**Implementation and Partial Demo**

**Implementation and Partial Demo**

**GameSAT**

Andris Docaj

Mostofa Adib Shakib

Ayush Dhumal

Scott McFarlane

Shree Shrestha

1. **Project Description**

GameSAT is an android device app which intends on making learning fun. This aim is achieved by transforming learning into an enjoyable play, where the user’s competitive aspects are challenged. The main intention of GameSAT is to prepare high school students for the verbal section of the SAT exam, or undergraduate students for the verbal section of the GRE exam.

Our app will first require the user to create an account so that their progress can be traced. Once an account has been created the user is then presented with the option of playing or training. The playing option allows the user to play a word game, which challenges the user’s vocabulary comprehension of words. In addition to this, the user can also play a passage game, where randomly chosen short passages, obtained from various sources, are presented followed by questions. The questions will be related to the passage and will test the user’s reading comprehension, and their understanding of words in context. The play mode will start the user with 5 points and the maximum number of points that the game allows is 50. If the score reaches zero, the game is over. The game will have many levels, and each level will challenge the user by imposing a time limit on the answer window. This time limit will reduce as higher levels are reached. If the user misses a question they will be penalized, but a correctly found answer will lead them to a reward. In addition, if the user obtains five correct answers in a row, they will be rewarded a bonus point. The users will have the option to skip questions they find too difficult and avoid penalty.

In training mode no scoring is kept, and no time limit exists. The training is available for words individually but also for passages of text. The training mode is strictly designed for free learning, without any pressure. In this mode the questions are randomly chosen, but if a question is missed it will show up more often than others, until the user correctly selects for a particular number of times.

1. **Implementation Overview**

The implementation of the project was quite enjoyable because none of our group members had much experience with android development, and the project provided us with the opportunity to learn new things. For this project we had a GitHub repository where edits could be managed. Two of our team members will handle the question database part, while two others will handle the app design part. The final of the group member acts as a mediator helping both groups with their issues, and with development. The tools we used for the project are **Android Studio**, **XAMPP**, **Java**, and **PHP**.

1. **Project Implementation Update**

By the current pace of development, we are well on our way to completing the project in time. Issues remain, such us importing questions from the database into the app. Setting up a database for the highest scorers and allowing the app to access and display this information. We also need to handle a database of questions the user misses, so that these questions show up more often in the training part of the game and are deleted when the user selects the correct choice for five times in a row.

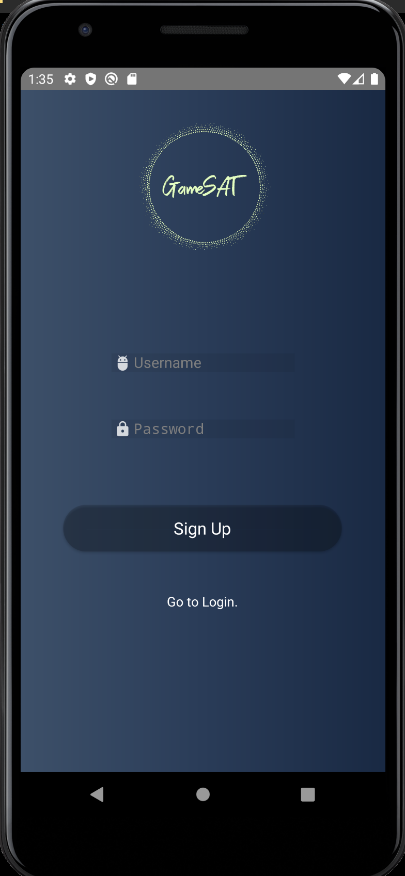
1. **Functionalities Implemented**

The first functionality we implemented was to record and register the user’s data. The initial screen presented to the user is the following,

A picture containing text, monitor, electronics, cellphone

Description automatically generated

Upon clicking the registration button, the user is taken to the following sign-up screen,



For this project we decide to use XAMPP for our database management, using the Apache server and MariaDB. The following picture illustrates what we are currently running,

Graphical user interface, application

Description automatically generated

The initial database, called logindb, looks as follows,

A picture containing text, screenshot, monitor, indoor

Description automatically generated

Presently there are no users, but if we register a user through our sign-up page as follows,

A picture containing text, monitor, electronics, iPod

Description automatically generated

Then our database will register the user, and this can be proved from the following pic,

A picture containing text, screenshot, indoor, monitor

Description automatically generated

Our user’s data is now stored into the login database, and this data can be used to access the full functionality of our app. Upon entering the data in the log-in screen as follows,

A picture containing text, electronics

Description automatically generated

The user will be greeted with the following welcome screen,

A picture containing text, monitor, electronics

Description automatically generated

If the user decides to play the game, then the following play screen will be presented to the user,

A picture containing text, monitor, electronics, iPod

Description automatically generated

If the user decides to play a word game, the following screen will be displayed,

A picture containing text, monitor, phone, cellphone

Description automatically generated

The user can enter the answer by selecting any of the choices provided and pressing confirm. If the user decides to go back, they will be taken to the welcome screen, from there the user can decide to play or train. If instead the user decided to play a passage game, the user would be presented with the following screen,

A picture containing text, phone, cellphone, electronics

Description automatically generated

Due to the limitations imposed by the screen size and the time limit, we have to be careful not to overwhelm the user with too much to read and too little time to do so. If the user decides to train, they will be presented with the following screen,

A picture containing text, monitor, electronics, green

Description automatically generated

If the user decides to train on word meanings only, then this is the screen which they will be presented with,

A screen shot of a cell phone

Description automatically generated with low confidence

There is no time constraint or score penalty for the training aspect of the game. The passage training screen looks as follows,

A screen shot of a phone

Description automatically generated with low confidence

If the user decides to close the game, they can click the back button and then press the exit button on the welcome screen.

1. **Functionalities currently being implemented**

We are currently working on transferring question data from a database into the app. We are deciding on the proper format that the questions need to be in, to better facilitate the interaction between the database and the app. Moreover, the questions need to be chosen randomly, and each game should have different sets of questions presented to the user in order to avoid redundances and keep the game enjoyable.

1. **Functionalities remaining**

We are currently working on adjusting the timer for particular levels of the game, for both the word play and the passage play. In addition to this, we will be tracking the user’s score, and imposing game ending conditions based on those scores, as well as creating a reward system. Moreover, questions that the user misses will have to be stored in a special database of missed questions and used later on by the training aspect of the game. If during training the user correctly determines the answer five times in a row, for questions that were previously missed, then such a question will be removed from the missed questions database. We will add a skip button to the word game and passage game screens so that the user moves to the next question without point penalties.

1. **Implementation issues**

Based on some limitations we have in terms of server availability there are some features of the app that might not be implemented. The global score would have to make the app available for remote use, and store data on a particular server designated solely for that purpose. So, quite likely this feature might not be implemented. Also, depending on how the missed question database interacts with our app during the time span we have available to us, we might or might not be able to erase the questions that the user understands from the missed questions database. If the interaction gives us problems, instead of deleting such questions we might decide a random way of picking questions or tracking questions that have been missed.

1. **Member Contribution**

Andy and Ayush are responsible for the implementation of the app interface. While Scott and Shree are designing the questions database. Adib is helping on both sides, on the implementation part, and the question design part.

1. **Link to the demo**

* https://www.youtube.com/watch?v=Xw6HUp0uSUM