{TypeMania}

Preliminary Project Website:TypeMania.github.io

CSCI 441 – Project Report 1 Completed

Group 2:

Shaka Elmore

Nargis Sultani

Heather Vroman

Alexandra Stevens

Sep 25, 2022

Table of Contents

Individual Contributions

	Shaka Elmore	Nargis Sultani	Heather Vroman	Alexandra Stevens
Discord Management	85%	5%	5%	5%
Github Management	25%	25%	25 %	25%
Member Contribution	25%	25%	25%	25%
Meeting Minutes	70%	0%	30%	0%
Project Report Task Schedule (Part one)	70%	30%	0%	0%
Task Scheduling (Part Two)	0%	100%	0%	0%
Task Scheduling (Part Three)	0%	100%	0%	0%
Problem Statement	0%	0%	40%	60%
Decomposition into Sub-problems	0%	0%	10%	90%
Glossary of Terms	10%	0%	90%	0%
Business Goals	100%	0%	0%	0%
Enumerated Functional Requirements	0%	100%	0%	0%
Enumerated Nonfunctional Requirements	0%	100%	0%	0%
User Interface Requirements	0%	100%	0%	0%
Stakeholders	0%	0%	100%	0%
Actors and Goals	0%	0%	100%	0%
Use Cases	0%	100%	0%	0%
Casual Description	0%	100%	0%	0%

Use Case Diagram	0%	100%	0%	0%
Traceability Matrix	0%	100%	0%	0%
Fully-Dressed Description	0%	100%	0%	0%
System Sequence Diagrams	0%	0%	0%	100%
Preliminary Design	100%	0%	0%	0%
User Effort Estimation	0%	0%	0%	100%
Identifying Subsystems	0%	0%	100%	0%
Architecture Styles	0%	0%	100%	0%
Mapping Subsystems to Hardware	0%	100%	0%	0%
Connectors and Network Protocol	0%	100%	0%	0%
Global Control Flow	0%	0%	0%	100%
Hardware Requirements	100%	0%	0%	0%
Project Size Estimation	100%	0%	0%	0%
Project Management	100%	0%	0%	0%
Editing Proposal	20%	20%	20%	40%
Editing Report One Part One	20%	40%	20%	20%
Deployment Platform Research	0%	100%	0%	0%
Database Research	0%	50%	0%	50%
Google Doc Creation	25%	25%	25%	25%
Framework Research	0%	0%	0%	100%
Main Menu Design	5%	5%	85%	5%

1 Customer Problem Statement

1.1 Problem Statement

As technology grows, remote activities in the workplace and in academics are growing as well. As a result, the demand for computer skills and interactive, online educational tools is increasing for both adult and youth learner populations. Typing is a foundational computer skill that promotes productivity and efficiency amongst students and those in the workforce. Furthermore, typing skills can promote learning amongst students with developmental learning disabilities by circumventing handwriting difficulties (Marom & Weintraub, 2015). However, studies have shown that many students lack proficient typing skills, which can have negative effects on academic outcomes, especially for those with learning disabilities, who rely on electronic devices. Moreover, it was found that individuals with and without learning disabilities both benefit from typing instruction equally (Marom & Weintraub, 2015). Therefore, there is a verifiable need for engaging, interactive, and online tools to promote typing skills within the adult and youth education industry. To meet this need of the education industry, the customer would like our team to develop an interactive, online education tool to build typing skills (QWERTY keyboard) of its users, which include learners of varying ages and backgrounds.

Because the key goal of this tool is to build typing skills of its users, understanding what makes up typing skill is critical to realizing a successful software. Typing on a keyboard is a neurologically complex activity including sensory-motor, linguistic, and cognitive skills. Furthermore, typing skill can be evaluated based on one's ability to type at adequate speeds with few errors and limited required attention and motor effort (Marom & Weintraub, 2015). As a result, the proposed educational tool must be able to enhance the user's typing speed and accuracy at increasing levels of difficulty. Additionally, the tool must evaluate the user's continuous progress and improvement related to his or her typing skill. Therefore, the tool must contain both an engaging, hands-on aspect to practice the motor skills related to typing as well as a user-specific evaluation aspect to track and monitor the user's progress related to typing speed and accuracy.

In developing such a typing educational tool, there is research that supports gamification instructional strategies as a means to creating engaging and interactive content for learners. Gamification encompasses using games for learning and has been shown to be effective amongst learners of varying ages, cultures, and contexts (Kim, Song, Lockee, & Burton, 2017). Moreover, educational games have been found to be compelling, motivating, and can "enhance [learner retention,] peer communication and social skills" (as cited in Cheung & Ng, 2021, p. 2). Lastly, research has shown that success in gaming can help promote self-efficacy in those with low self-esteem (Davies & Hemingway, 2014). As a result, gamification of this typing educational tool provides an excellent framework for the software based on the customer requirements and broad user audience.

With the understanding of the described industry needs and software requirements, the customer proposes that the software engineering team develop an online, web-based rhythm typing game as an educational tool to be marketed to and implemented in a variety of settings, including school-based, work-based, private, and remote-learning sectors. The customer aims to utilize this tool to meet the industry demand for interactive, online educational tools to improve typing skills amongst both adult and youth users in a broad range of learning sectors. Currently, most of the typing applications are aimed only for a specific group of learners. For example, they are specifically designed for either only kids, adults, or some other groups. This proposed application is aimed for learners/users of all ages and backgrounds. The game interface will have a simple and efficient design to meet the needs of its varied audience. Additionally, this gaming tool will be web-based and will be accessible in all settings with browser and internet access.

The customer has emphasized that game design is an important factor of this game's development as the user's perception of the game is critical to its success as an educational tool. Games, whose users perceive it as lacking entertainment qualities, are less likely to provide the desired motivational and engaging qualities (Cheung & Ng, 2021). To ensure the online, web-based rhythm typing game is perceived as entertaining, the customer has a great stake in the tool's gameplay design. Moreover, the gameplay design must address and challenge the user's ability to type keys at increasing levels of difficulty based on typing speed and key inclusion.

It can be assumed that music is popular among various ages and cultures of the world, so a rhythm-based game is a great way to engage all users in learning to meet one of the most in-demand technical skills in this technological era. The overall premise of gameplay can be summarized as having the user press a specified sequence of keys on their keyboard at timings that correspond to the beat of selected music. To prompt the user to press specified keys, graphics will move across the screen at the tempo of the music and reach a hit zone. The graphic entering the hit zone will cue the user to type the specified key at the specified beat of the music. When the song ends, the user will be given feedback on their performance in the form of a score for a particular song and accompanying difficulty level. This score can then be compared to other users' scores as well as their own previous performances to encourage improvement.

In achieving the gameplay and features previously listed, the customer wants to develop a user-specific interface, which requires unique login, as well as a guest interface option, which allows for anonymous play. The user-specific interface can be accessed through a login portal, in which the user can customize graphics and upload music, view their history, track their progress as typing skills improve, and view other useful user personalized statistics, as well as compete for high scores in a global leaderboard. The guest interface will not include user-specific features, like customizability and performance tracking, but will enable anonymous

gameplay and as a result skill improvement. Providing a guest interface provides flexibility in the software for users who cannot or choose not to have a custom experience while still being able to improve upon typing skills.

To ensure the software is appropriate for children as well as adults, the feature that enables registered users to upload their own music can be locked. For example, it can be locked by parents, who may want to restrict types of music their children are listening to, or by teachers, who are wanting to keep music choices school appropriate. Moreover, uploading custom music tracks is strictly optional as there are songs that are already in the game's database that any user can choose from. This optional music customizability allows improved user engagement.

The registered user and guest both have the option to choose gameplay levels, of varying difficulty, they want to take on next. Registered users can also choose to set the level to automatically adjust to previous performance. This helps the user to better their typing skills by always pushing the difficulty level.

While creating the interface and graphics of the game, the customer notes that the design is the first aspect which catches the eye of the user. It also must meet the needs of the broad user audience therefore is another important aspect to the game development. To meet these needs, the customer proposes that the interface be simple and efficient. The color scheme should remain neutral so as to not overwhelm the adult user, but contain highlights of various colors to appeal to the younger age groups. A game menu and navigation bar will be fixed at the top of the page to promote easy, straight-forward game functionality, navigation, and setting control to enhance user experience for all ages and backgrounds. Various style choices the game offers include changing the background, type font, font color, note colors, and note style on the game if the player wishes to utilize those options. The basic options will be easy for users of all ages to see and understand making it seem "dull" to younger users. Lastly, a full-screen gameplay should be an optionable feature for users to increase visibility as well as limit distractions during completion of levels.

1.2 Decomposition into Sub-problems

Projection is being used to decompose the problem architecture of the proposed software into sub-problems, see Figure 1 shown below. The architecture can be broken down into 3 subsystems including the guest UI subsystem, the registered UI subsystem, and the game subsystem. The guest UI subsystem, subsystem 1, handles anonymous, unregistered guests who want to play. The guest UI subsystem has limited feature access and includes gameplay with default songbank and arcade-style leaderboard score submission.

The registered UI subsystem, subsystem 2, handles all registered guest functionality. Said guests have access to all gaming features, including customizable UI settings, the ability to

upload custom music for gameplay, speed/accuracy progress evaluation data, and account-linked leaderboard score submission. The registered UI subsystem requires a secure user login to access the user-unique interface, data, and gameplay. The game subsystem, subsystem 3, handles the game interface and functionality, including key hit/miss calculation methods and score/accuracy calculation methods.

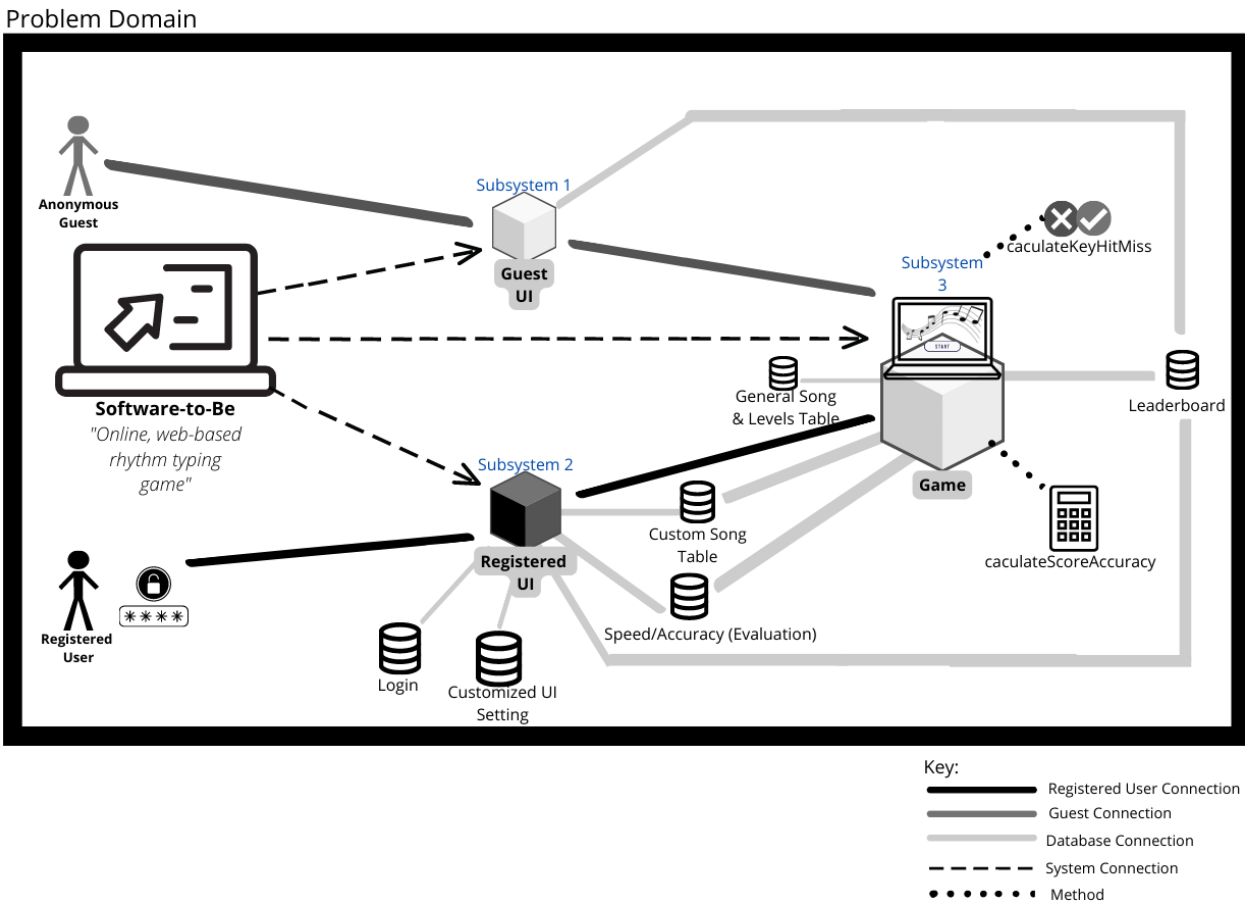


Figure 1: Problem Domain – Decomposition Sub-Problems
https://www.canva.com/design/DAFL3y5Heek/VCB4RDWu8F-Aprp3gME5lw/edit?utm_content=DAFL3y5Heek&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

1.3 Glossary of Terms

TypeMania - Name of the musical typing game aimed towards children and adults alike.

Music - Vocal and/or instrumental sounds created in such a way that appeals to each user.

Game database - Storage for various components essential to the game such as music, information which changes level difficulty, and how each game feature works.

Game feature - Options which can be utilized for a better user experience but do not affect the quality or performance of the game.

Typing speed - How fast each user can type as in words or letters per minute.

Statistics - Stores numerical data such as a user's typing speed, leaderboards, and if they are in or out of the average typing speed for their persona.

Persona - Age groups predetermined by the creator.

Registered user - Frequent users may want to register to receive emails when an update has occurred, and can view statistics from previous gameplay sessions.

Leaderboard - Users can view top scores for each song after gameplay, within other statistics and where they stand compared to other players.

Background - The gradient or image which appears behind gameplay which can be changed by the user.

Note - The graphic labeled with a keyboard character prompting the user on which key to press.

Hit zone - The position on the scroll track that cues the user in on when the note should be pressed.

Scroll track - The ribbon of space stretching horizontally across the gameplay scene. Notes scroll across this track from right to left towards the hit zone.

Note sequence - The timeline on which keys appear during gameplay.

Songmap - The song and its associated note sequence.

Type font - How the text appears on the screen for TypeMania which can be changed by the user.

Type color - The color which the text appears on the screen which can be changed by the user.

Note font - How the notes which prompt the user which key or keys to type appear on the screen which can be changed by the user

Note color - The color the note appears on the screen which can be changed by the user.

Style choices - Various options the user can utilize to change the image and how the screen appears to the player to match their personal preferences.

Performance - How quickly the game loads and how effectively the user types in sync with the music depending on the context used.

Lag times - THe delay between the user typing and the music exiting the speaker of the device the game is being played on, typically consisting of a laptop or desktop computer.

2 Goals, Requirements, and Analysis

2.1 Business Goals

All goals stem from the hope that our software will be used by many. For that to occur, we need to generate interest in & demand for our product. The best way to do this is to have a product worth using in the first place. A worthwhile piece of software is either useful or enjoyable. In our case, the software will be both useful and enjoyable.

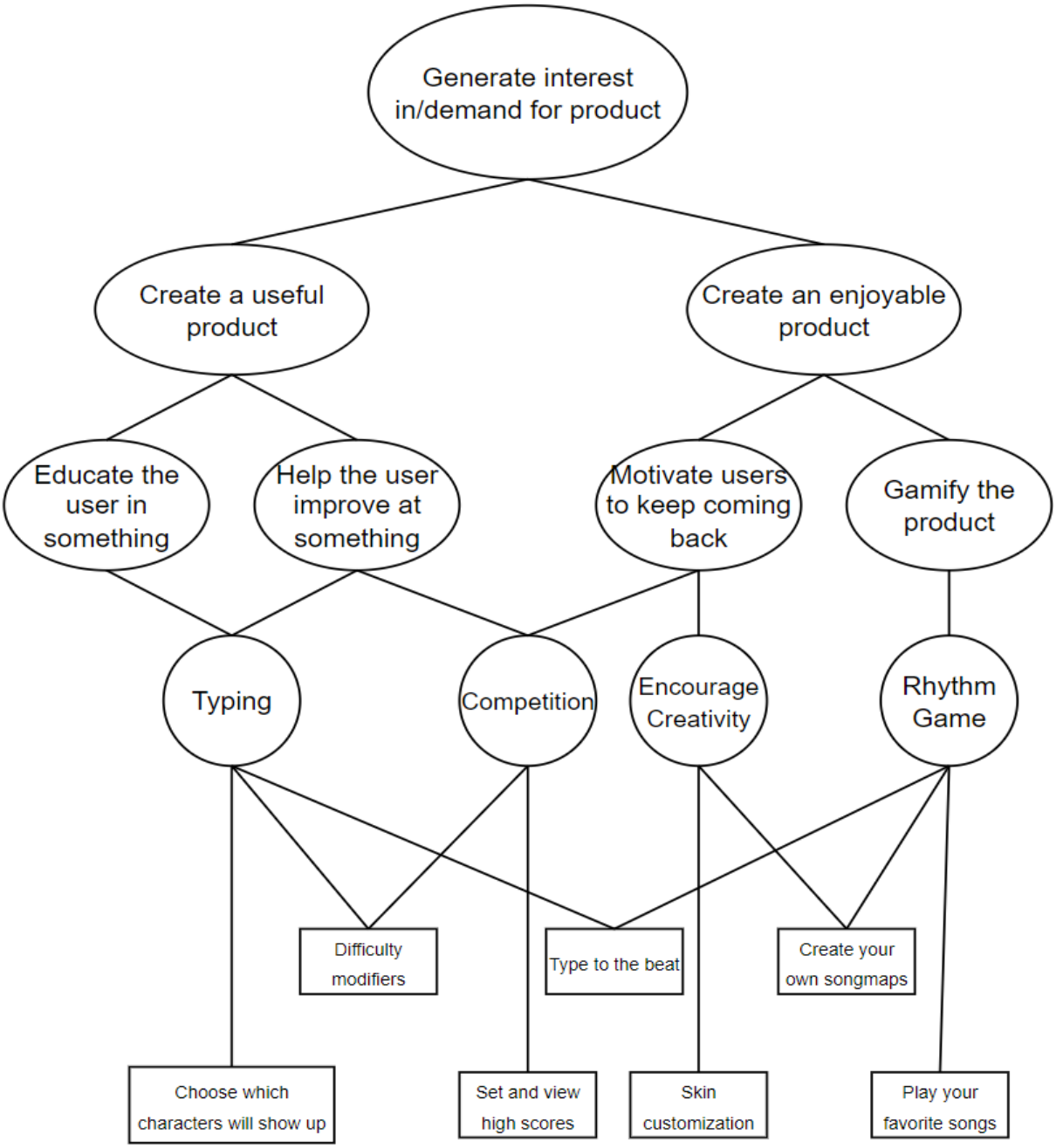


Figure 2.1: Business Goals

2.2 Enumerated Functional Requirements

Identifier	Priority Weight (Low 1 - 5 High)	Requirement Description
REQ-1	5	The users should be able to type to the rhythm of their selected songs.
REQ-2	3	The application should allow a user-specific interface; for unique logged-in and anonymous users.
REQ-3	3	The users should be able to modify the songmap option by specifying which characters can show up.
REQ-4	4	The users should be allowed to select their favorite songs from a pre-populated list.
REQ-5	4	The users should be allowed to upload their own music to allow parents to restrict the types of music their children are listening to.
REQ-6	3	The users should be able to modify the songmap option by specifying how fast the music playback should be.
REQ-7	4	The users should receive feedback from the application on their performance.
REQ-8	3	The users should be able to customize the look and feel of the interface .
REQ-9	4	The users should be able to view leaderboards to view other users' performance.
REQ-10	3	The users should be able to create a map for the songs they upload and should be able to share the map with others.
REQ-11	4	The logged-in users should be able to retrieve their data, such as their customized settings, previous scores, etc.
REQ-12	4	The user should be able to login.

2.3 Enumerated Nonfunctional Requirements

Functionality

Authentication will be supported for the users to retrieve their data. The users will have the ability to retrieve forgotten passwords. Also, the application will be supported on multiple web browsers for more ease for users.

Usability

The interface for this application is designed to be simple and easy to use. The ability to customize the style and the background of the interface makes it appealing to users of various backgrounds and ages.

Reliability

Redundancy and scalability of this application will help keep the failure frequency minimal. Also, the user's data will be backed up on a frequent basis.

Performance

The application is designed to be high performing so that the users could load the music of their choice in a timely manner and start focusing on typing . The application will offer minimum load and lag times. It is designed to serve concurrent requests by multiple users logged in. Also, it is designed to be scalable to keep performing high as the number of users increases.

Supportability

The application is designed to be easy to use and understand because the end users will be of various age groups. Since this application will be used by many, it will be made scalable to include an additional number of servers to achieve load balancing in order to meet the increasing workload and number of users. It will provide the feasibility and flexibility to update and extend any of its components or functionalities to meet the change in user's taste and preferences over time. It will offer improved versions over time that can be installed only by administrators. Also, it will be backed up to a remote server so new features and functionalities could be added and tested in a controlled environment

2.4 User Interface Requirements

Identifier	Priority Weight (Low 1 - 5 High)	Requirement Description
REQ-11	5	The interface should default to the guest page, where the user has the option to login and play as a guest. The guest only has a scoreboard and the option to select songs.
REQ-12	4	After clicking on the login button, the user should be taken to the login page, where they have the option to login, be taken to the sign up page, or retrieve password.
REQ-13	5	The logged in users should have the option to view leaderboards, their own statistics, customize UI settings, create a songmap by uploading songs

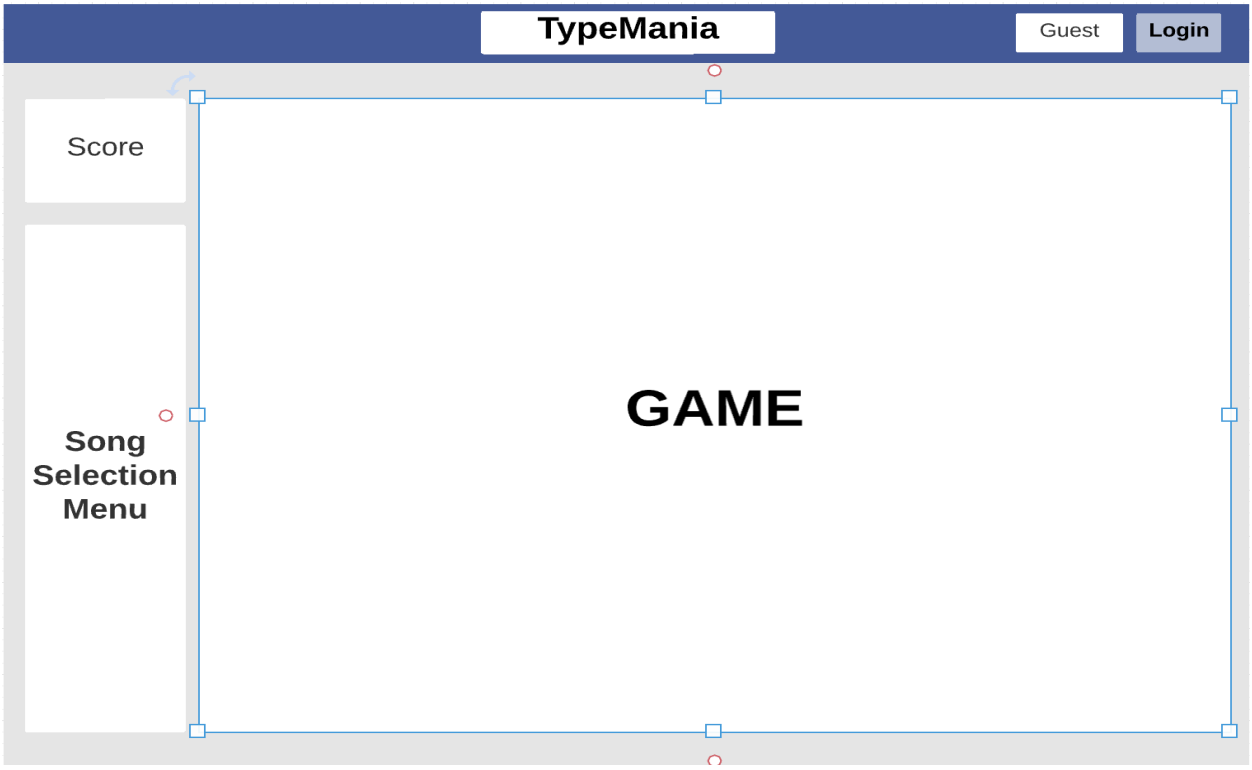


Figure 2.2: User Interface - Default Page

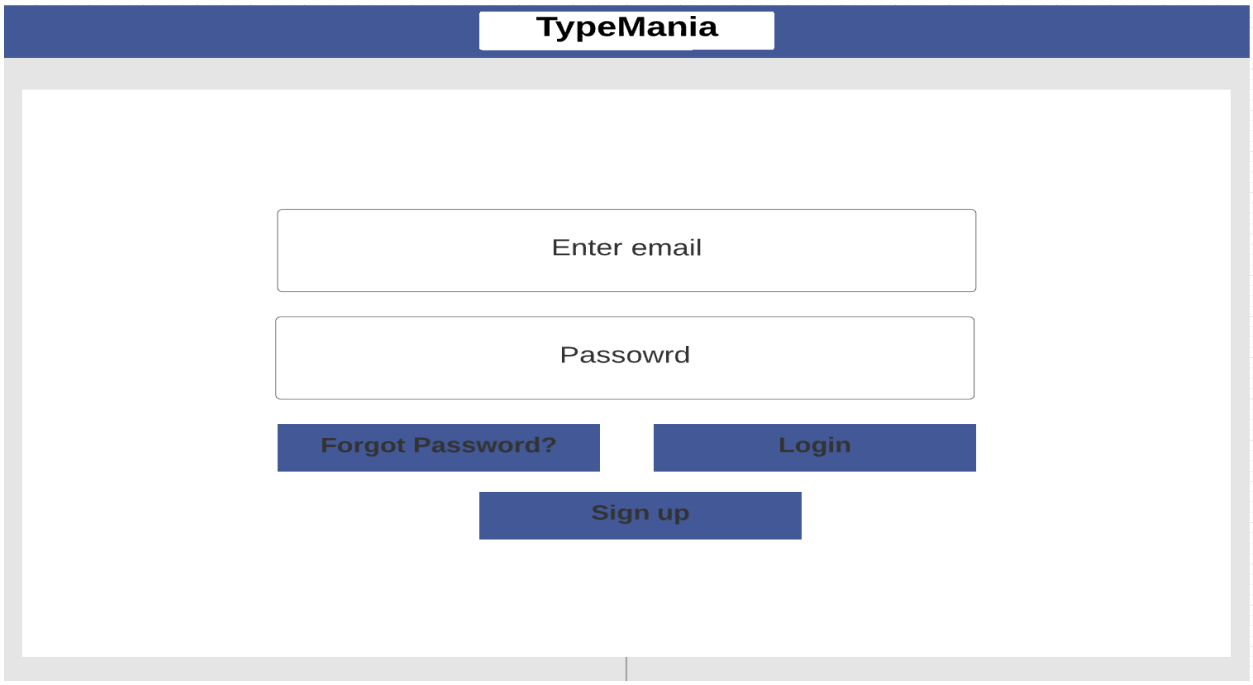


Figure 2.3: User Interface - Login Page

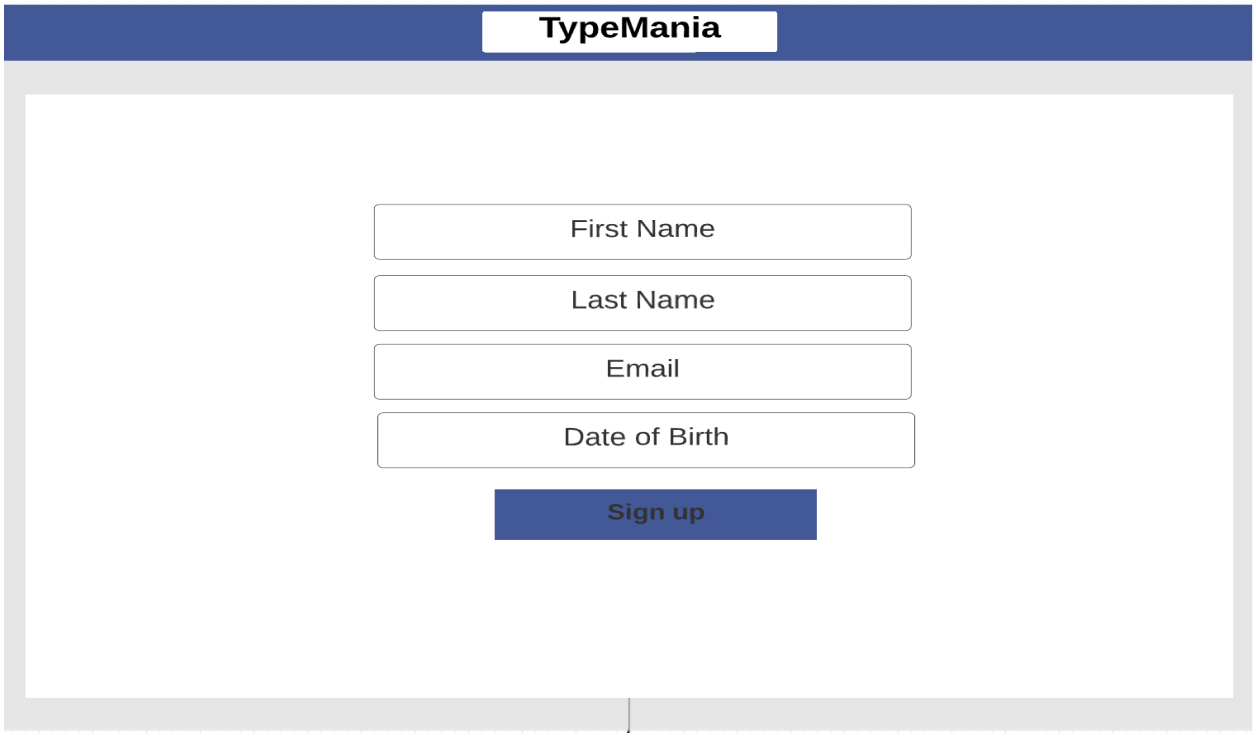


Figure 2.4: User Interface - Signup Page

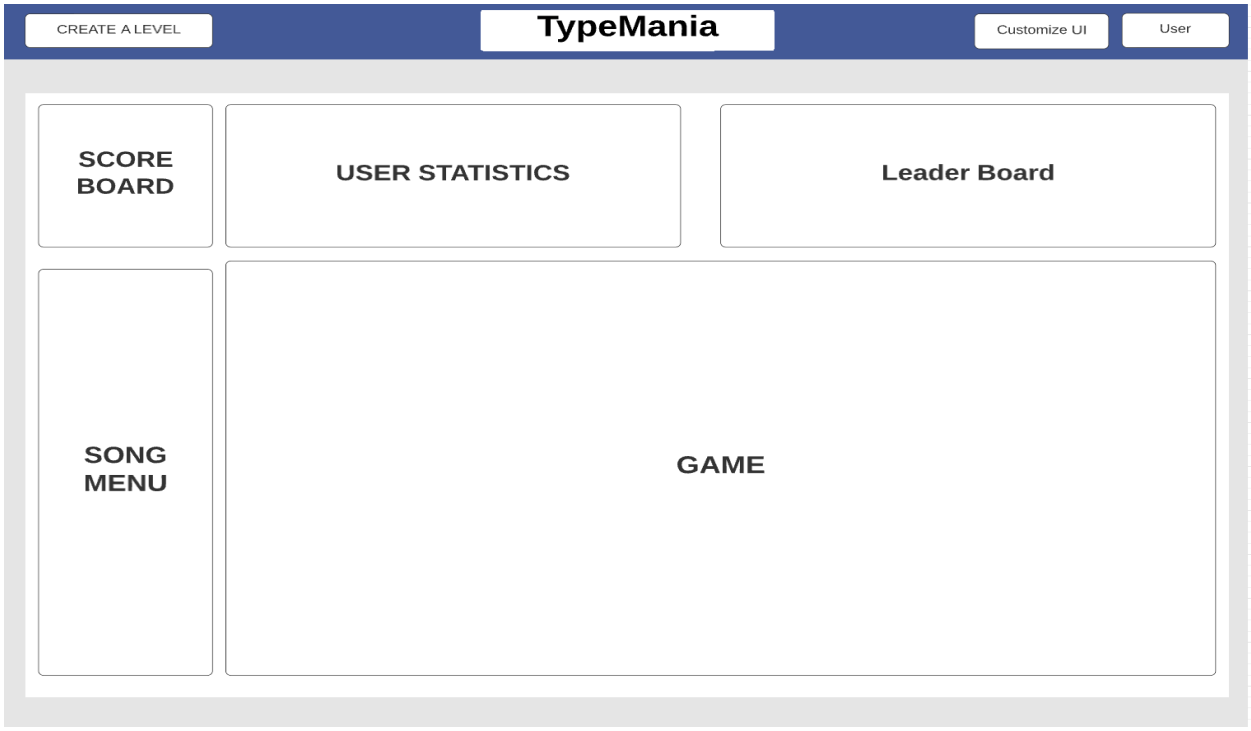


Figure 2.5: User Interface - Logged in User Page

3. Use Cases

3.1 Stakeholders

A. User Appeal

1. Externally, public educators, parents, homeschooling groups, and individuals of all ages are stakeholders due to their interest in using TypeMania.
2. It is important to conduct extensive testing resulting in smooth running software with minimal bugs.
3. To achieve this, identifying who will be using the software and reaching out to potential users is essential.

B. Investors

1. Obtaining investors is crucial to sustain finances such as subscriptions, payroll, business expansions, marketing
2. How many investors depends on the stage the software is at. For example, a software in the beginning stages needs less financial backing in the above

scenario than a large company such as Apple launching a new update or product.
3. Any sort of growth requires financial backing, which is why we need investors.

C. Internal Connections
1. Internal connections are essential in software development not only to have a team of diverse talents working together, but for support, problem solving, and everything else that goes on behind the scenes.
2. Multiple teams come together to make an operation such as TypeMania come to life this includes developers, marketing, and data analysts coming together to work as a team.

D. Suppliers
1. There are not many suppliers needed for TypeMania compared to opening a restaurant, but every operation needs support from established organizations.
2. Suppliers would consist of github for storing code, Figma for storing design prototypes, and Heroku where we deploy the website on.

E. Community
1. Community is important regardless of scenario, making our community an important stakeholder.
2. Receiving consistent feedback from users to take into consideration, and other communication such as success stories keeps the internal teams going to improve TypeMania to its utmost potential.

3.2 Actors and Goals

Role	Type	Goals
Guest User	Initiating Actor	● Register ● Select Song ● Select Typing Speed ● Play Game ● Receive Feedback
Registered User	Initiating Actor	● Login ● Select Song ● Select Typing Speed ● Play Game ● Receive Feedback
Database	Participating Actor	● Update User Table ● Authenticate User Details

		• Grant/Deny Access • Store Songs & Statistics

A. User Appeal
    1. The groups and individuals TypeMania appeals to each interact differently with the system depending on the goal they have set out to accomplish.
    2. Potential users will test TypeMania with the goal to better the software for the company.
    3. This will allow developers to work out any bugs which may appear and receive user feedback before launching the entire software to the public.
    4. The potential users will gain typing skills, a free trial period to use the software after launch along with knowing they were the first to use TypeMania with the opportunity to impact the software by offering quality feedback.
B. Investors
    1. Investors in the company usually do so within stock regardless if that is public or private.
    2. In the long run investors expect a profit from taking the risk and backing the company from the beginning stages.
    3. TypeMania needs to have a stock to offer for this to be an option.
C. Internal Connections
    1. Developers need to ensure the program runs smoothly without bugs, error code, and lag time.
    2. The marketing team finds outside stakeholders so the operation continues to run smoothly, and data analysts study the trends on the website such as which age group visits regularly and how often individuals stay on the page.
    3. In exchange for their services compensation is received for every team member along with access to TypeMania.
D. Suppliers
    1. Suppliers support us by hosting our website, and providing storage for essentials such as code and design prototypes.
    2. In exchange, we pay for subscriptions when payment is due.
    3. Github provides the option to look at other individual and group projects allowing us inspiration and vital learning opportunities for future updates.
E. Community
    1. Community may seem like a small part to a larger network of individuals but its purpose is essential.

2. Users have the opportunity to provide feedback making a difference in the TypeMania which provides them a fun way to learn to type while providing us the opportunity to continually improve the users experience
3. Community is not only about bettering the software, but each other. The opportunity to to hear success stories will encourage the teams behind the scenes and other users benefiting everyone involved.

3.3 Use Cases

The users will have access to the game as soon as they enter the site. By default, the guest page is landed where the user can play anonymously with limited functionalities. On the default page, the users will also have an option to login and play as a logged in user where they can access more functionalities of the game, such as customizing UI settings, uploading their own music and creating a songmap, and much more. The ability to customize features will attract users of various backgrounds and age groups. The core functionalities of the software will be elaborated below.

3.3.1 Casual Description

UC-1 Guest User Page - The default page allows all guest users to play with limited functionalities and gives the option to register/login with access to more functionalities.
Derivations: REQ-2

UC-2 Register - On the Register page, the user has the option to login or register so they get the benefits of accessing more functionalities.
Derivations: REQ-2

UC-3 Registered User Page - The registered users have the option to customize UI settings, view leaderboard, detailed scoreboard, upload music, create a songmap, etc.
Derivations: REQ-2

UC-4 Typing Speed Selection - All users have the ability to choose a typing speed in words or letters per minute.
Derivations: REQ-5, REQ-6

UC-5 View Score - All users are allowed to view their scores.
Derivations: REQ-7

UC-6 Music Selection - Gives all users the ability to select their favorite music to type to the rhythm of.
Derivations: REQ-4

UC-7 Play Game - Allows the users to type to the rhythm of their favorite music and typing speed that they have selected.
Derivations: REQ-1

UC-8 Creating a Songmap - Allows the registered users to create a songmap to the music that they have uploaded.
Derivations: REQ-5

UC-9 Uploading Music - Allows the registered users to upload their own music.
Derivations: REQ-5

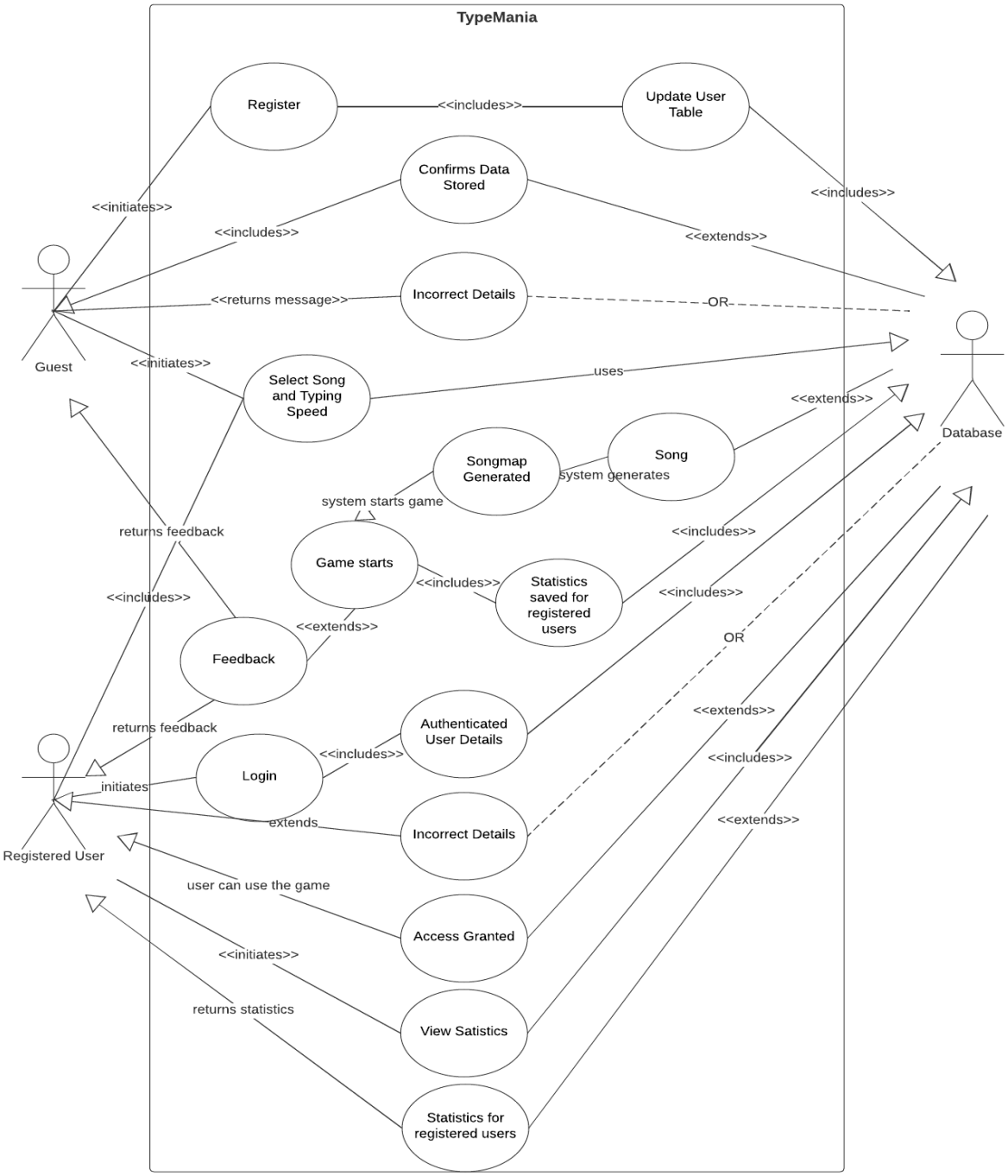
UC-10 Style Choices Selection - Allows the registered users to change the look and feel of the interface to make it more appealing to them. They are allowed to select and change the background, type font, type color, note font, and note color.
Derivations: REQ-3, REQ-8

UC-11 Leaderboard Display - Allows the users to view top scores for each song after gameplay, within other statistics and where they stand compared to other players.
Derivations: REQ-9

UC-12 View Statistics - Allows the registered users to view their statistics.
Derivations: REQ-11

UC-13 Login - Allows the registered user to login to be able to access the system as a registered user and access more functionalities.
Derivations: REQ-12

3.3.2 Use Case Diagram



3.3.3 Traceability Matrix

Requirements	Priority Weight	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8	UC-9	UC-10	UC-11	UC-12	UC-13
REQ-1	5							X						
REQ-2	3	X	X	X										
REQ-3	3										X			
REQ-4	5						X							
REQ-5	4				X				X	X				
REQ-6	3				X									
REQ-7	3					X								
REQ-8	3										X			
REQ-9	4											X		
REQ-10	3													
REQ-11	4												X	
REQ-12	4													X
Total Priority		3	3	3	7	3	5	5	4	4	6	4	4	4

3.3.4 Fully-dressed Description

Use Case: Register

Related Requirements: REQ-2
Initiating Actor: Guest User
Actor's Goal: To be able to register
Participating Actor: Guest user, Database
Preconditions: Not be a registered user
Postconditions: Get registered in the system

Flow of Events for Main Success Scenario:

1The user selects the option to register.
The user gets redirected to the register page.
The user enters info the registration form and submits
The system checks for any errors.
The system sends the data to the user table in the database.

Flow of Events for Alternate Scenarios:

1. If there is any error, the database returns a message and asks the user to re-enter info.
2. The user re-enters info, and the user is saved into the database and the database returns confirmation.

Use Case: Login

Related Requirements: REQ-2
Initiating Actor: Guest User
Actor’s Goal: To login and play as a registered user
Participating Actor: Registered user, Database
Preconditions: Be a registered user
Postconditions: Login and access the system as a registered user

Flow of Events for Main Success Scenario:

1. The user selects the option to login.
2. The user gets redirected to the login page.
3. The user enters credentials and submits.
4. The system authenticates the user against the database.
5. The access is granted and the user is able to use the system as a registered user.

Flow of Events for Alternate Scenarios:

1. The system returns an error message.
2. The user re-enters credentials.

Use Case: Play Game

Related Requirements: REQ-1, REQ-3, REQ-4, REQ-7, REQ-8
Initiating Actor: Guest user, Registered User
Actor's Goal: To be able to play the game to their selected song and typing speed.
Participating Actor: Guest user, Registered user, Database
Preconditions: None
Postconditions: Playing the game and viewing scores or feedback.

Flow of Events for Main Success Scenario:

1. The user selects music and typing speed and the play button.
2. The system loads the song from the database and generates the songmap.
3. The user can start playing.
4. The system returns feedback.
5. The registered user's statistics are saved in the database.

Use Case: View Statistics

Related Requirements: REQ-2
Initiating Actor: Registered User, database
Actor's Goal: To be able to view statistics.
Participating Actor: Registered user
Preconditions: Must be a registered user and logged in.
Postconditions: User can view statistics

Flow of Events for Main Success Scenario:

1. The user initiates selects to view statistics.
2. The user is redirected to the statistics page.
3. The user's statistics are loaded from the database into the page.
4. The user can view statistics.

Flow of Events for Alternate Scenarios:

1. If the system runs into any issue, an error message is returned.

3.4 System Sequence Diagram

https://www.canva.com/design/DAFMaPutaO8/Iq9tL-EeQ-NJic_gmN-WQQ/view?utm_content=DAFMaPutaO8&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

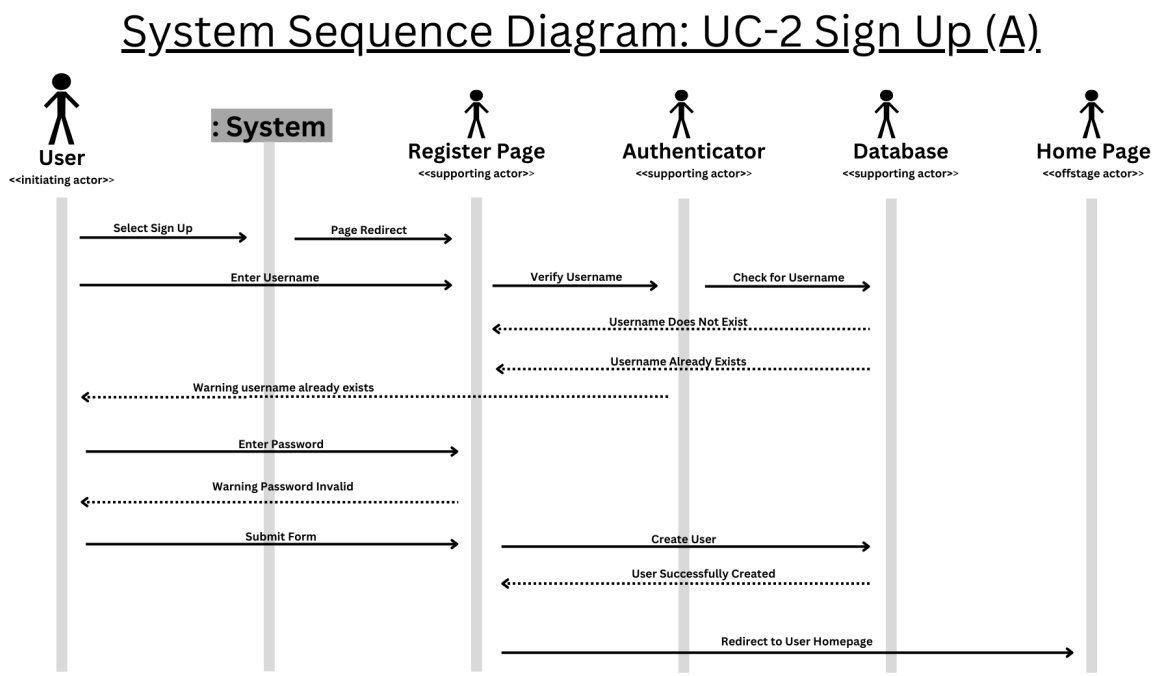


Figure 2: System Sequence Diagram – UC-2 Sign Up (A): Register a new user, main success scenario

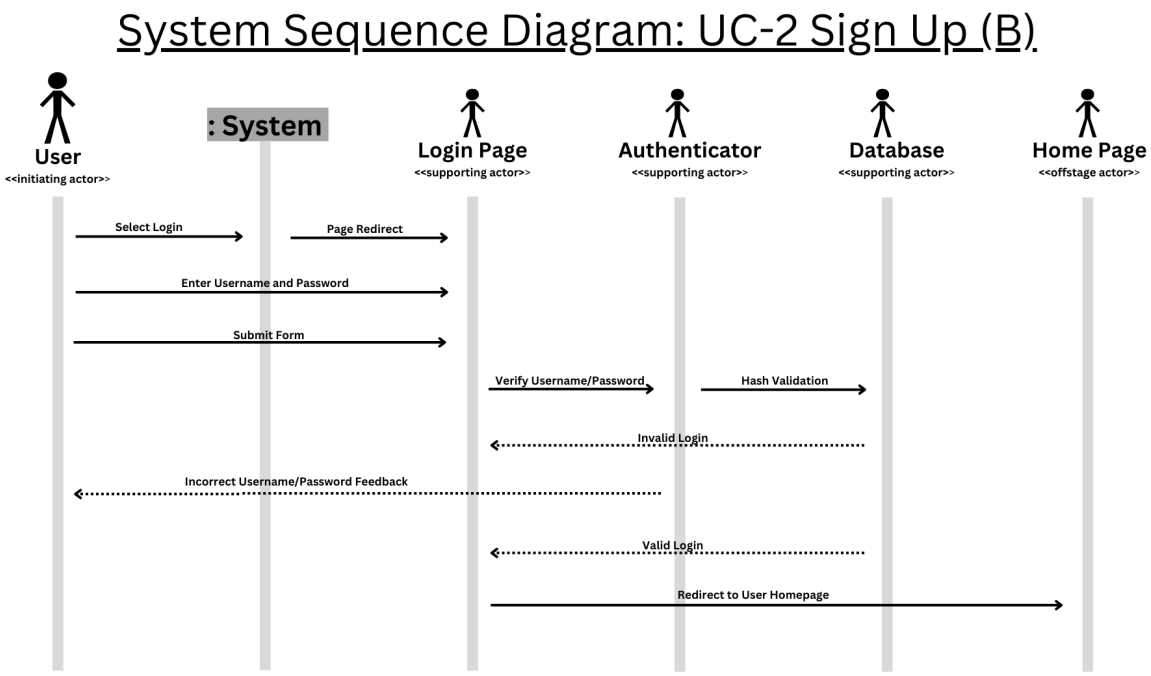


Figure 3: System Sequence Diagram – UC-2 Sign Up (B): Login Existing User, main success scenario

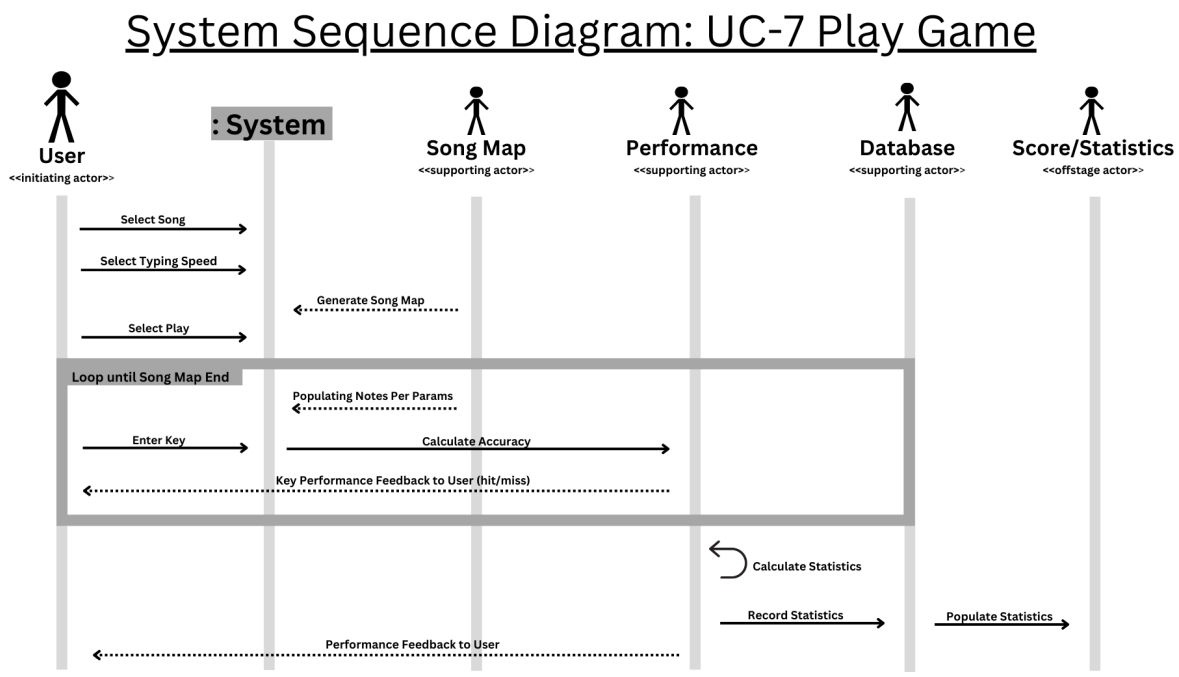


Figure 4: System Sequence Diagram – UC-7 Play Game, main success scenario

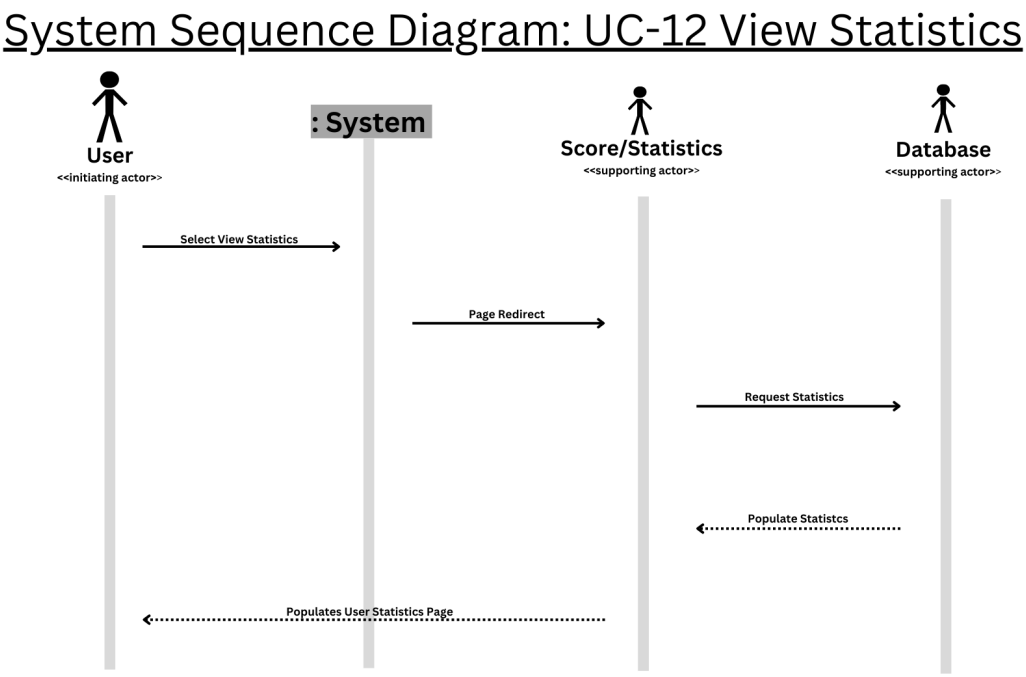


Figure 5: System Sequence Diagram – UC-12 View Statistics, main success scenario

4. User Interface Specification

4.1 Preliminary Design

UC-7 Play Game - Allows the user (or guest) to type to the rhythm of their favorite music and typing speed that they have selected.

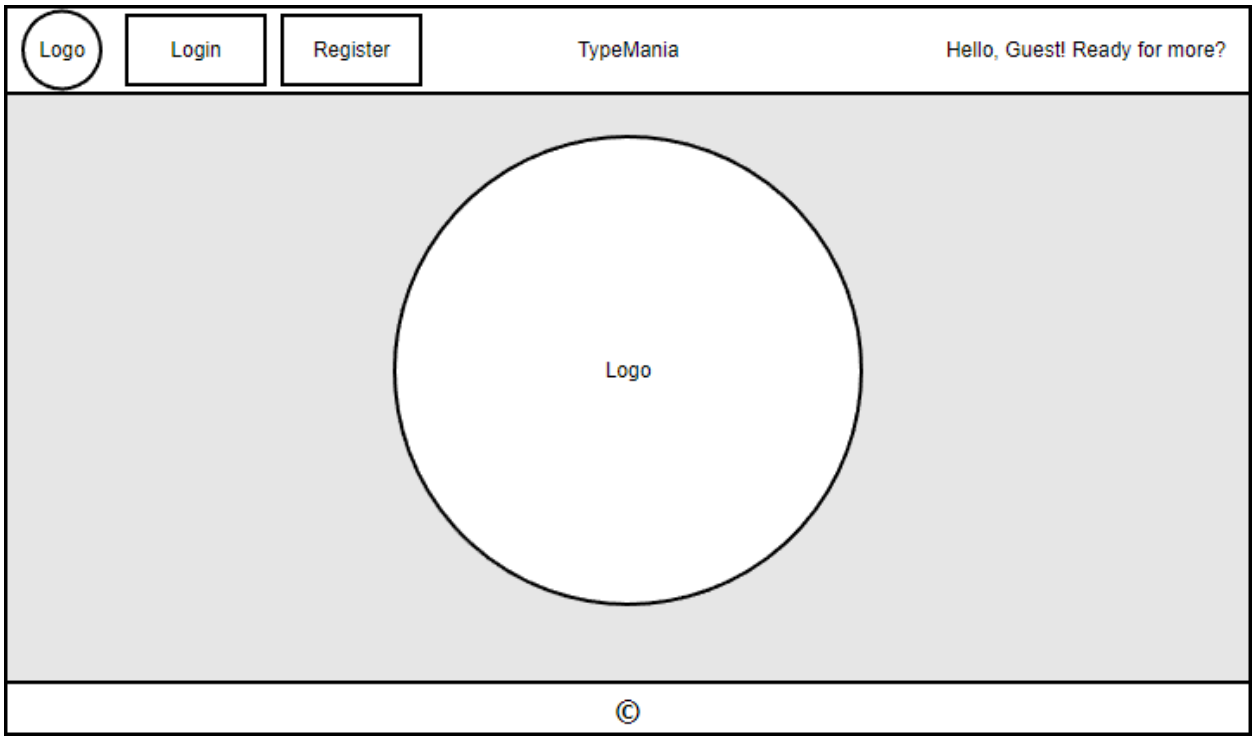


Figure 4.1.1: Landing page for guests.

1. Starting from the landing page as a guest, the user clicks the center logo to reveal main menu options.

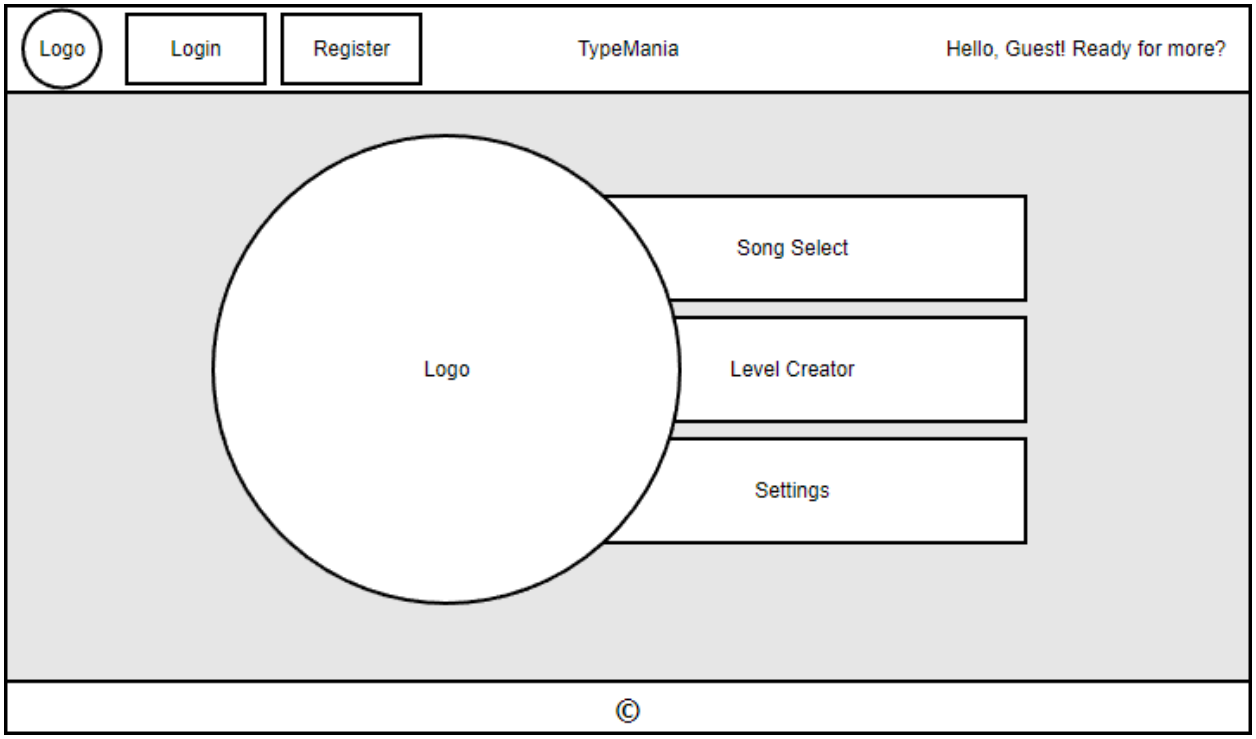


Figure 4.1.2: Expanded landing page after central logo click while not logged in.

2. From the main menu options the user clicks “Song Select”

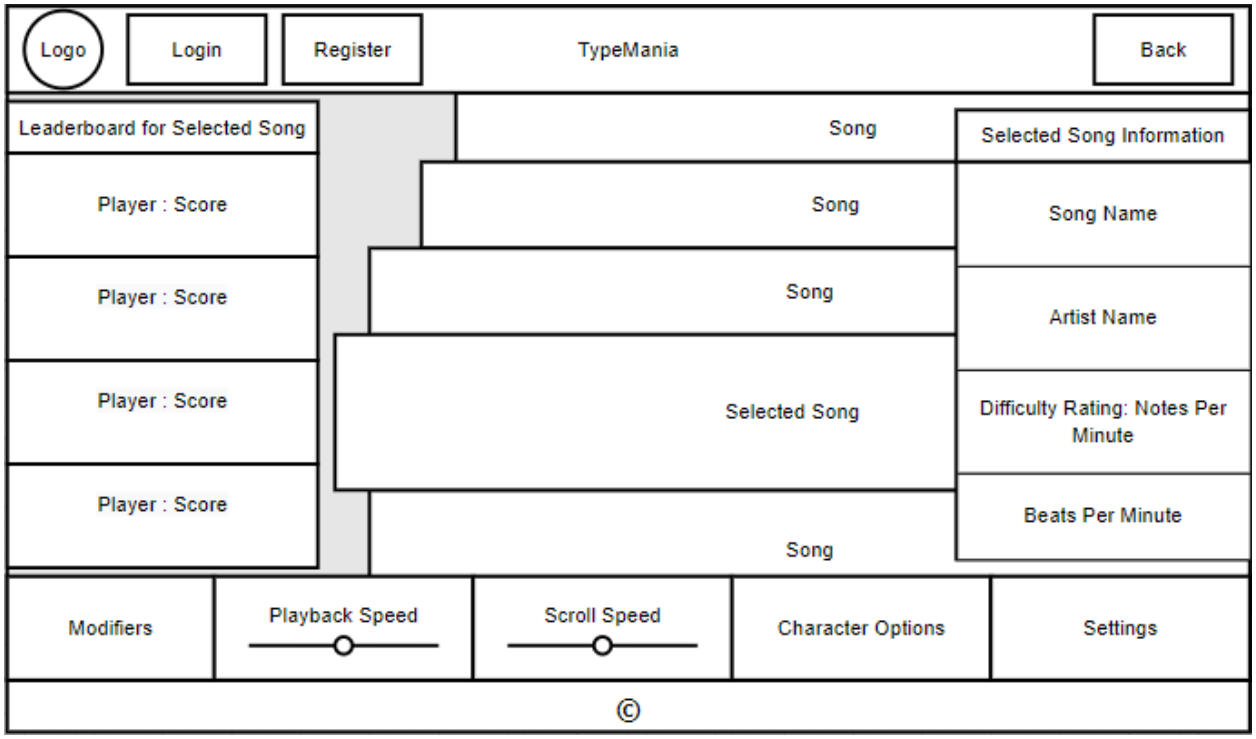


Figure 4.1.3: Song Select scene after clicking "Song Select" from the expanded landing page as a guest. One song will already be selected as "Selected Song" without user interaction.

3. From the Song Select scene the user will click the "Selected Song" box to enter the Gameplay scene.

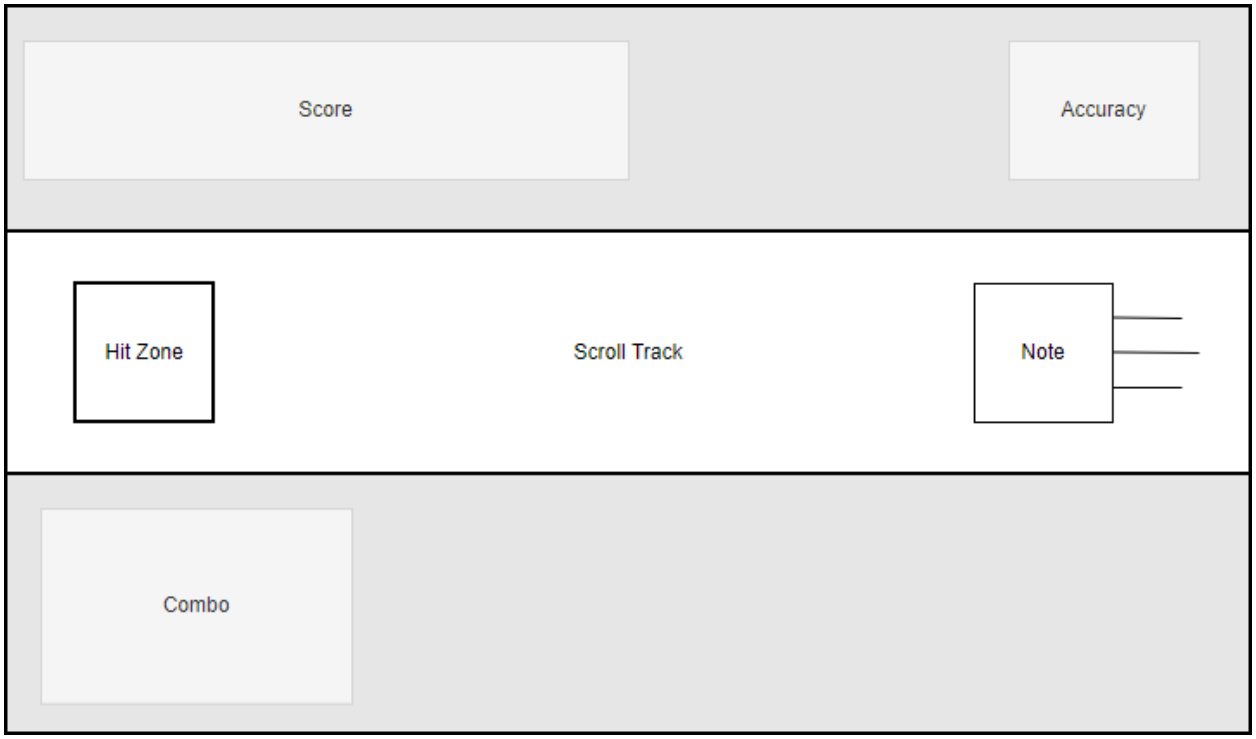


Figure 4.1.4: Gameplay scene after clicking the “Selected Song” box.

4. The user will now utilize the keyboard to tap the corresponding characters found on each “Note” as it reaches the “Hit Zone”. The user is considered to be playing the game at this point so we conclude our use case here.

UC-2 Register - On the Register page, the user has the option to login or register so they get the benefits of accessing more functionalities.

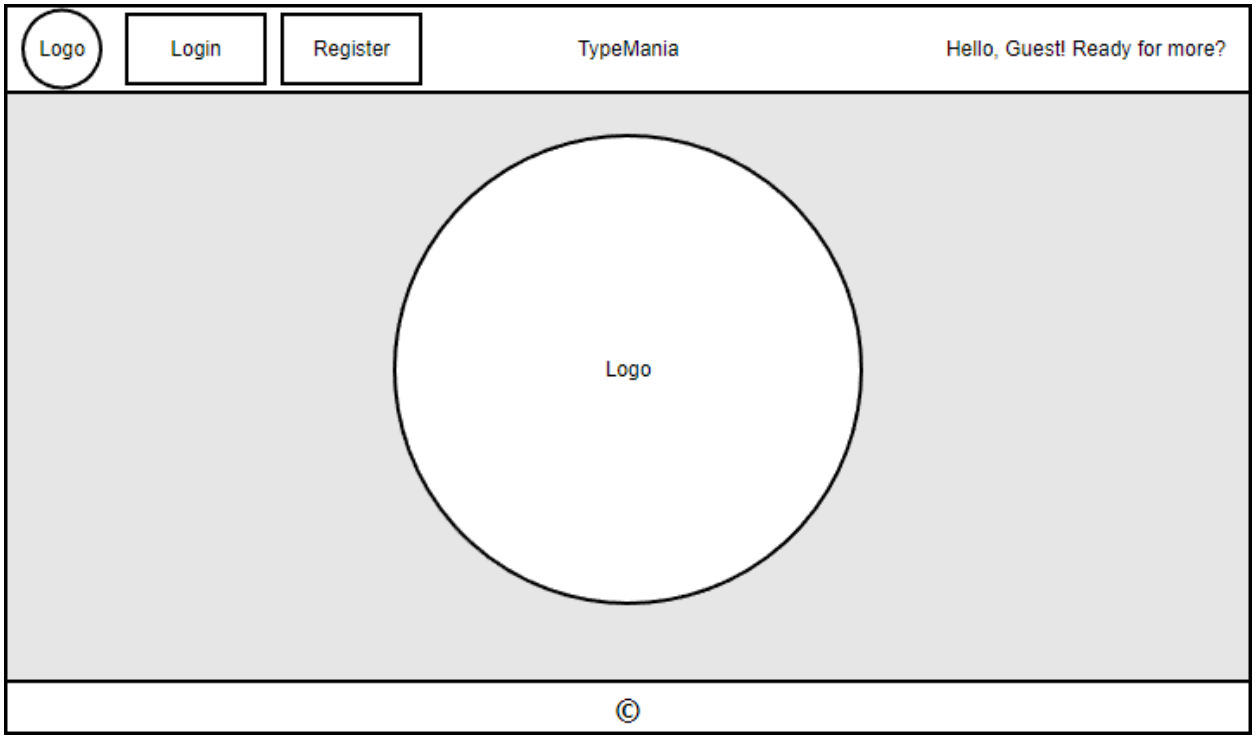


Figure 4.1.1: Landing page for guests.

1. Starting from the landing page as a guest, the user will click “Register”.

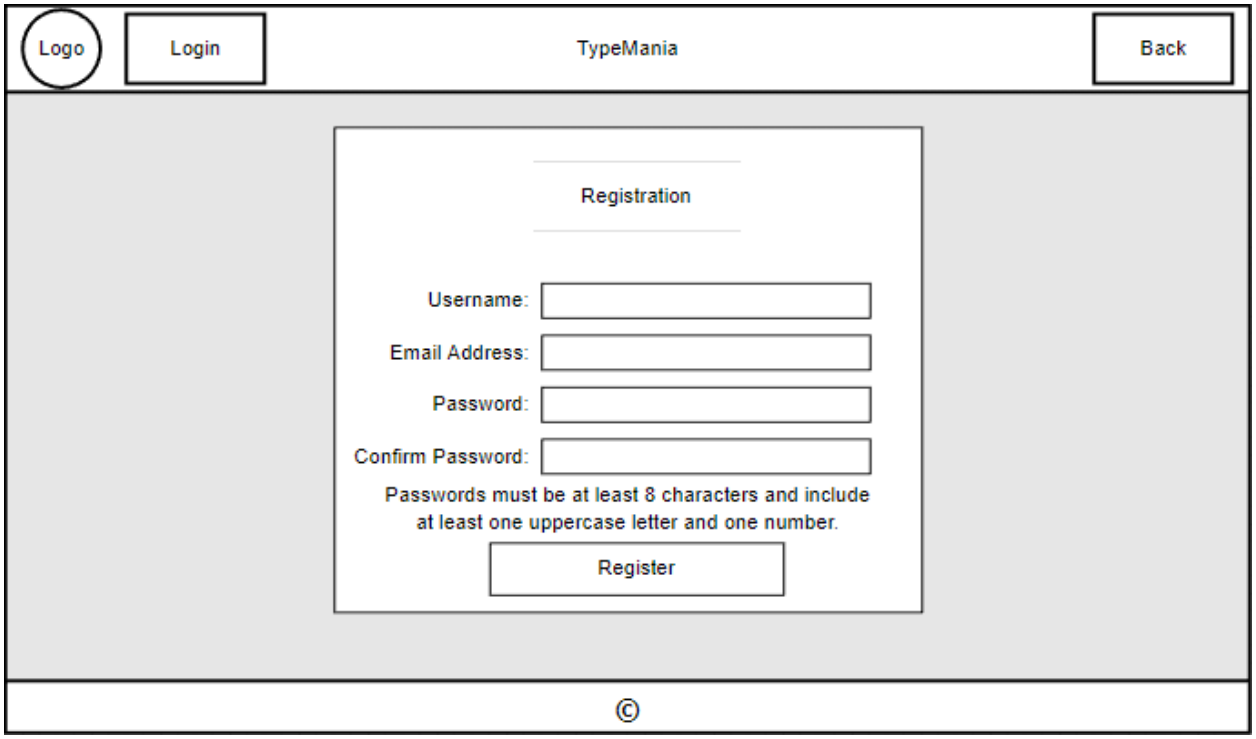


Figure 4.1.5: Registration page after clicking "Register".

2. The user clicks the field to the right of "Username:" and enters their preferred name via the keyboard.
3. The user clicks or uses tab on the keyboard to select the field to the right of "Email Address:" and enters their email address via the keyboard.
4. The user clicks or uses tab on the keyboard to select the field to the right of "Password:" and enters their preferred password via the keyboard.
5. The user clicks or uses tab on the keyboard to select the field to the right of "Confirm Password:" and enters the same password they did above via the keyboard.
6. If all goes well, the user now clicks "Register" or uses enter on the keyboard to complete registration and will arrive at the "Registration Succeeded" page.

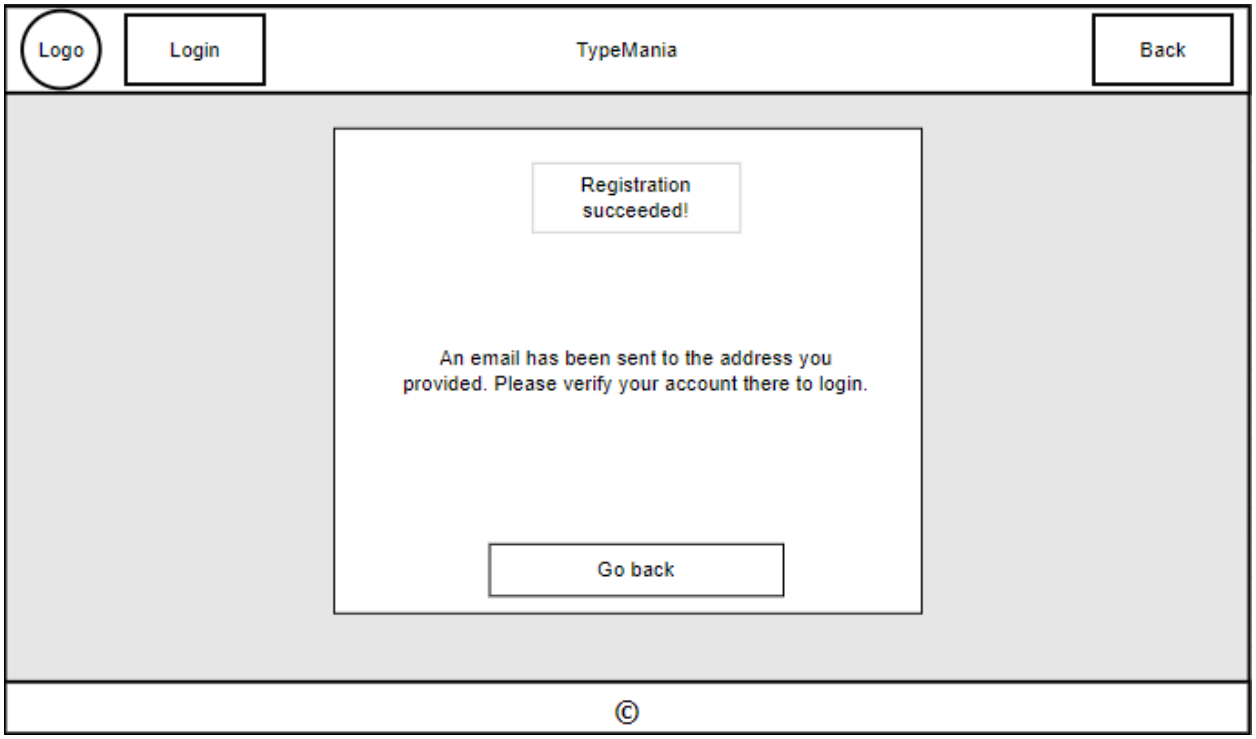


Figure 4.1.6: Registration succeeded page after entering the correct information into the registration page and clicking "Register".

7. The user will click "Go back" and do the rest from their email. The user is considered registered at this point so we conclude our use case here.

UC-13 Login - Allows the registered user to login to be able to access the system as a registered user and access more functionalities.

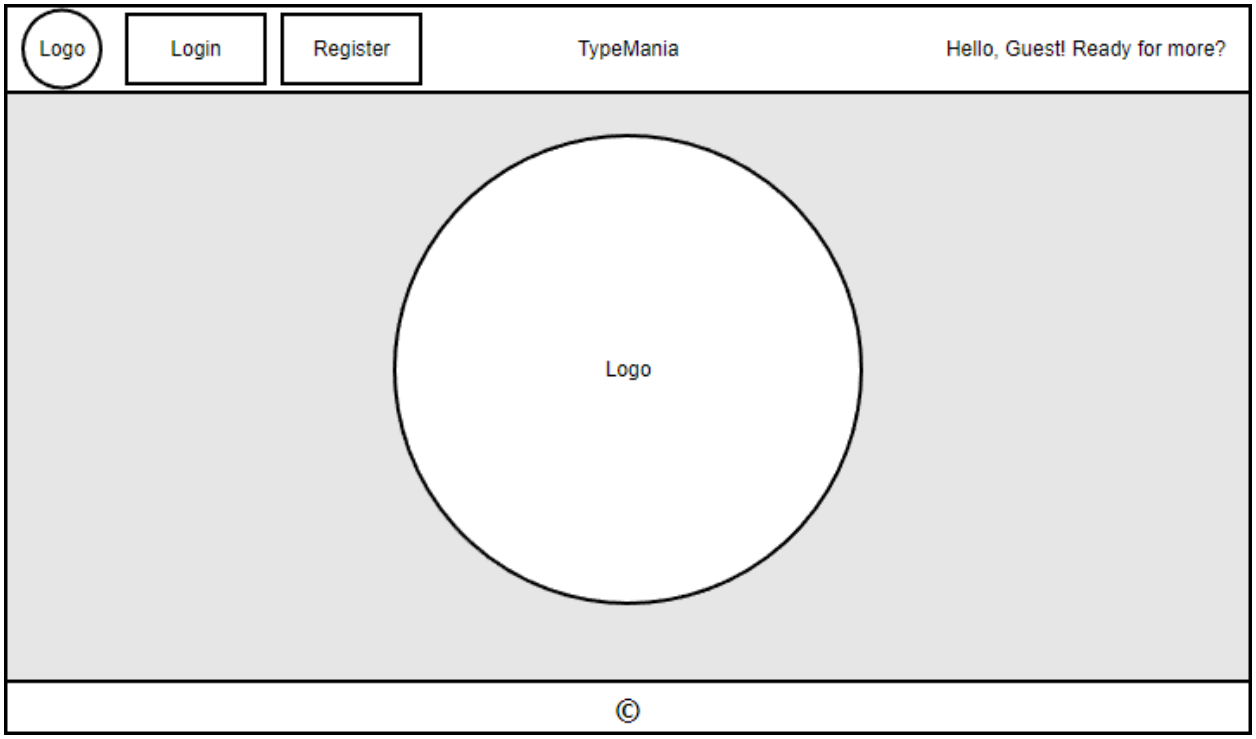


Figure 4.1.1: Landing page for guests.

1. Starting from the landing page as a guest, the user will click “Login”.

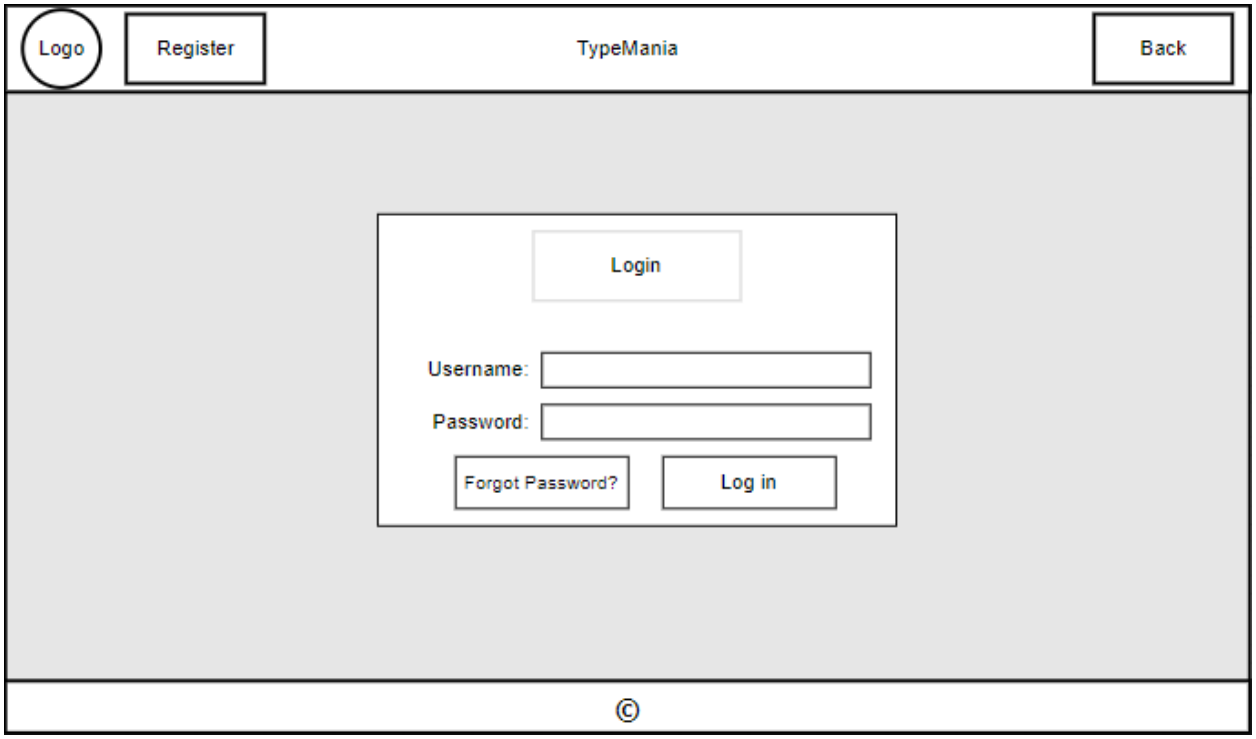


Figure 4.1.7: The login page after clicking “Login”.

2. The user clicks the field to the right of the “Username:” text and enters their username.
3. The user clicks or uses tab on the keyboard to navigate to the field to the right of “Password:” and enters the password associated with the entered username.
4. The user clicks “Log in” or presses enter on the keyboard.

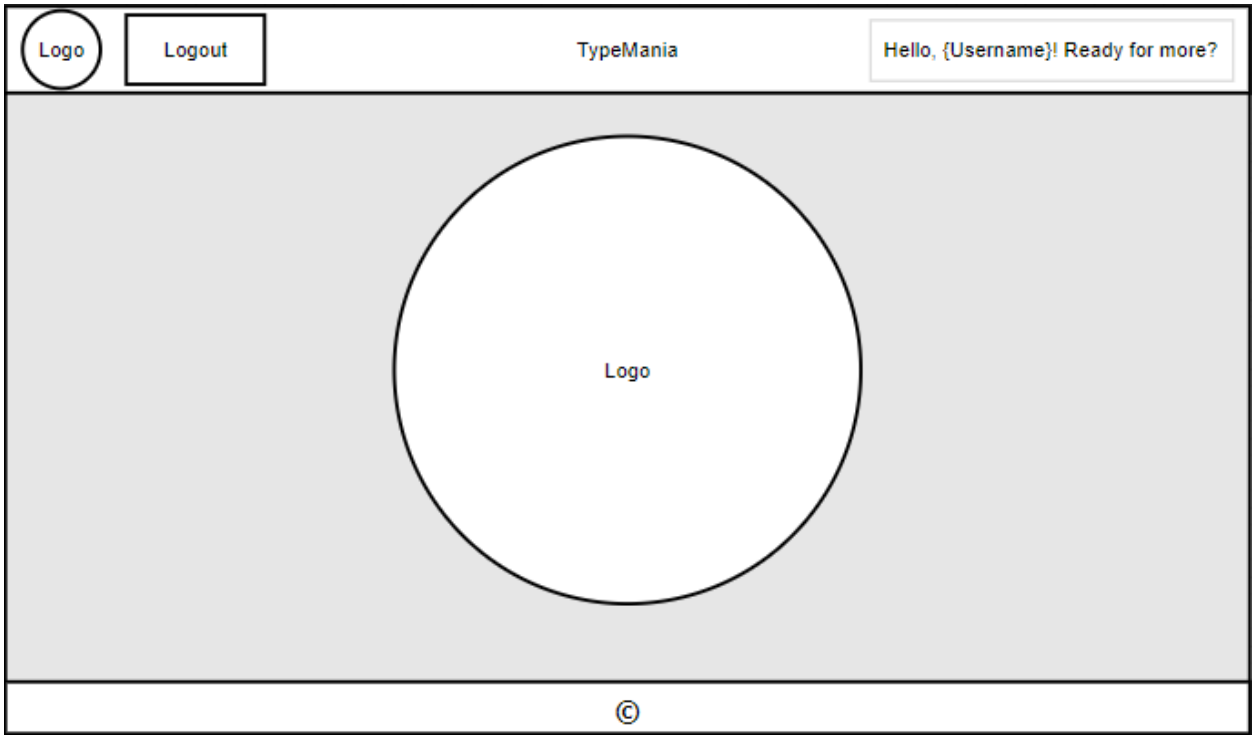


Figure 4.1.8: Landing page for logged in registered users.

5. If authentication is successful, the user will be brought to the landing page for logged in registered users. The user is considered to be logged in at this point so we conclude the use case here.

UC-12 View Statistics - Allows the registered users to view their statistics.

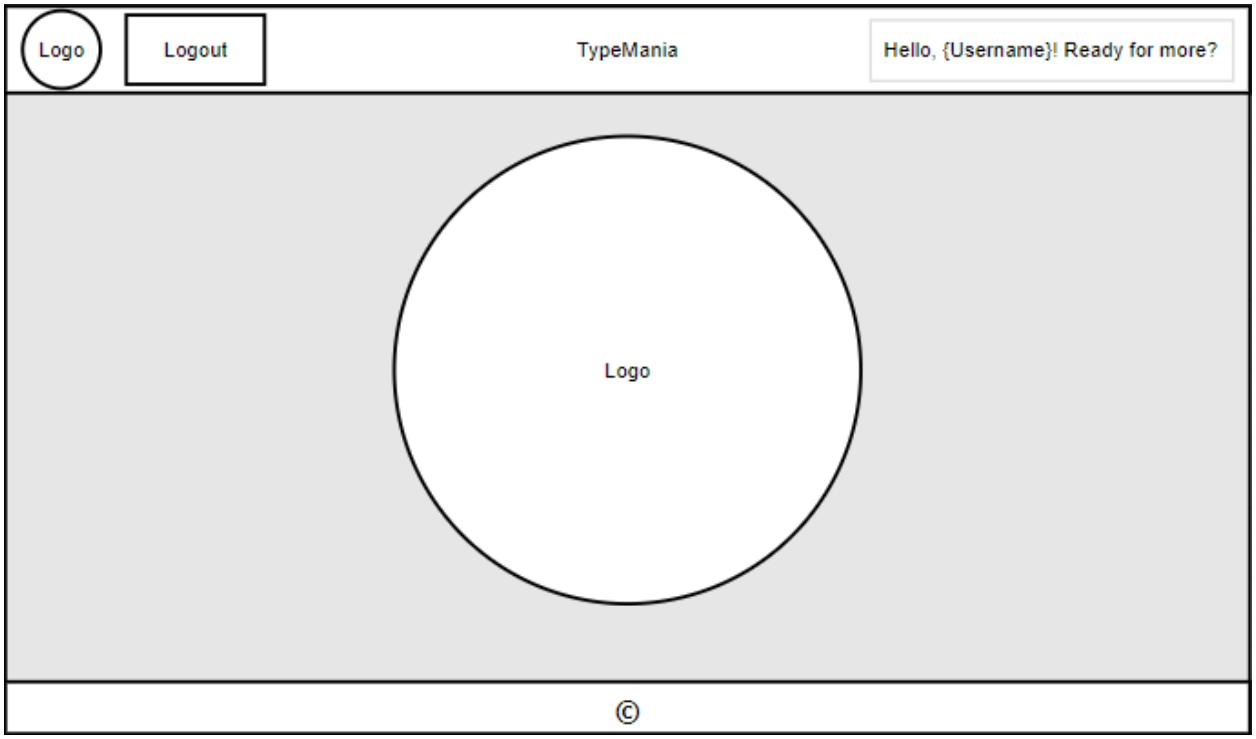


Figure 4.1.8: Landing page for logged in registered users.

1. Starting from the landing page as a logged in registered user, the user will click the central logo to reveal main menu options.

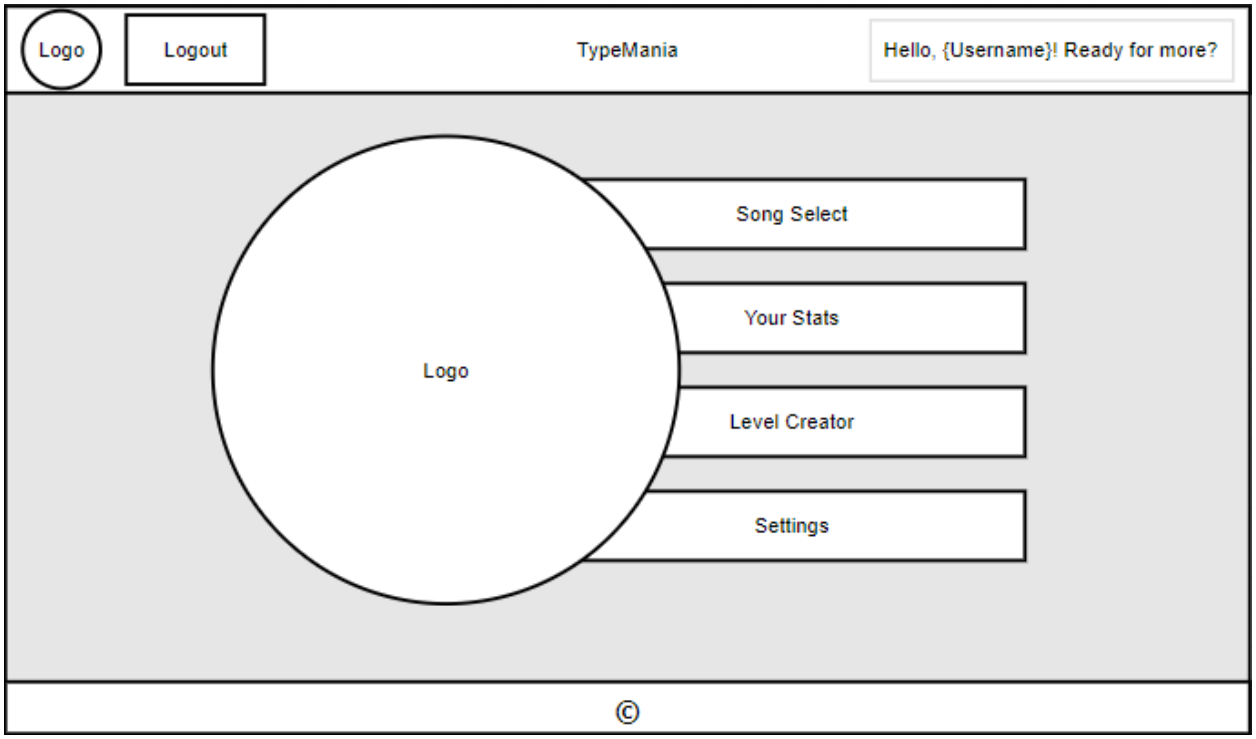


Figure 4.1.9: Expanded landing page for registered users.

2. The user will click “Your Stats” from the main menu options.

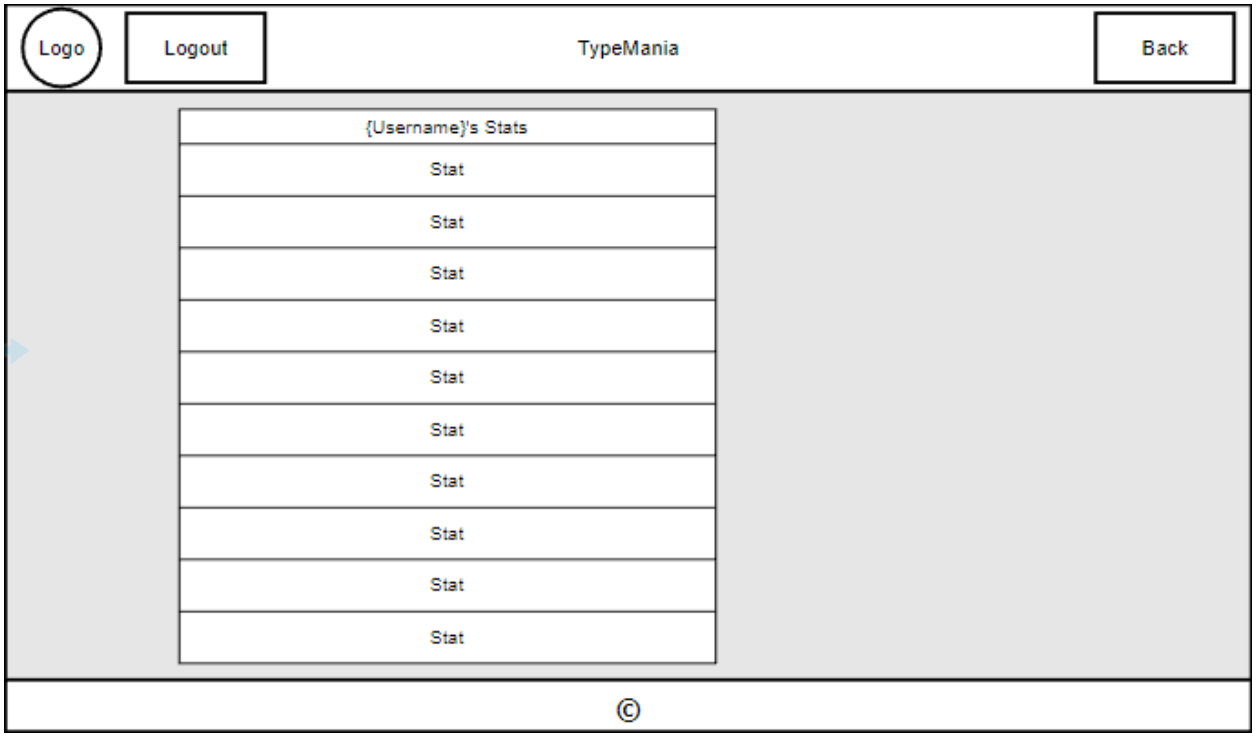


Figure 4.1.10: Statistics page with historical data on the registered user. Accessed by clicking "Your Stats" from the expanded landing page while logged in.

3. The user is considered to be viewing their statistics at this point so we conclude our use case here.

4.2 User Effort Estimation

Usage Scenario: Guest User Game Play

A. **NAVIGATION**: Best Case: total 3 mouse clicks

— after navigating to the web page shown below —

1. Click the logo to reveal main menu options
2. Click "Song Select" from the main menu options

— From here the user may wish to select a different song from the one already selected. In such a case they will click a "Song" box to turn it into the "Selected Song" box —

— <u>Worst case scenario</u>: the user clicks a "Song" box that isn't the "Selected Song" infinitely and never moves to the next step —

3. Click "Selected Song" from the song select screen

B. **DATA ENTRY**: Best Case: total 1 mouse click and 342 keystrokes (18 keystrokes to navigate to the website from the browser and average 324 keystrokes per song)

1. Type in the TypeMania website url into the browser address bar
    a. ex. www.typemania.com
2. Press “Enter”
   — <u>Worst case scenario</u>: the user incorrectly types in the web address and needs to retype the correct url —
   — after completing song selection as shown above —
3. Press matching key as key graphic appears in hit zone on game play screen

** In calculating the number of keystrokes per gameplay, the average song contains 108 beats per minute, and the average song is 3 minutes; this results in the average beats, or keystrokes per song, to equal about 324 (Fountain, 2020). This estimate may vary based on song selection.*

Usage Scenario: New User Sign Up (precondition: navigated to website)

A. **NAVIGATION**: Best Case: total 2 mouse clicks

1. Click “Register” button on home page menu
   — after entering verified username, email, password, and confirm password in the appropriate text boxes shown in data entry below —
2. Click “Register”
   — after clicking “Register”, the user will be directed to check their email to confirm their email address, then login —
   — <u>Worst case scenario:</u> continuously clicking on wrong menu option and never being redirected to desired register page —

B. **DATA ENTRY**: Best Case: total 4 mouse click and # of characters in username and password keystrokes (51 keystrokes from example entries shown below)

1. Click the “username” text box
2. Type username in the text box
    i. Ex. acstevens
3. Click the “email” text box
4. Type user email in the text box
    i. Ex. acstevens@fhsu.edu
5. Click the “password” text box
6. Type the user password in the text box
    i. Ex. password123!
7. Click the “confirm password” text box
8. Type the user password in the text box
    i. Ex. password123!

— if username is not available, the "Register" window will show a warning message and will not allow the user to submit the form, the user will need to enter a new email that has not yet been used to create an account —

— <u>Worst case scenario:</u> continuously entering unavailable usernames and incorrectly matching passwords and never able to click "Register" —

Usage Scenario: Registered User Login (precondition: navigated to website)

A. **NAVIGATION**: Best Case: total 2 mouse clicks

1. Click "Login" button on home page menu

   — after entering correct username and password in the appropriate text boxes shown below —

   — <u>Worst case scenario</u>: continuously clicking on wrong menu option and never being redirected to desired login page—
2. Click "Login"

   — if username and password verified by authenticator, the user is redirected to their homepage, otherwise, the user is notified that username and password were incorrect and must be entered again —

   — <u>Worst case scenario:</u> continuously entering incorrect login credentials and attempting to login continuously —

B. **DATA ENTRY**: Best Case: total 2 mouse click and # of characters in username and password keystrokes (21 keystrokes from example entries shown below)

1. Click the "username" text box
2. Type username in the text box
   i. Ex. acstevens
3. Click the "password" text box
4. Type the user password in the text box
   i. Ex. password123!

— <u>Worst case scenario:</u> making errors in text boxes and needing to delete and retype characters —

Usage Scenario: Registered User Game Play (precondition: logged in)

A. **NAVIGATION**: Best Case: total 3 mouse clicks

1. Click the logo to reveal main menu options
2. Click "Song Select" from the main menu options

   — From here the user may wish to select a different song from the one already selected. In such a case they will click a "Song" box to turn it into the "Selected Song" box —

— <u>Worst case scenario</u>: user clicking a “Song” box that isn’t the “Selected Song” infinitely and never moving to the next step —

3. Click “Selected Song” from the song select screen

Usage Scenario: Registered User View Statistics (precondition: logged in)

A. **NAVIGATION**: Best Case: total 2 mouse clicks

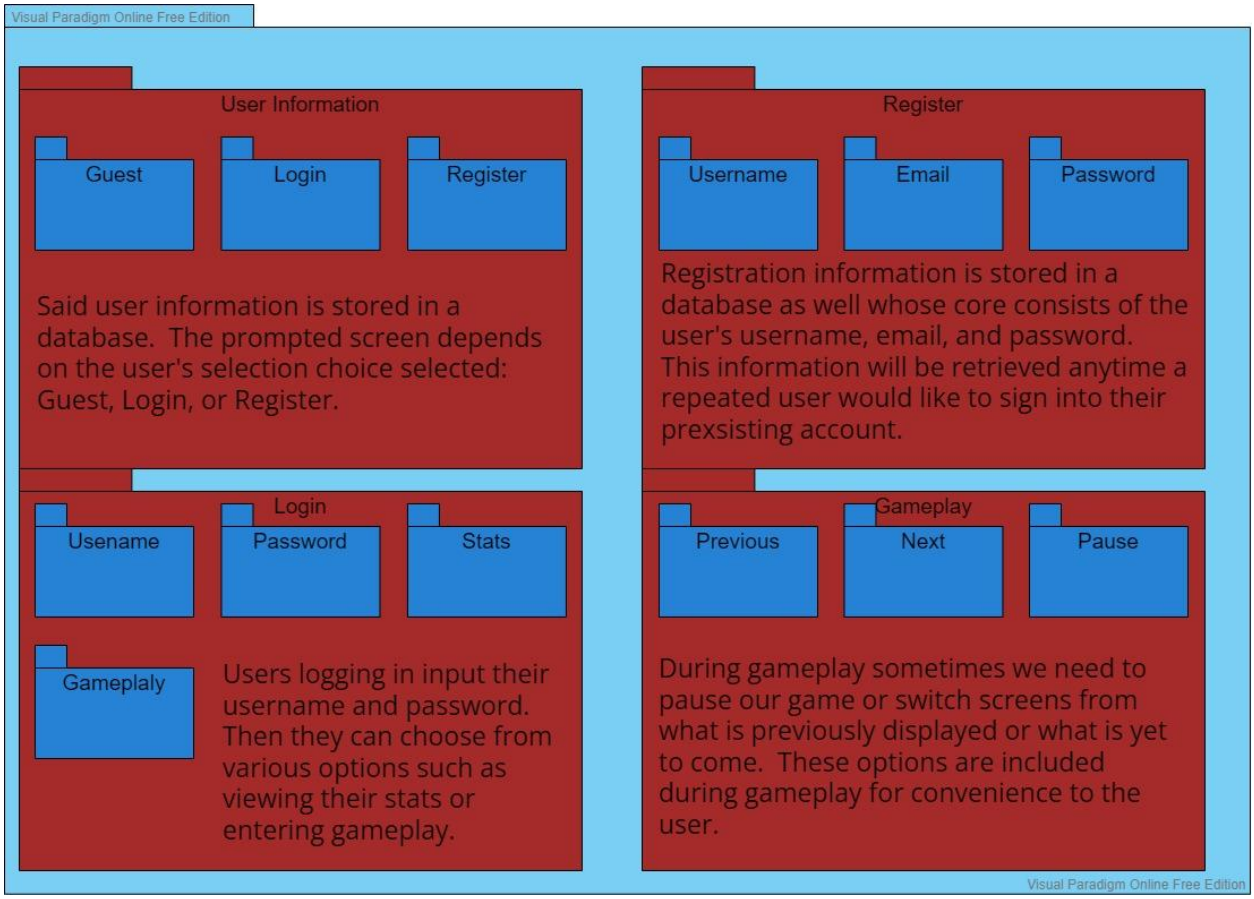
1. Click the logo to reveal main menu options
2. Click “Your Stats” from the main menu options

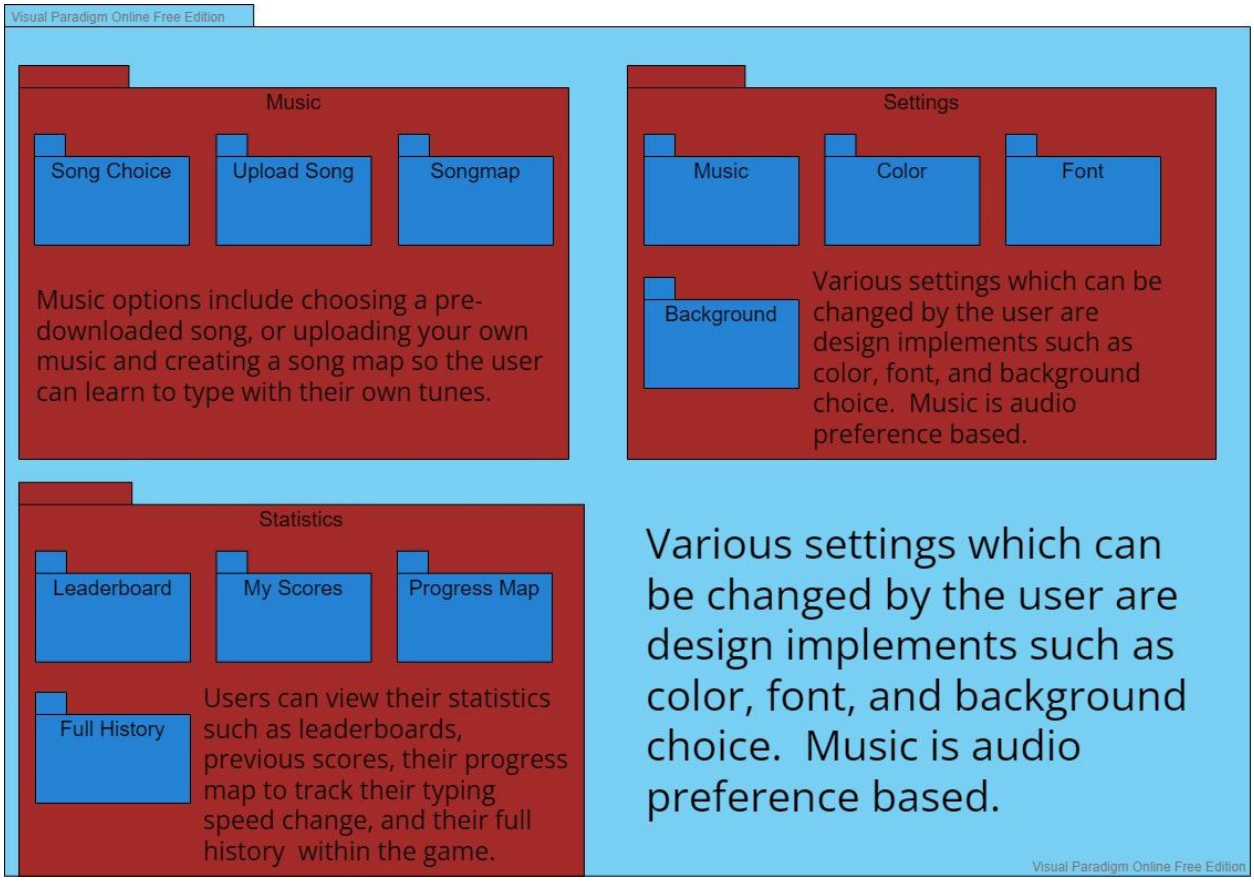
— User can view performance statistic summary —

— <u>Worst case scenario</u>: continuously clicking on wrong menu option and never being redirected to desired stats page—

5. System Architecture

5.1 Identifying Subsystems





Each image above is categorized in folders displaying the subsystems of TypeMania. The primary subsytems are user information, registering, login, play game, music, settings, and statistics. Inside of each subsystem contains components that will be found within that area of TypeMania froma development standpoint. The team has decided to use one a one webpage layout minimizing the mapping. However an order pertains to everything being that first the user will either choose to play as a guest, login, or create an account. Then the user will edit their settings and choose their song based on their music preferences. Finally, playing it is time to play the game. After gameplay the user can view their statistics, play again, or return to the home screen.

5.2 Architecture Styles

TypeMania displays data-centered architechure when the user first enters the homepage. The center of this is user information stored within a database such as their username, email adress, password, security questions, and gameplay statistics. Various features revolve around said data such as logging in, viewing various statistics, and creating a user account.

Subprogram architecture can be found within the data-centered architectural style as well. TypeMania is layered. For example if you go into the various settings the player can change before game play the music option is not just choosing a song. Music consists of choosing or uploading your own song. Then, if the user uploads their own song they can create a songmap to apply to the gameplay in correlation with the song they uploaded. Another example is the statistics menu after gameplay. The user can glance at their score and replay the game or return to the home screen. However the various layers exsist so the user can view how they rank compared to the leaderboards or view their history in order to track their progress.

This layered architecture is created so the data is entered on the top level and as more information is gathered, the program works its way down to the core level which is usually the database. In TypeMania this is where the statistics and other use information is stored. This specific architecture is most common due to the fact that it is maintainable, testable, simple enough to divide amongst group members, and easy to update. When updating each layer can be updated separately, making it easier to identify various bugs and errors which may appear.

5.3 Mapping Subsystems to Hardware

The system will be deployed on two separate virtual computers on Heroku. On one machine, a PostgreSQL database will be deployed. On the second machine, the application itself will be deployed. The database will act as a microservice to request queries and respond with results over HTTP. On the second machine, the javascript and static contents that will be rendered on the client side by the user's web browser, will be served. The application will send the queries over HTTP to the PostgreSQL server and respond to the client side with the query result. Also, as the number of users grows, the number of machines could be scaled up to meet higher demand and more traffic.

5.4 Connectors and Network Protocol

The system will use the standard Hypertext Transfer Protocol, HTTP. HTPP is an important protocol between the client and application interface. The users will be able to access the contents of the application via HTTP. They can also send request queries, which are sent to PostgreSQL server by the application, and receive query results all over HTTP.

5.5 Global Control Flow

Execution Orderliness

Overall, our system is event-driven with various procedure driven aspects. When first navigating to the website, the system is event-driven, in which the user has multiple procedure-driven actions they can generate. These actions include guest play, register, and

login. If a player chooses to login, this triggers a larger event-driven experience, including more procedure-driven actions, like song select, your stats, level creator, and settings.

Each procedure-driven action has linear steps the user must follow to complete. For example, if the user chooses to register, they click the login option, enter in the required text inputs, click register, then verify their account in their personal email. Or if the user chooses to play game, they must click the logo to reveal the main menu options, click song select, and click the chosen song they want to play to begin the game.

The actual game play is also procedure-driven as the screen shows various notes the user must select as the song plays. Once the song ends, the game ends, and the user is shown their score. Then, the system is event-driven as the user can decide to play again, exit to the main menu, or view statistics, all of which have varying subsequent event-driven and procedure-driven actions to be performed.

Time Dependency

There is event-response time dependency for two aspects in this system with no concern for real-time: user login and game play. Because this is a web-based game with a user login feature, there will be an event-response time dependent aspect regarding the login of a user. For example, once the user logins, after a period of inactivity, the user will be automatically logged out. For example, if the user does not manually logout, the user will be automatically logged out after a period of 24 hours of inactivity through the expiration of session cookies in the user's browser.

The game play also has an event-response time dependency. For example, each game play session will last the duration of the selected song. Therefore, the game play is time dependent on song length. Once the song completes, the game will end. As a result, each game play session is dependent upon the selected song.

5.6 Hardware Requirements

Our system is a web application that will be hosted via Heroku. That is, the server-side equipment is outsourced by Heroku. The client-side hardware can be found in most modern laptops and desktops and refers to the hardware requirements for users.

Server-side hardware requirements

Database storage space

The hosting database will need enough long term storage space to hold several songs and quality graphics. The amount of songs is expected to grow continually over time, so the expected hard disk space required for the term is approximately 512MB.

Internet connection

The database server must be able to communicate with its clients from a remote location. To communicate in this way the database server will make use of the internet. To serve multiple clients simultaneously, it is recommended that the database's network bandwidth be able to transfer at a rate of at least 512kbps.

RAM / Main Memory

The database server must communicate with only a few clients at one time with a relatively small database. The database server will utilize 512MB of RAM to handle these interactions.

Client-side hardware requirements

Color display

The user should be able to see the user interface in order to interact with it properly. The user's computer system should have access to a monitor to view the graphics. The minimum recommended resolution for such a monitor is 640 x 480 pixels.

Physical keyboard

The user must be able to see most of their display whilst they type, so on-screen keyboards aren't recommended. The minimum character set makes use of the alpha characters, so the user should have access to a physical keyboard that can emulate such keys.

Internet connection

The user must access the database to request the code to be loaded to their computer. To make this request, internet access through a modern web browser is required. Because the game makes use of graphics, a minimum bandwidth of 128kbps is recommended.

RAM / Main Memory

The user must use a modern web browser for the game to load properly. Between the user's OS, web browser, and web application, a RAM of at least 2GB is necessary.

6. Project size estimation based on use case points

Unadjusted Actor Weight (UAW)

Actor	Description of relevant characteristics	Complexity	Weight
Database (Participating)	Another system that interacts with our system through a well defined API.	Simple	1
Guest User (Initiating)	A person interacting with our system via GUI.	Complex	3
Registered User (Initiating)	A person interacting with our system via GUI.	Complex	3
		UAW Total:	7

Unadjusted Use Case Points (UUCP)

Use Case	Description of relevant characteristics	Category	Weight
UC-7 Play Game	Simple user interface interaction of 3 steps for main success scenario. One Participating Actor (Database).	Simple	5
UC-12 View Statistics	Simple user interface interaction of 2 steps for main success scenario. One Participating Actor (Database).	Simple	5
UC-2 Register	Moderate user interface interaction of 6 steps for main success scenario. One Participating Actor (Database).	Average	10
UC-14 Login	Moderate user interface interaction of 4 steps for main success scenario. One Participating Actor (Database).	Average	10
		UUCP Total:	30

Technical Complexity Factors (TCF)

Technical Factor	Description	Weight	Perceived Complexity	Calculated Factor
T1	Distributed system.	2	3	2 * 3 = 6

T2	Performance objective.	1	5	1 * 5 = 5
T3	End-user efficiency.	1	2	1 * 2 = 2
T4	Complex internal processing.	1	1	1 * 1 = 1
T5	Reusable design or code.	1	4	1 * 4 = 4
T6	Easy to install.	0.5	0	0.5 * 0 = 0
T7	Easy to use.	0.5	3	0.5 * 3 = 1.5
T8	Portable.	2	0	2 * 0 = 0
T9	Easy to change.	1	3	1 * 3 = 3
T10	Concurrent use.	1	3	1 * 3 = 3
T11	Special security features.	1	1	1 * 1 = 1
T12	Direct access for third parties.	1	0	1 * 0 = 0
T13	Special user training.	1	0	1 * 0 = 0
			Technical Factor Total:	26.5
TCF = 0.6 + 0.01 * 26.5 = 0.865				

UCP = UUCP * TCF
= 30 * 0.865
= **25.95** or **26** use case points (only for the use cases elaborated in this report)

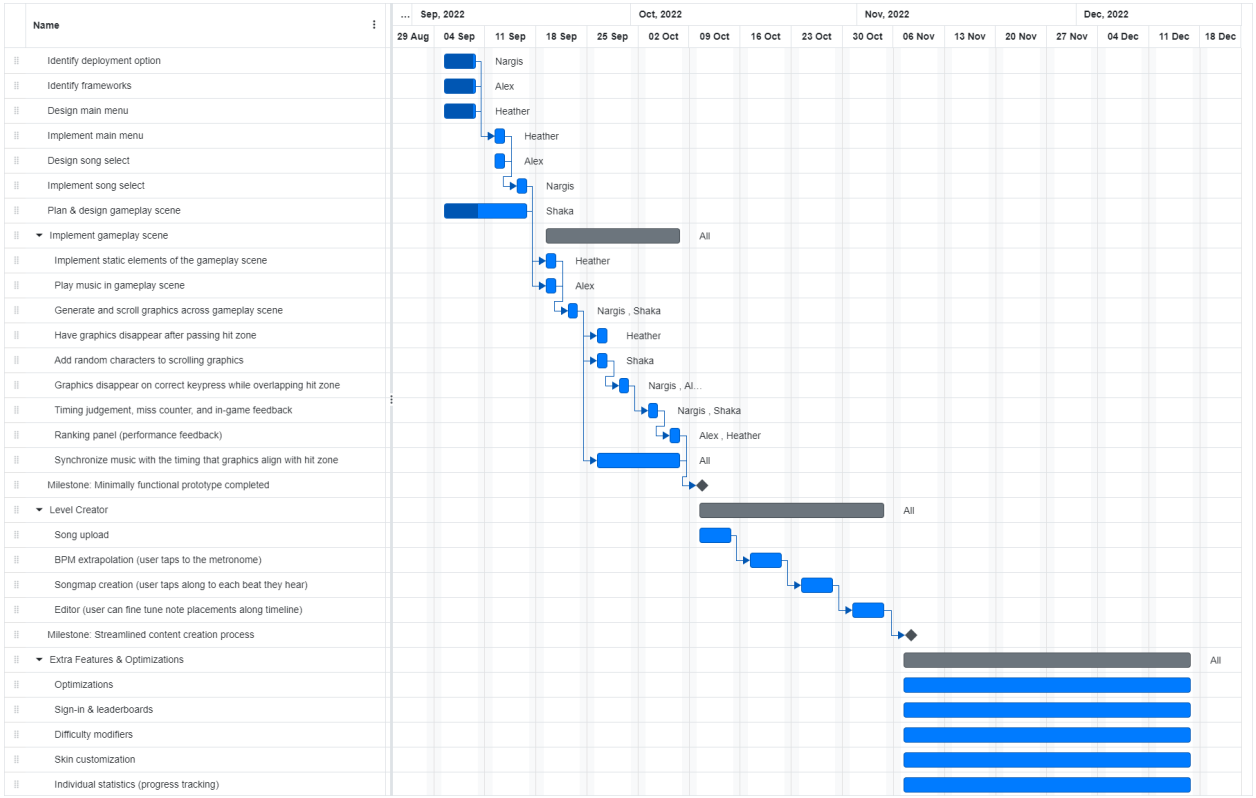
7. Plan of Work

Project Management

Plan from now until the end of the semester

After the submission of report one the group plans to push for a minimally functional prototype. After establishing all of the non-negotiable core mechanics of the game in this prototype, we plan to streamline the process for adding additional playable songs. Extra features such as leaderboards, sign-in, skin customization, and more will be added as time allows. Below is the projected timeline which provides more details.

Roadmap with projected milestones and accomplishment dates



[Full size image here.]

Product Ownership

Functional Features: What the user will be able to do with our system	Description	Pair Responsible
Play a songmap	Users may type the specified characters to the beat of the song they chose from the song select screen.	Shaka/Nargis
Receive feedback	Users are assigned a score highlighting miss count and mistiming info based on their performance in the songmap.	Shaka/Nargis
Skin / Change graphical elements	Users can customize the look and feel of the graphical gameplay elements.	Alex/Heather
Create a level	Users can upload their own song and create a map for it to share with others.	Shaka/Nargis
View leaderboards	Users may view other users' top scores on songmaps.	Alex/Heather
Modify songmap options	Users can fine tune which characters can show up or how fast the music playback is.	Shaka/Nargis
Sign in	Users can retrieve their data across different computers & browsers such as: customized settings, previous scores, skin options, etc.	Alex/Heather

Breakdown of responsibilities

Task	**Responsible Member(s)**
Discord & github management	Shaka
Google docs management	Heather
Assignment task scheduling & management	Nargis
Project task scheduling & management	Shaka
Identify deployment option	Nargis
Identify frameworks	Alex
Design main menu	Heather
Design gameplay scene	Shaka
Design song select	Alex
Implement main menu	Heather
Implement song select	Nargis
Implement static elements of gameplay scene	Heather
Play music in gameplay scene	Alex
Generate and scroll graphics across gameplay scene	Nargis, Shaka
Have scrolling graphics disappear after passing the hit zone	Heather
Add randomized characters to each scrolling graphic	Shaka
Have scrolling graphics disappear when the character on it is pressed during the window of time when the graphic overlaps the hit zone	Nargis, Alex
Synchronize the music with the timing that	Team effort

the graphics align with the hit zone.	
Timing judgment, miss counter, & other during play user feedback	Nargis, Shaka
Ranking panel and other after play user feedback	Heather, Alex
Song upload	Heather
Level creator/editor	Team effort
Leaderboards	Alex
Skin customization	Heather
Difficulty modifiers	Shaka
Difficulty rating system	Nargis
Historical stats for users to track their progress	Alex
Optimizations	Nargis

Sources

Cheung, S. Y., & Ng, K. Y. (2021). Application of the Educational Game to Enhance Student Learning. *Frontiers in Education*, *6*, 1–10. https://doi.org/10.3389/feduc.2021.623793

Davies, J. J., & Hemingway, T. J. (2014). Guitar Hero or Zero? *Journal of Media Psychology*, *26*(4), 189–201. https://doi.org/10.1027/1864-1105/a000125

Kim, S., Song, K., Lockee, B., & Burton, J. (2017). *Gamification in Learning and Education: Enjoy Learning Like Gaming (Advances in Game-Based Learning)* (1st ed.). Springer.

Marom, H. W., & Weintraub, N. (2015). The effect of a touch-typing program on keyboarding skills of higher education students with and without learning disabilities. *Research in Developmental Disabilities*, *47*, 208–217. https://doi.org/10.1016/j.ridd.2015.09.014

Fountain, T. (2020, July 13). *How Many Bars is a 3-Minute Song? + 9 Three-Minute Song Videos - OpenMic*. Open Mic UK. Retrieved September 17, 2022, from https://www.openmicuk.co.uk/advice/how-many-bars-is-a-3-minute-song/#:%7E:text=Taking%20into%20account%20all%20types,the%20song%20divided%20by%20four.

osu! (2022, April 24). Osu! Retrieved September 17, 2022, from https://osu.ppy.sh/home

[(osu!, 2022) used as inspiration for the design of the TypeMania user interface illustrated in section 4.1 Preliminary Design.]