CSN-254
Software Engineering
Design Document

Group-4

Name	Roll No	Contribution
Meet Sindhav	22114053	Discussion, preparation of diagrams, articulation and documentation
Mohammed Haaziq	22114055	Discussion, preparation of diagrams and final review
Aditya Mundada	22114058	Discussion, preparation of diagrams, and articulation of document
Nayan Kakade	22114060	Discussion, preparation of diagrams and final review
Polasa Hariharan	22114068	Discussion, preparation of diagrams and final review
Roopam Taneja	22125030	Discussion, preparation of diagrams and final review

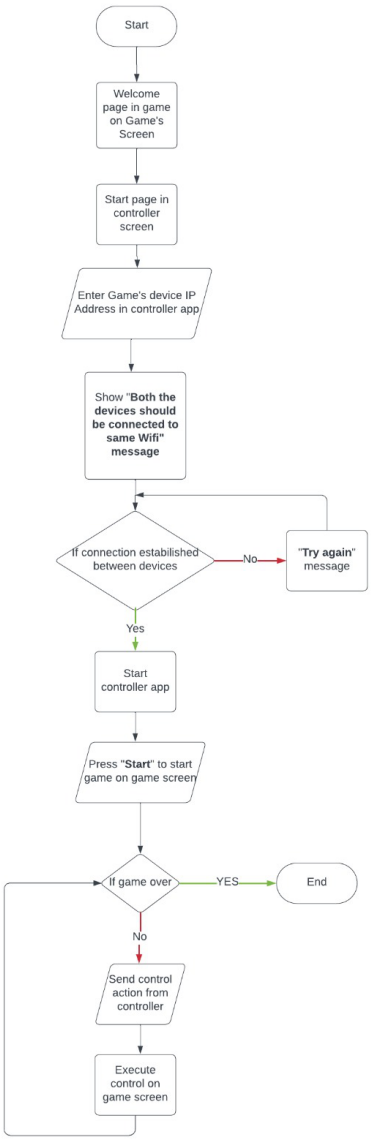
SUMMARY

- **User interface:** For the user, we have designed an app with intuitive, simple and clear controls that support haptic feedback. This will make gaming a fun and joyful experience as the clean UI along with features that help gamer know the current status of game and control it easily are made available. For the PC game, the animations are clear and we have various arenas made available to the users that help them play as per their choice. Also the smooth transitions help get proper real time experience which is fundamental to our proposition. For the PC app, it is implemented in Python and for mobile controller, it is implemented in Flutter.
- **Security:** We have tried to encrypt the IP address as a game key using robust mechanisms that ensures secure communication. In order to not allow noises in the networking involved we have ensured that the players who connect are subject to proper verification.
- **Functionality:** In order to implement all attacks and also perform the required background tasks like verification and data transfer we have divided our project into various modules, some dealing with foreground and some with background activities exclusively. This makes project design properly structured and ensures that implementation of functionalities does not interfere with each other.
- **Adaptability:** Currently our game can work on all desktops supporting python and our app can work on all devices that are based on android OS. This caters to a wide range of users.
- **Performance:** We have strived to maintain optimal resource utilization by including robust algorithms in our code and ensuring that system crashes and bugs do not occur.

DIAGRAMS

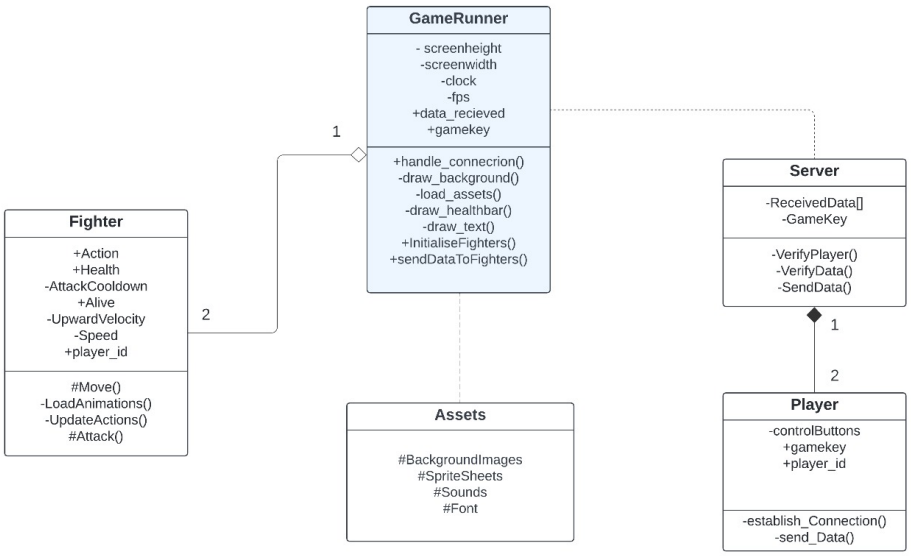
The following relevant diagrams were constructed as a follow up to our design review mentioned above:

Flow Chart:

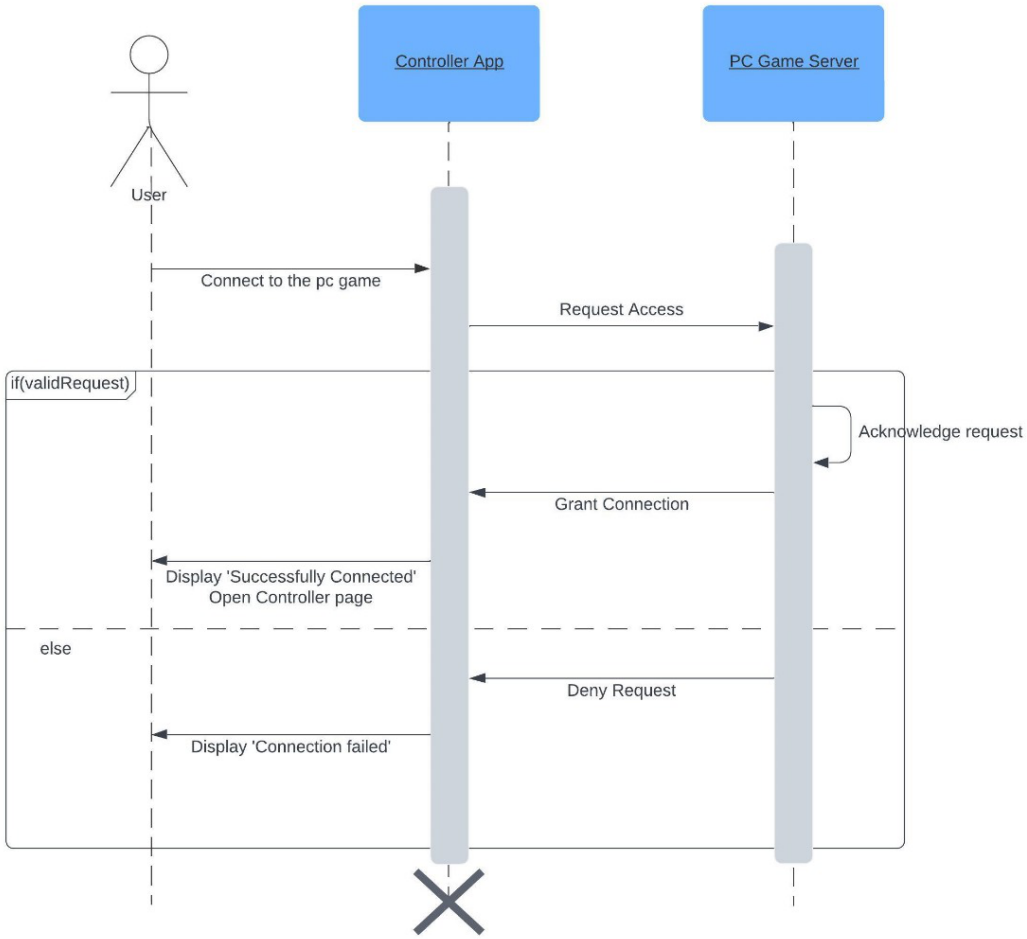


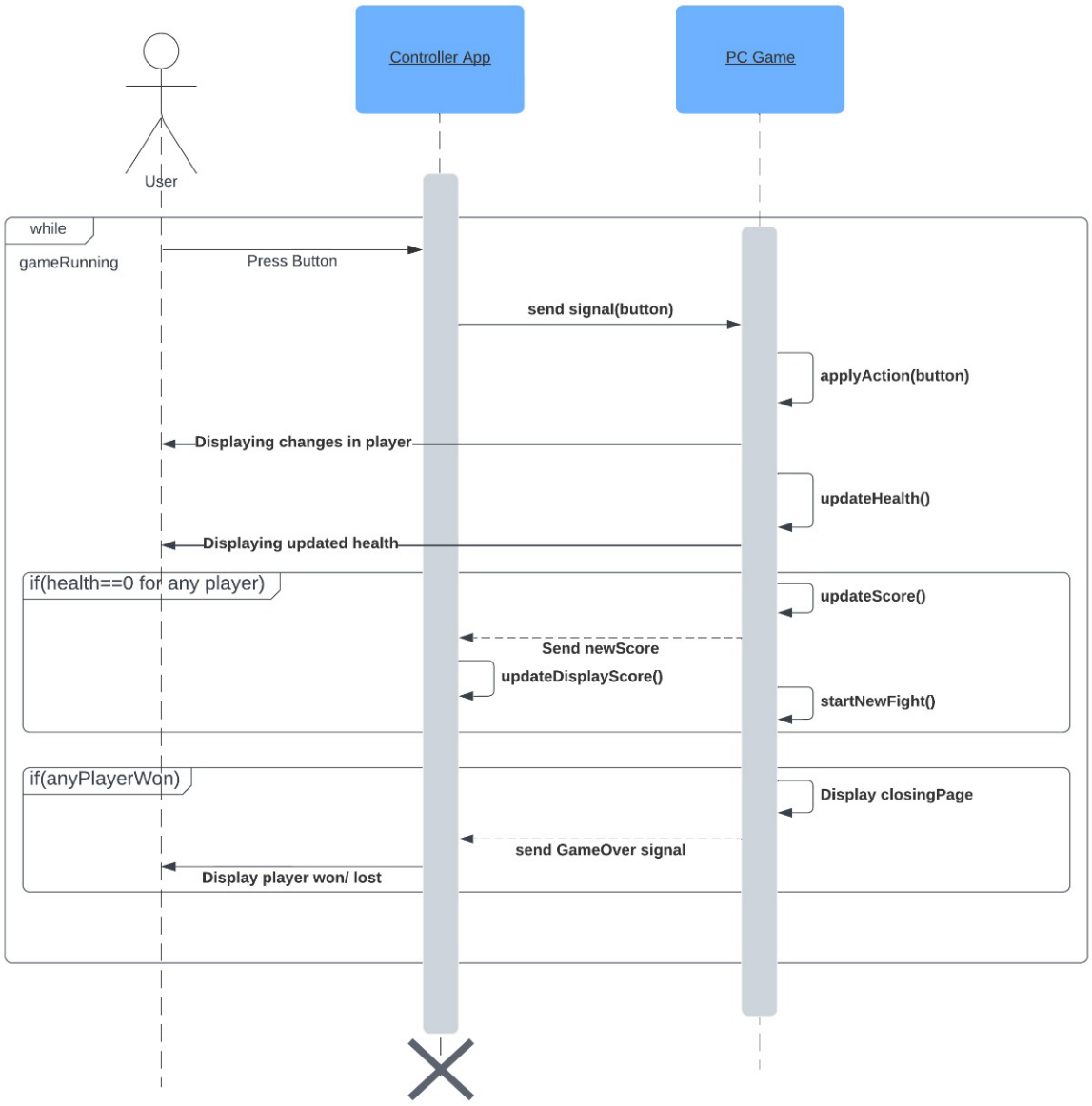
Use Case Diagram:

Class Diagram:



Sequence Diagrams:




Controller App
PC Game
User
while
gameRunning
Press Button
send signal(button)
applyAction(button)
Displaying changes in player
updateHealth()
Displaying updated health
if(health==0 for any player)
updateScore()
Send newScore
updateDisplayScore()
startNewFight()
if(anyPlayerWon)
Display closingPage
send GameOver signal
Display player won/ lost

ALGORITHMS

- Data reception from touch controls of mobile app.
- Encryption of IP address to ensure security.
- Sending touch control data to PC GameRunner file in proper format.
- Using Websockets and asyncio for communication across devices.
- Decoding the message received at the server end and converting it to actionable version on which code can work.
- Using the data to control player actions and implementing game features properly ensuring various player parameters like speed, movement, attack type, cooldown, etc.
- Checking conditions for start and end of next gameplay rounds and ensuring its proper implementation.

Future modifications in later versions:

- Making a computer play against 1 player and supporting different difficulty levels using various algorithms that will determine computer performance.
- Making choice of characters and various arenas random and making special abilities for special characters and in special arenas.

QUESTIONS

- **Are we using appropriate methods to represent the design solutions?** Indeed, we are employing suitable methods to illustrate our design solutions effectively. This includes the use of UML diagrams, encompassing use case diagrams, class diagrams, sequence diagrams, and data flow diagrams (DFD). These diagrams adeptly capture all the essential facets of our design solution, providing comprehensive insights into the system's functionality and architecture.
- **Does the design satisfy the specified requirements?** Our design satisfies all of the functional requirements specified in the SRS document, including real-time communication, UI and cross-device compatibility.
- **Is the user interface design well documented?** While all the UI design elements have been comprehensively described within the design document, it's important to note that detailed diagrams corresponding to the UI elements have not been fully documented at this stage. These detailed diagrams may be incorporated in subsequent versions of the document as the development progresses. This approach allows for a more detailed and refined representation of the user interface as the design evolves.
- **Are non-functional requirements taken into consideration?** Certainly, we have taken measures to take care of non-functional requirements. A unique variable PlayerID is used to distinguish the contols of different players and how they get executed. This increases and security and robustness of system. In similar ways, steps are taken to make the game responsive.
- **Does the design appear to be satisfactory to proceed to the next phase of the development?** Yes, the design document specifies all the required design elements satisfactorily along with necessary diagrams which is sufficient to proceed to the next phase of the development activity.