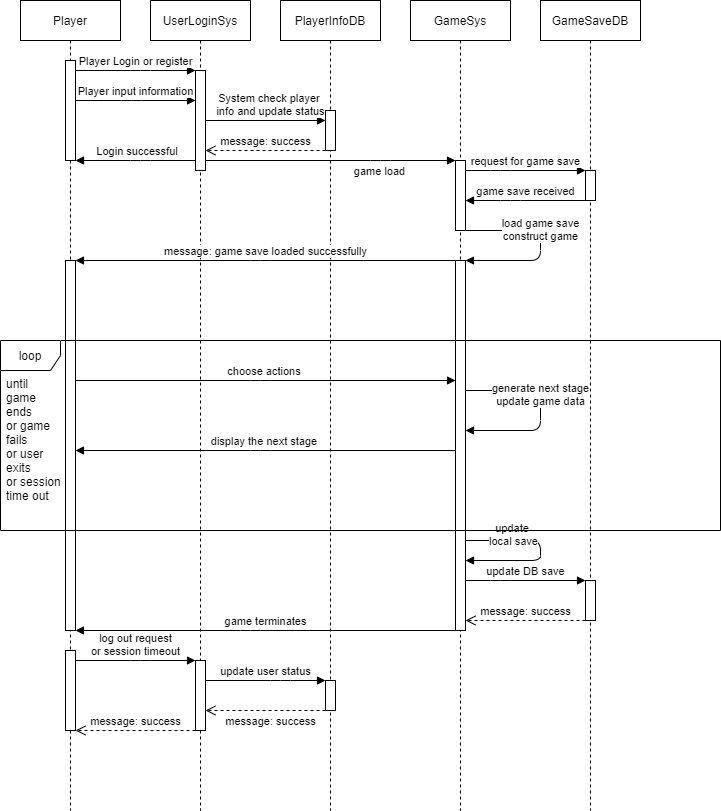
For Game\_round class, when player play the game, userID and the health values should be saved locally in the function. Furthermore, for implementation convenience, the question ID should also be stored. The main process functions are under this class and more details can be found in this diagram.

For Comment class, it represents a comment block which have parent comment and child comment normally. Based on the user-friendliness, User ID and date & time are stored and displayed. Besides, for implementation practice, comment ID and parent ID are also part of the attributes of this class.



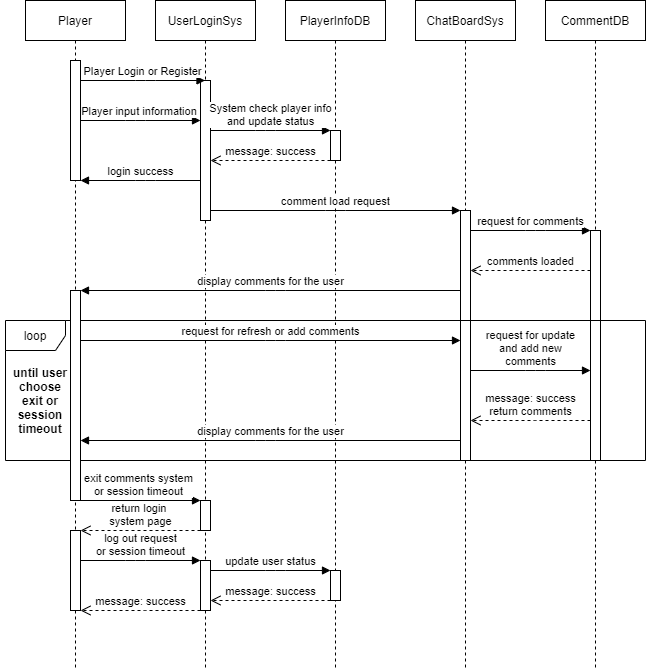
* + 1. Sequence Diagram
       1. Player Gaming Component

First player would login to the system in order to start the game. If the player has no account before, he should register at first and after registration, he will login automatically. The system would handle the login and registration request and interact with the player database, updating the status from “offline” to “online”. If the status is “online” when player trying to login, the system would reject the request because the account is occupied by other devices. After login, the system would direct the player to the game board system, which interacts with save database to fetch the nearest save for that player. Then player would start the game until the game ends or player exits, or session has timed out if the player does not do anything for a while. Once the game ends, the game system should upload all its local saves to the save database of player. Then the player could choose to log out, so that system would interact with the player database to update the status of that player from “online” to “offline”. The whole component ends.



* + - 1. Player Chat Board Component

Using similar logic, we handle player login and logout. The focus of the diagram is on the loop. Player either add new comments or refresh the chat board. Upon that, chat board system interacts with the comments database to insert new comment or fetch the newest comments for displaying.



* + - 1. Administrator Operation Component

With similar logic, we handle admin login and log out. The focus of the diagram is on the operation loop. The admin would choose one of the operations from CRUD (Create, Retrieve, Update and Delete) on some databases such as player database and comments database. Then the admin system would send the corresponding request to the corresponding database to execute the statement.

