
Design Model Report of

Typing tutor

Version <1.0>

Prepared by
Group Name: 4

Nguyễn Trường Giang
Phạm Nguyễn Hoàng
Nguyễn Văn Quang
Lương Ngọc Huyền
Nguyễn Văn Tiến

K59CA
K59CA
K57CA
K59CA
K59CA

nguyentruonggiang71096@gmail.com
14020663@vnu.edu.vn
quanguet@gmail.com
lqnhuyen@gmail.com
14020695@vnu.edu.vn

Instructor: Dr. Dang Duc Hanh

Course: Software Engineering 2016

Date: 04/12/2016

Table of Contents

Contents

List of Figures	4
1. Introduction	5
1.1 Purpose	5
1.2 References	5
2. Design Model.....	5
2.1 Identify the object classes in the system.....	5
2.2 Choose the design model.	7
2.2.1 The use case: Login	7
2.2.2 The use case view profile	8
2.2.3 The use case Learn Basic Lesson	8
2.2.4 The use case Practice Lesson.....	9
2.2.5 The use case Play Bubble Game.....	10
2.2.6 The use case Play Random Game.....	10
Index	13

List of Figures

Figure 1: The object classes.....	6
Figure 2 Two object classes derived from the abstract.....	6
Figure 3 Two classes inherited from Game class.	7
Figure 4 - System Environment.....	Error! Bookmark not defined.
Figure 5 – User Use Cases.....	Error! Bookmark not defined.

1. Introduction

1.1 Purpose

The purpose of this report is to present a detailed description on the Design Model of the Typing Tutor Website System. This document is intended for both the stakeholders and the developers of the system.

1.2 References

The report was completed following the book and documentation recommended and provided by Dr. Dang Duc Hanh including:

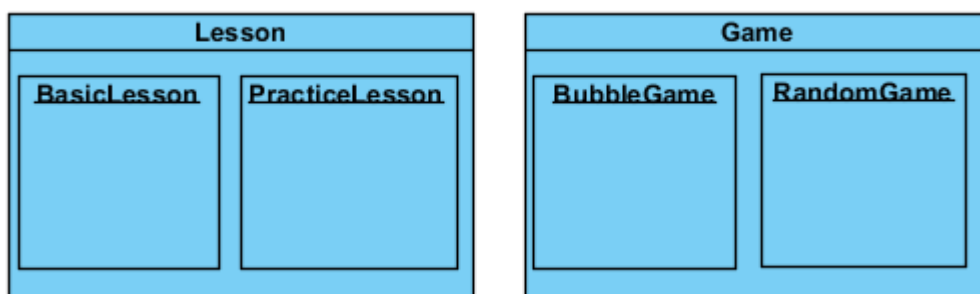
- The textbook Software Engineering 9th Edition written by Sommerville
- The series of lectures and slides of Software Engineering Course 2016 provided by Dr. Dang Duc Hanh.
-

2. Design Model

2.1 Identify the object classes in the system

From the requirements documenting steps, we determine what is the object classes in the typing tutor system. Then, we also identify the attributes and operations or methods for each class from the description of the system requirements documentation.

There are totally 6 object classes of the Typing Tutor system shown in the following figures. To design the inheritance hierarchy of the system, we figure out the common features including attributes and operations, then we generalize them into more abstract classes. More precisely, we abstract 2 objects BubbleGame and RandomGame in the same interface Game, and 2 objects BasicLesson and AdvancedLesson in the same interface Lesson.



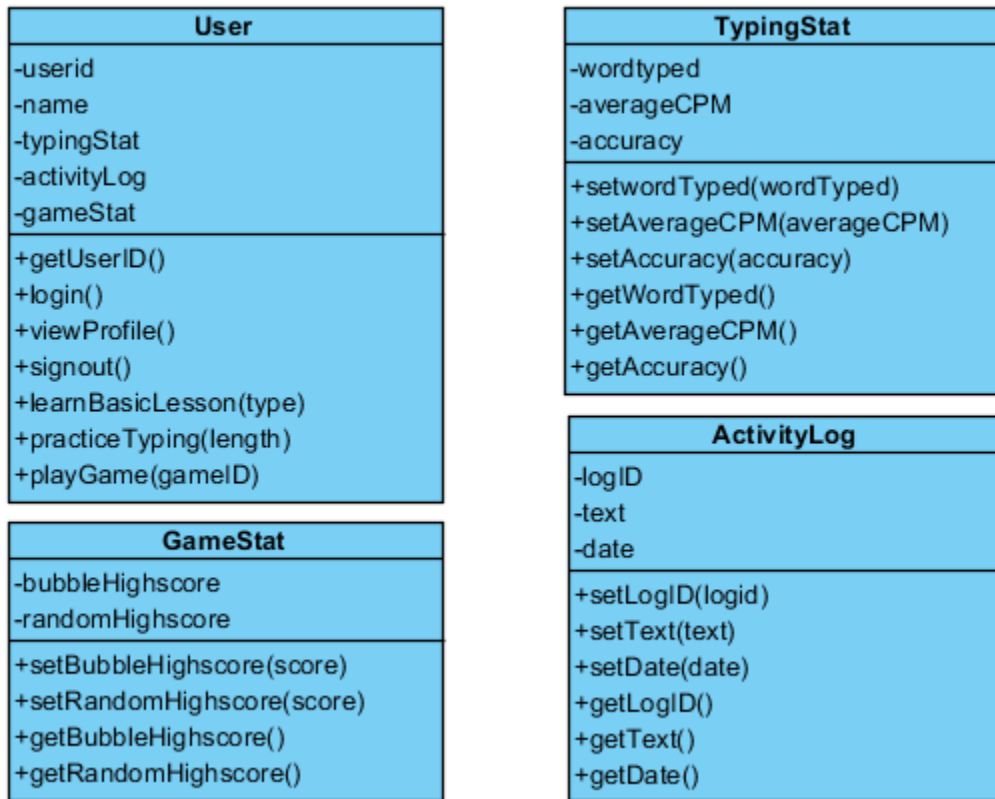


Figure 1: The object classes of the project

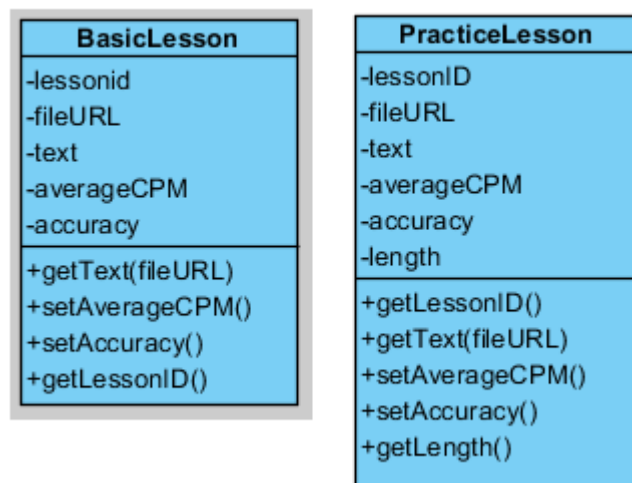


Figure 2 Two object classes derived from the abstract

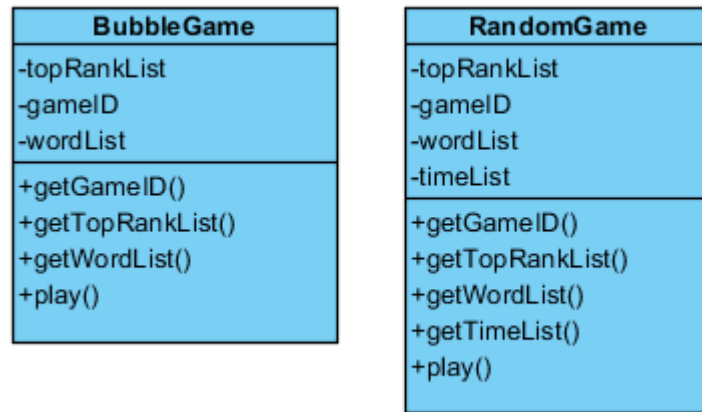


Figure 3 Two classes inherited from Game class.

2.2 Choose the design model.

In this project, we harness the design models in our system for showing the associations and relationships between the object classes in the system. The design model utilized in this project is **sequence model**, which describe the dynamic structure of the system and show the interactions between the system objects. The model is the bridge connecting the system requirements and the implementation of the project together.

We use UML to describe and visualize the interactions between the objects of the system by deploying the sequence model. Below are the significant use cases of the system where the users play the role of main actors.

2.2.1 The use case: Login

Description of Use Case:

The users log in to the website system using their Facebook account.

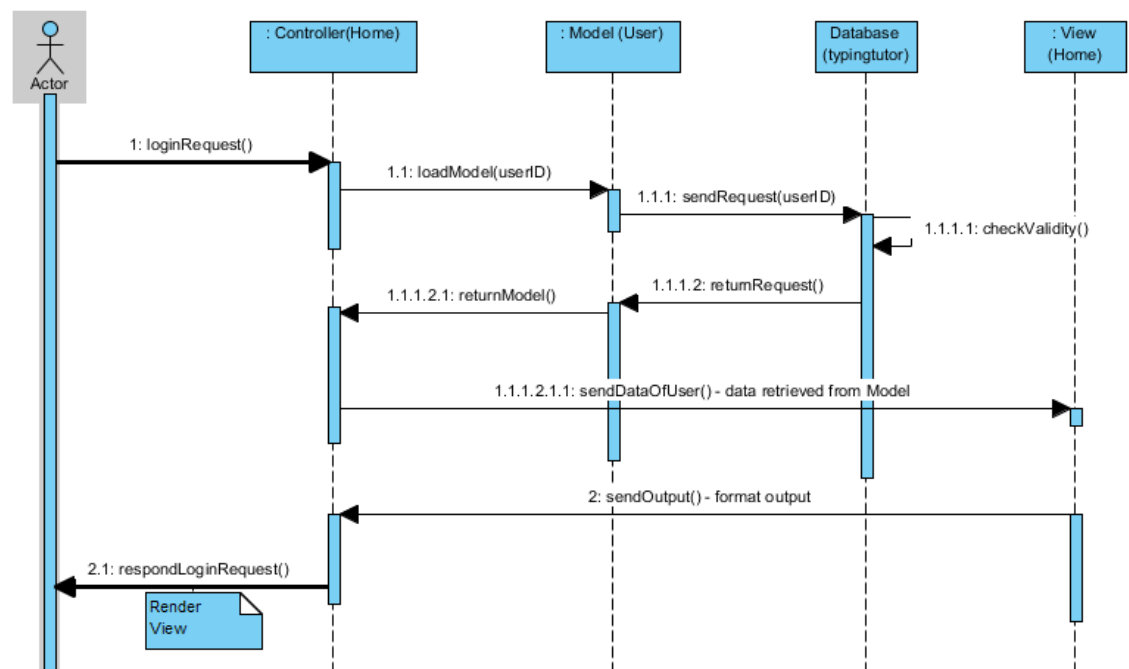


Figure 4 Login Use Case - Sequence Diagram

Firstly, the users send a login request to the controller, it uses the authentication of Facebook API to achieve the user ID and passes it to model to achieve the data from the database. After that, the model returns the data to the controller and then, the controller loads the view with the data retrieved which is the response to the login request.

2.2.2 The use case view profile

Description of Use Case:

The users view their profile in the website system.

Firstly, the users send a view profile request to the controller, it achieves the user ID and passes it to model to achieve the data of users including personal information, activity log, typing statistics, game statistics from the database. After that, the model returns the data to the controller and then, the controller loads the view (Me Profile) with the data retrieved which is the reply to the view profile request.

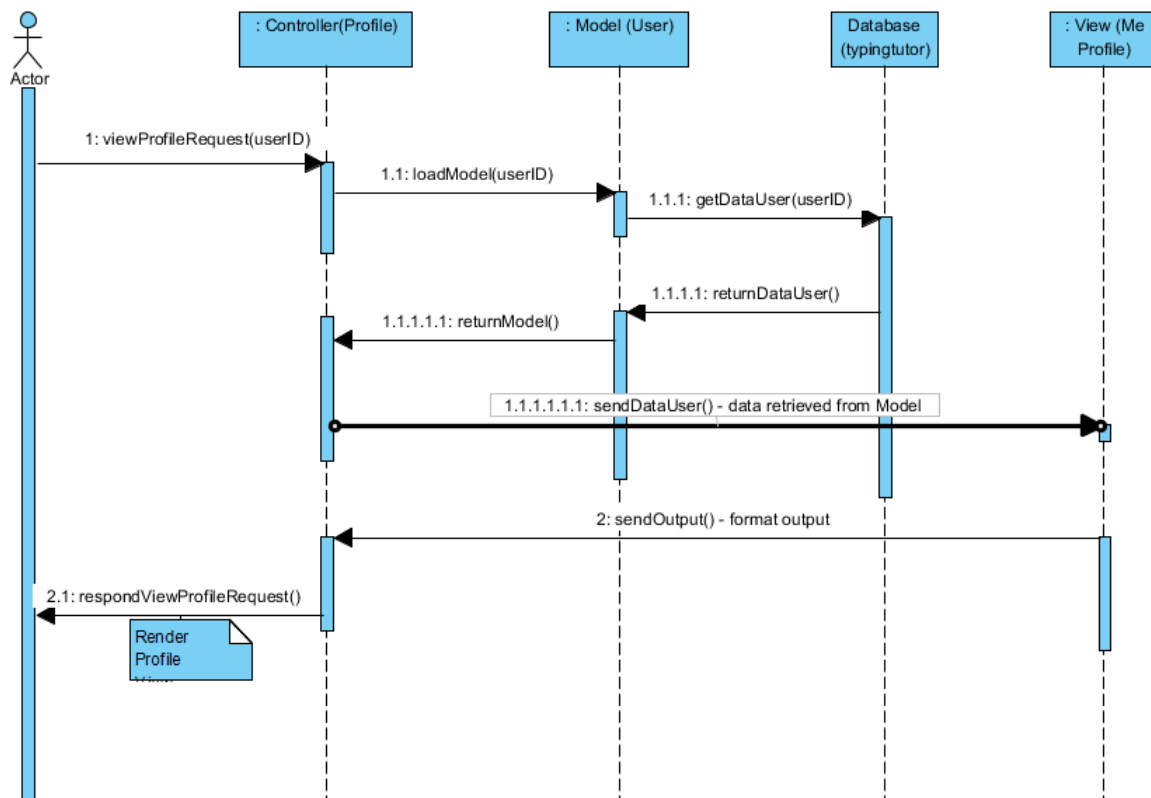


Figure 5 View Profile Use Case - Sequence Diagram

2.2.3 The use case Learn Basic Lesson

Description of Use Case:

The users want to learn the basic lesson of typing in the website system.

Firstly, the users send a learn-basic-lesson to the controller by clicking the specific icon of the basic lesson list in the Lesson Controller, it achieves the basic lesson ID

and passes it to model Lesson to achieve the data of the lesson including src link and text, from the database. After that, the model returns the data to the model Lesson and then, the controller loads the view (Basic Lesson) with the data retrieved which is the reply to the learn-basic-lesson request.

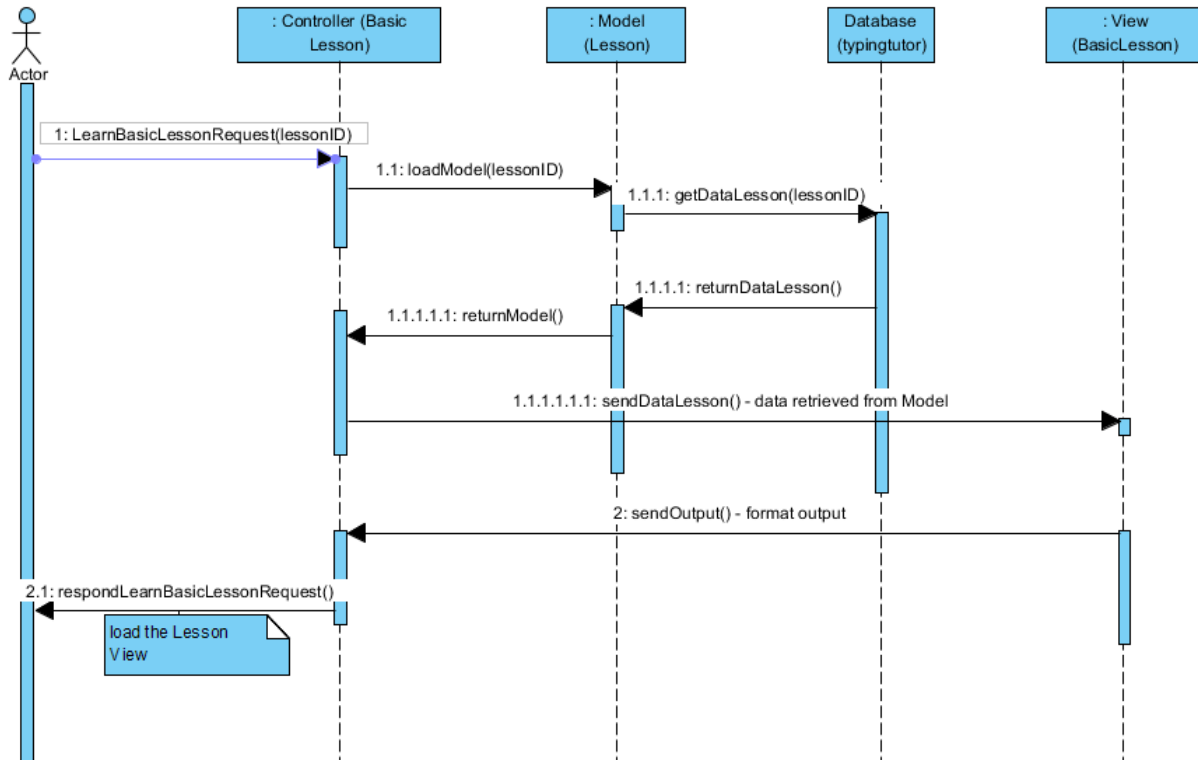


Figure 6 Learn Basic Lesson - Sequence Diagram

2.2.4 The use case Practice Lesson

Description of Use Case:

The users want to practice the lesson of typing in the website system.

Firstly, the users send a practice-lesson to the controller by clicking the specific icon of the advanced lesson list in the Practice Controller, it achieves the advanced lesson ID and passes it to model Practice Lesson to achieve the data of the lesson including src link, length and text from the database. After that, the model returns the data to the model PracticeLesson and then, the controller loads the view (practice Lesson) with the data retrieved which is the reply to the practice-lesson request.

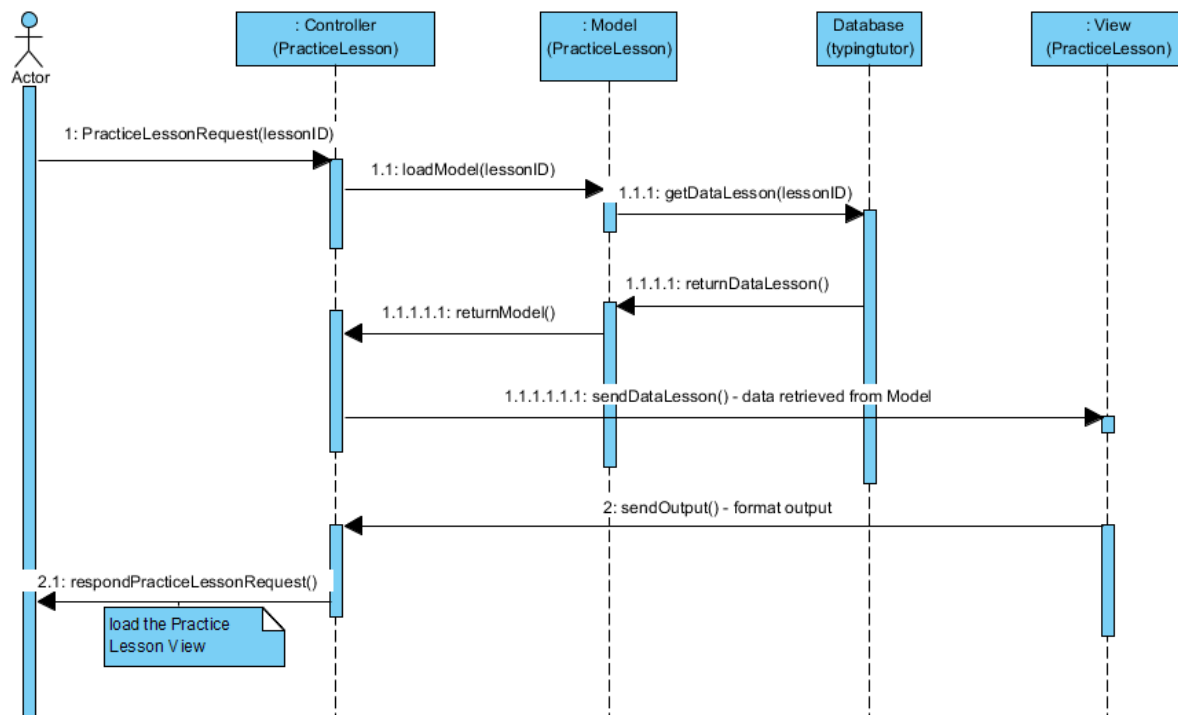


Figure 7 The use case Practice Lesson - Sequence Diagram

2.2.5 The use case Play Bubble Game

Description of Use Case:

The users want to practice the lesson of typing in the website system.

Firstly, the users send a play-bubble to the controller by clicking the Bubbles of game list at the bar in the Home Controller, it obtains the bubble game ID and passes it to model Bubble Game to achieve the data of the game from the database. After that, the model returns the data to the model Bubble Game and then, the controller loads the view bubble Game with the data retrieved which is the reply to the bubble-game request.

The use case's interactions are visualized by sequence diagram in **Figure 8**.

2.2.6 The use case Play Random Game

Description of Use Case:

The users want to practice the lesson of typing in the website system.

Firstly, the users send a play-random to the controller by clicking the Random of game list at the bar in the Home Controller, it obtains the randomgameID and passes it to model Random Game to achieve the data of the game from the database. After that, the model returns the data to the model Random Game and then, the controller loads the view Random Game with the data retrieved which is the reply to the play-random-game request.

The use case's interactions are visualized by sequence diagram in **Figure 9**.

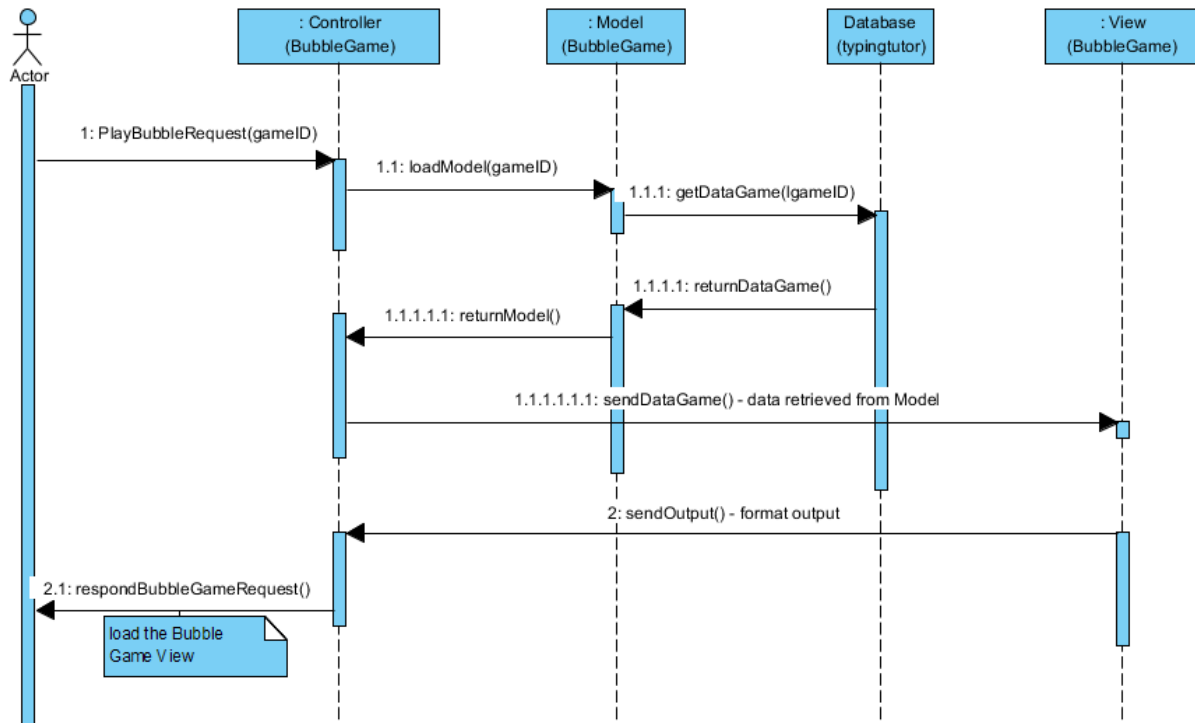


Figure 8 The Play Bubble Game Use Case - Sequence Diagram

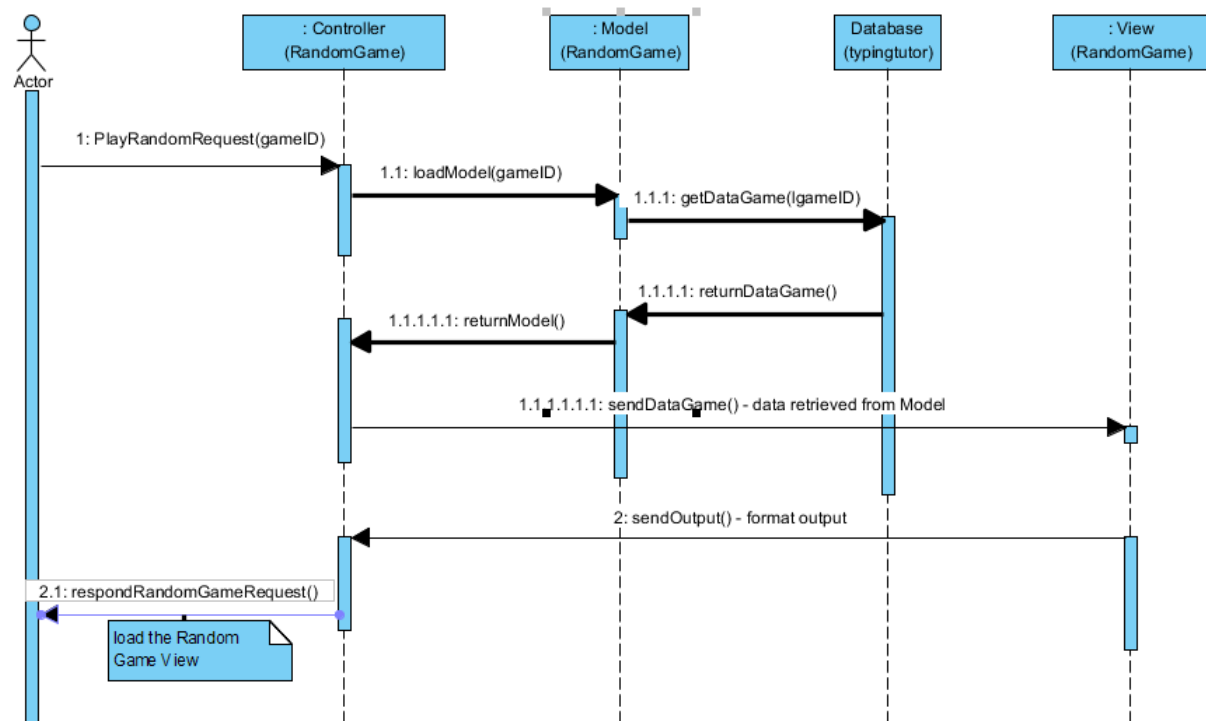


Figure 9 The Play Random Game Use Case - Sequence Diagram

Index

Add 7, 8, 11, 12

Database 1, 9, 10, 11

Editor 1, 3, 7, 8, 9, 10

Lesson 1, 3, 4, 6, 7, 8, 9, 10, 12

Profile 4, 5, 6

Update 4, 6, 7, 8, 10, 12

User 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12