Software Implementation and Testing Document

For

Group 29

Version 1.0

Authors:

Ryan Carmichael

Ashley Oliveira Andrade

Arlie Haire

Jason Kenyon

Lizbeth Pulles

1. Programming Languages (5 points)

Our project will be for Androids, so we will be using Java for the main language and XML for the appearance of the application.

2. Platforms, APIs, Databases, and other technologies used (5 points)

Our group is using Android Studio as our IDE, Wi-Fi Direct is one of the APIs being used.

3. Execution-based Functional Testing (10 points)

Our group is in the process of making the functionalities for our project, but once we do, we will test it on multiple devices with different users. We will get non-affiliated parties to do quality testing and find any bugs that we've missed. We have a few Android devices on hand to sideload the app onto for testing.

4. Execution-based Non-Functional Testing (10 points)

Our group is still working on the functional part of the project. After we have completed and made sure that our functionality is implemented, we will focus on the non-functional testing to see how efficient and reliable the project is.

5. Non-Execution-based Testing (10 points)

We have implemented a source control system, in which we use pull requests to allow Jason to review the code for potential errors before merging.

Progress Report

- Increment 3 -

Group 29

1) Team Members

Name	FSU ID	Github
Jason Kenyon	jk12d	fsujk12d
Ashley Oliveira	aro22b	xshlxy
Arlie Haire	ahaire	arliehaire
Ryan Carmichael	rcc22e	rctarzan
Lizbeth Pulles	lp21h	lizzietriestocode

2) Project Title and Description

Mobilympics is a gaming app that allows users to make and join P2P offline lobbies that are either publicly broadcast or locked with a password. Once in a lobby, users can play games with each other to gain rankings and medals- games will be a list of options that either the host can start for everyone or two users can play with each other within the group.

3) Accomplishments and overall project status during this increment

We have successfully developed a working Android app that allows two phones to connect and select various games to play. The app includes a lobby feature that enables users to locate and connect with each other. Both Dots and Boxes and Tic Tac Toe are fully functional, while Chess, Checkers, and Mandala are nearing completion.

4) Challenges, changes in the plan and scope of the project and things that went wrong during this increment

- **We were unable to implement all the games into the app as planned:**
    - **Dots and Boxes and Tic Tac Toe are fully functional and integrated.**
    - **Mancala and Chess have been integrated but no extensively tested**
    - **Chess was developed but has not yet been integrated into the app.**
- **The leaderboard feature was not completed within the given timeframe.**
- **These delays required adjustments to our priorities, focusing on refining the app's core functionality.**

5) Team Member Contribution for this increment

Ryan made the Dots and Boxes and implemented it into the app.

Jason created the lobby and implemented the WifiDirect functionality. Jason also implemented his Tic Tac Toe game.

Ashley completed her chess game.

Lizbeth made checkers and yelled at her laptop.

Arlie completed his mancala game.

6) Plans for the next increment

There is no next increment. Lizbeth plans to sleep. Ryan plans to drink Cheerwine and travel in Florida. Arlie plans to go to the gym and leetcode. Jason is getting engaged and preparing for grad school. Ashley is doing christmassy stuff. Goodbye Mills, you're a chill guy and we like you. Great professor!!!! Have a good break and a Merry *Christ*mas! (:

7) Stakeholder Communication

Can be found on next page:

Dear [Stakeholder Names],

I hope this message finds you well.

We have arrived at our deadline and have some good news to share! We have successfully implemented and tested two games that communicate using the WiFi Direct library to establish peer-to-peer communication without the requirement of having cell-signal or WiFi connection. The application allows two phones to Discover each other, automatically determine who is hosting, launch, play, and exit activities.

As with all good news, it must be accompanied by some bad. My team was unable to finish testing all of the games we had initially planned, and the leaderboard functionality is a work in progress. Given another month, we are confident that we'd be able to get all these features and more up and running, but alas time is not infinite. With some more testing and ideation, this app could genuinely be put out for the masses to enjoy. Perhaps over the winter break I'll get some more work done on it, as a proof of concept.

It has been an absolute pleasure working with you, and we hope that if you ever have any technology needs in the future, you'll think of Group 29 first.

Happy Holidays,

[Your Full Name]

[Your Job Title]

[Company Name]

[Contact Information]

8) Link to video

Paste here the link to your video.

https://youtu.be/ASqeh5sgNVY

Software Requirements and Design Document

For

Group 29

Version 1.0

Authors:

Ryan Carmichael

Arlie Haire

Jason Kenyon

Ashley Oliveira Andrade

Lizbeth Pulles

1. **Overview (5 points)**

Give a general overview of the system in 1-2 paragraphs (similar to the one in the project proposal).

Mobilympics uses WifiP2P to form connections between Android devices, allowing the users in a given network to invite each other to play two-player games with each other and complete those games over the local P2P network. Lobbies can be password protected or open to any Mobilympics user. Games include Chess, Checkers, Dots and Boxes, and TicTacToe.

2. **Functional Requirements (10 points) -**

1. Main menu - high
2. Lobby info dialogue - high
3. Lobby class - high
4. Leaderboard - low
5. Lobby host activity - high
6. Find lobby activity - high
7. Options activity - medium
8. Tic Tac Toe - medium
9. Checkers - medium
10. Chess - medium
11. Dots and Boxes - medium
12. Mancala - medium

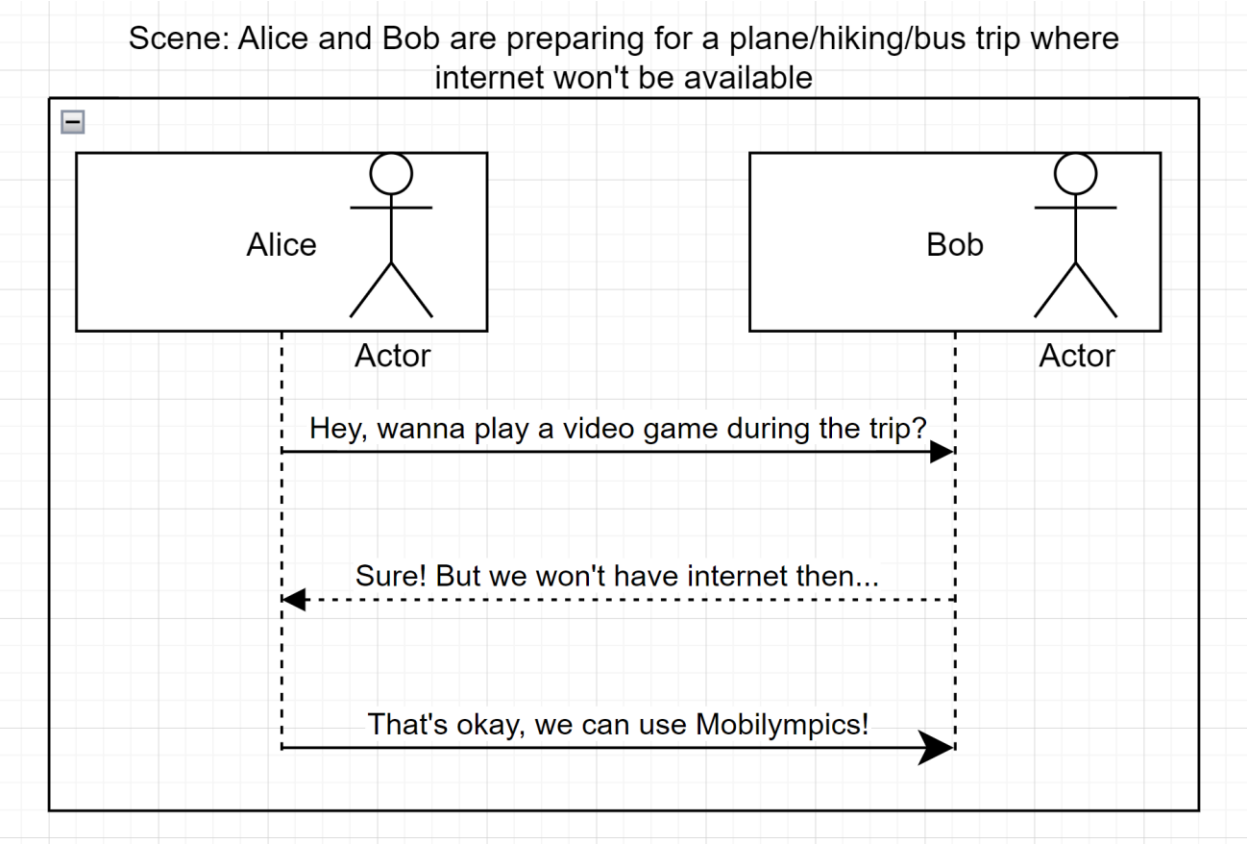
3. **Non-functional Requirements (10 points) -**

1. The app will have a good response time - medium
2. Intuitive user-friendly interface - low
3. Secure and encrypted communication - medium
4. We want it to be compatible with older Android devices - low
5. Maintainability (The code is modular and well documented) - high
6. Scalability (It should be easy to add new games) - high
7. Battery Efficiency - medium
8. Data Management (keep data if disconnected, maintain quantity of medals, etc.) - situational based

9. Optional simple animations for games like Dots and Boxes (i.e. a pen drawing) or Mancala (i.e. marbles moving).
10. Graphics for some games like the images for the chess pieces (most of which will be available on android studio)

4. Use Case Diagram (10 points) - UPDATE?

*This section presents the **use case diagram** and the **textual descriptions** of the use cases for the system under development. The use case diagram should contain all the use cases and relationships between them needed to describe the functionality to be developed. If you discover new use cases between two increments, update the diagram for your future increments.*

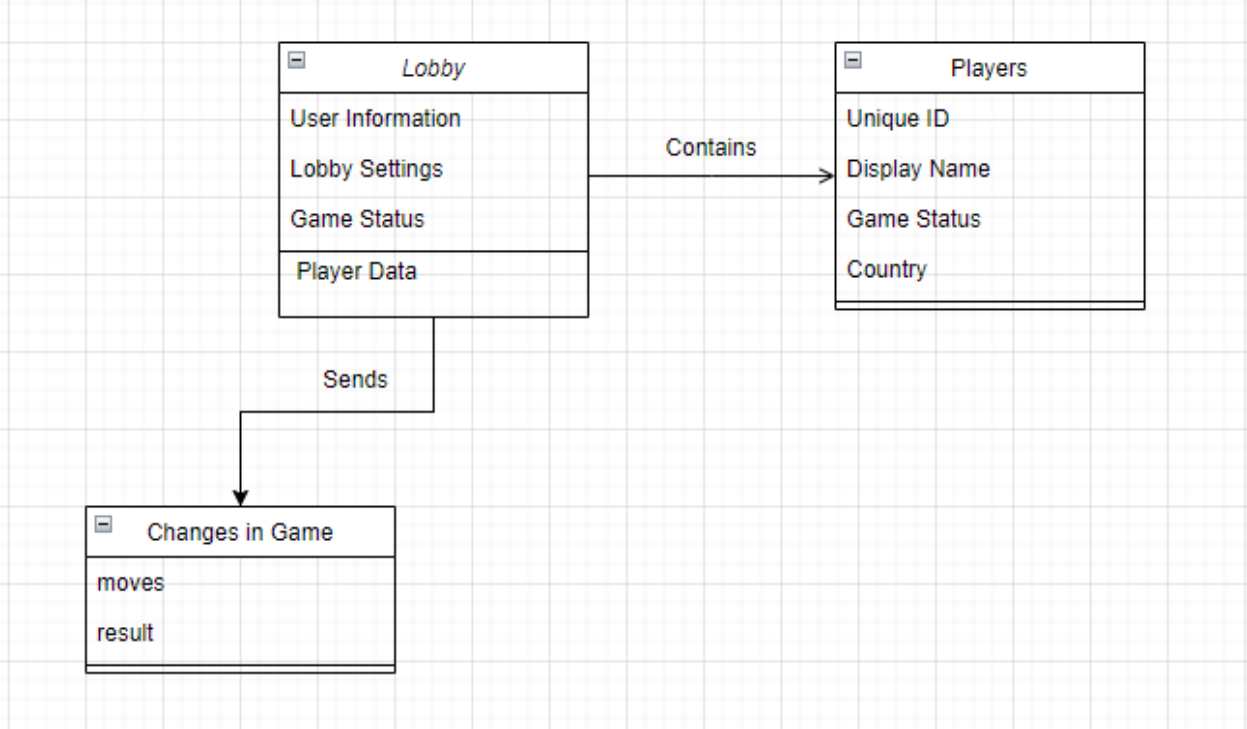


Hiking, airplane rides, power/internet outages, Idk man have you ever been bored?

5. Class Diagram and/or Sequence Diagrams (15 points) - I DONT THINK THIS NEEDS UPDATING

This section presents a high-level overview of the anticipated system architecture using a ***class diagram*** *and/or* ***sequence diagrams****.*

If the main ***paradigm*** *used in your project is* ***Object Oriented*** *(i.e., you have classes or something that acts similar to classes in your system), then draw the* ***Class Diagram of the entire system and Sequence Diagrams for the three (3) most important use cases in your system.***



If the main ***paradigm*** *in your system is* ***not*** ***Object Oriented*** *(i.e., you* ***do not*** *have classes or anything similar to classes in your system) then only draw* ***Sequence Diagrams****,* ***but for all the use cases of your system.*** *In this case, we will use a modified version of Sequence Diagrams, where instead of objects, the lifelines will represent the functions in the system involved in the action sequence.*

Class Diagrams *show the* ***fundamental objects/classes*** *that must be modeled with the system to satisfy its requirements and* ***the relationships*** *between them. Each class rectangle on the diagram* ***must also include the attributes and the methods of the class*** *(they can be refined between increments). All the* ***relationships between classes and their multiplicity*** *must be shown on the class diagram.*

*A **Sequence Diagram** simply depicts **interaction between objects** (or **functions** - in our case - for non-OOP systems) in a sequential order, i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function.*

6. Operating Environment (5 points)

The Operating Environment is Android mobile devices.

7. Assumptions and Dependencies (5 points) -

List any assumed factors (as opposed to known facts) that could affect the requirements stated in this document. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.

We are assuming that the user has an Android phone. They are bored and don't have access to an internet connection. Additionally we will utilize WiFI-Direct for peer to peer functionality in multiplayer games for local connectivity. This will allow users to connect to nearby devices without needing to connect to a hotspot or network.

We are making the assumption that there are others users nearby who also have Android and the app.