TeleTag-Requirements - 487W

Cameron Bussom, Love-Divine Onwulata, Mahima Abraham, Greg Salisbury

September 10 2023

1 Team Info and Policies

1.1 Roles of Team Members:

- Cameron Bussom - Front End and Bluetooth
- Mahima Abraham - Front End
- Love-Divine Onwulata - Bluetooth
- Greg Salisbury - Back End

1.2 Project Artifacts

- Git Repo - Click to go to the Git Repo
- Proposal PDF - Click to go to the Proposal PDF

1.3 Communication channels

- Team will meet either in person or on zoom to work on the project.
- Using Github to organize all of our documents and code.
- Use Monday.com to keep track of due dates, who is working on what, and tasks to be done.
- Use discord to communicate about the project and develop a knowledge base for collaboration when necessary.

2 Product Description

- TeleTag is a mobile app that people can download to play tag with their friends. The game will have different rooms that players can join to play tag with each other. Their phones' Bluetooth will help them find other players and once they are in range their tag button will be enabled. One the player is out of the range to tag another player, their button will be disabled. There will be a leader board that displays the player's rankings, wins, and losses.

2.1 Four Major Features

- Detecting Bluetooth devices
- Connecting to a Bluetooth device
- Detecting proximity of Bluetooth devices
- Sending data through Bluetooth devices
- Recognizing devices that are in the current game

2.2 Two Stretch Goals

- Different Game Modes
  - Games that have conditions that once met end the game. i.e Elimination mode, Multiple taggers, etc
- Map System
  - A service that can track all players in proximity to the user

3 Use Cases

3.1 Use Case 1: User Starts a Game

- Actor: User
- Overview: A user wants to play a game with n other users. The user connects all mobile devices via Bluetooth
- Trigger: A user and n other users want to play TeleTag.
- Precondition: n users want to play a game of TeleTag.
- Post-condition: N users phones have the data and Bluetooth connections of n-1 users
- Steps:
  1. N users gather to play a game of TeleTag
  2. One user presses the host game button
  3. The users not hosting, gather in range of the host user's device's Bluetooth range
  4. The phones connect via Bluetooth
  5. The of of all the users phones are sent to each device

- Extensions
  - One user decides to not play, and does not connect to the game
  - Another user decides to play as well, and connects to the other phones
- Exceptions:
  - One or more user's device is not connecting to the other phones correctly and is kicked from the lobby

3.2 Use Case 2: User Leaves a Game

- Actor: User
- Overview: A user wants to leave the current game permanently. The user clicks on the 'Leave Game' button. The user's device no longer transmits data to other devices.
- Trigger: A user wants to disconnect from the game
- Precondition: A user wants to stop playing the current game of TeleTag.
- Post-condition: The user's device no longer transmits data to the other devices in the game
- Steps:
  1. The user stops running
  2. The user presses the 'Leave Game' button
  3. The device transmits to the other devices that it is leaving the game
  4. The device stops transmitting data
  5. The database removes the player from the player database
  6. The devices of the other players no longer transmit to the former player's device
- Extension
  - The user's device stops working or dies and is removed from the player database
- Exception:
  - The user's device fails to update the player database, and the other users phones continue to send data

3.3 Use Case 3: The Host Ends the Game

- Actor: User
- Overview: The host decides that the game should end and presses the 'End Game' button. The host's device sends a signal for the other devices to disconnect from the game and stop transmitting data.
- Trigger: The host presses the 'End Game' button
- Precondition: The host user wants to end the game
- Post-condition: The users' devices stop transmitting data to one another
- Steps:
  1. The host presses the 'End Game' button
  2. The host device transmits to the other devices that the game has ended and instructs them to stop transmitting data to one another
  3. The other devices receive the instruction to end the game and stop transmitting data
  4. The player database removes all the player
  5. The game ends
- Extension
  - All the users' devices stops working or dies and are removed from the player database, ending the game
  - The game goes on for too long, and the player database removes all the players and ends the game
- Exception:
  - The devices are not updated on the game's status and continue to transmit data as if still in the game

3.4 Use Case 4: A User is 'Tagged'

- Actor: User
- Overview: The user that has the tag gets in range of another user. The tagger then presses the 'Tag' button, transferring the ability to tag to the other player. The ability to tag then goes on cool-down for a set time, preventing the new tagger from tagging another player until the cool-down is finished.
- Trigger: The tagger, in range of another player, hits the tag button
- Precondition: The tagger is in tagging range of a player

- Post-condition: The tag has been transferred from the user to the other player
- Steps:
  1. The tagger gets in range of a player they desire to tag
  2. The user presses the ’Tag’ button
  3. The tag is transferred to the player
  4. The player receiving the tag is notified that they have been tagged
  5. The tag then goes on cool-down for a set time
- Extension
  - There are multiple players in range, and the user has to select which to tag
  - The tagger presses the ’Tag’ button, but the tag is in cool-down resulting in nothing happening
- Exception:
  - The tag is not in cool-down, and another player is in range, but the device does not transmit the tag

4 Non-Functional Requirements

1. Usability: An easy and efficient game that players can easily join the game and connect with other users to play the game this will allow for anyone to be able to play TeleTag
2. Scalability: Multiple players are able to join at once and choose if they want to join a current game or start a new one. The limit of players will be salable to however many players are in the same vicinity of each other.
3. Security/Privacy: Database that stores players’ information that cannot be accessed by anybody but the player themselves. This information will not have any person information only game information so nothing personal will be accessed by TeleTag.

5 External Requirements

5.1 Tools Being Used and Why:

- Android Studio - A great Android app developer that has different tools and features, that is constantly updated, which can be used to develop a good apps. Also the Android emulator helps us to see what the app will look like as we design it without having to use an actual android device.

- Kotlin - Google's preferred programming language for Android app development, Kotlin helps us code more efficiently, is easy to pick up, and provides code safety.
- SQLite - Is an open-source database engine that is already installed on Android devices. We can use SQLite to easily create a database to store player information. Due to our familiarity with MySQL, and since this software is recommended by Android, it seems like the obvious choice for the project.

5.2 Team Roles and Justification

- Cameron Bussom - Front End and Bluetooth
  - A front end developer is needed in order to be able to create different functions for the players in the game such as the history tab and tag button. The role of developing the Bluetooth is needed in order to connect the players' phones and to determine the range of each player. Cameron has experience creating many different types of front ends and has an interest in Bluetooth technology which would make him perfect for the task
- Mahima Abraham - Front End
  - Another front end developer is needed for the project because there are multiple aspects needed for the game to look user friendly and to display different information such as the game tab and tying the database to the game. Mahima's experience in front-end work and interest in learning more front-end skills can help in making the game more enjoyable and the development of it faster.
- Love-Divine Onwulata - Bluetooth
  - Another Bluetooth developer is needed because it is one of the major aspects of our product. It is needed to find other devices, detect the range of players, and to tag other players. Having two people assigned to this tasks will help the Bluetooth function of our app to work properly and efficiently. Love's interest in learning new technologies and his experience in Java, a language similar to Kotlin, the language being used, makes him perfect for this task.
- Greg Salisbury - Back End
  - A database needs to be developed to satisfy the objective of creating a leader board for the game, as well as a history of who got tagged by who, and so any required data can be accessed from a central location. Being able to store information during the game makes the game more enjoyable for the players, who can be more focused on the game itself, and not remembering who's winning or losing.

Greg's personal interest in back-end development and background in MySQL should make the development of the database faster and more efficient.

6 Milestones

6.1 Front End Tasks:

1. Set Up History Tab (9/13/23)
2. Create Game Tab (9/13/23)
3. Tie Database to History Tab (9/29/23)
4. Improve the layout of the application (10/2/23)
5. Add Tag Button (9/13/23)

6.2 Back End Tasks:

1. Setup Database (9/22/23)
2. Create columns for player (9/22/23)
3. Create multiple game rooms (10/16/23)

6.3 Bluetooth

1. Find Other devices using Bluetooth (9/22/23)
2. Make sure the player is in range to tag other players (9/29/23)
3. Find how to send data through Bluetooth (10/2/23)

7 3 Major Risks

1. Bluetooth - Risk of not being able to use Bluetooth to send data, detect range, and to connect players.
2. Database - Having trouble with storing the players' information in the database and then not having a leader board that players can use for rankings.
3. Software - Risk of not having enough time to develop our app into a polished piece of software due to lack of knowledge of tools being used, or slow implementation of major parts of the project.

8 External Feedback

- External feedback is most crucial after the game is developed and ready to be used by others to test the quality of the game and make sure the features are user friendly and intuitive to use.