**Software Implementation and Testing Document**

**For**

**Group 29**

Version 1.0

**Authors**:

Ryan Carmichael

Arlie Haire

Jason Kenyon

Ashley Oliveira Andrade

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 2 -**

**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 a Github repo with base code and issue trackers for the base functionality requirements. Issues are being tackled in a Kanban style, with individuals selecting issues to take possession of and work on.

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

* **The original API was actually a framework for a coming API- Jason found an existing API and created a system for using it.**
* **Games are progressing slower than anticipated, so supplemental features like the leaderboard and chat have been deprioritized.**
* **Weekly in-person meetings may be helpful, as the group organizes and produces more efficiently and effectively with them.**

**5)** **Team Member Contribution for this increment**

**Ryan cried.**

**Jason created base code and core functionalities and coordinated with the group to keep us aware and mindful of deadlines.**

**Ashley kicked into gear and is getting chess/checkers upright.**

**Lizbeth is having a crisis.**

**Arlie is hard at work getting Mancala Game created.**

**6)** **Plans for the next increment**

**Get fully functional games into testing phase.**

**Implement chat.**

**Test group/lobby functionality in real-world scenarios.**

**7)** **Stakeholder Communication**

*Can be found on next page:*

Dear [Stakeholder Names],

I hope this message finds you well.

I am excited to share an important update on the progress of our offline gameplay app. We’ve successfully developed and finalized the core framework, and the first set of games are now nearing completion. With this foundational work in place, we are preparing to move into the testing phase very soon.

This is a critical milestone, as it means the app is ready for rigorous evaluation. We’re eager to ensure the games perform seamlessly across Android devices and deliver the smooth, engaging experience we’ve envisioned. As part of this next phase, we’ll be focusing on identifying and resolving any bugs, improving performance, and gathering valuable feedback to fine-tune the app before the official launch.

Key next steps:

We aim to begin testing within the next 1-2 weeks. Stakeholder feedback will be crucial during this phase. Your insights will help us refine the app to meet our quality standards, and to ensure the prototype is in line with your vision. Pending successful testing, we are on track to move forward with launch preparations.

If you have any questions or would like further details about the upcoming testing process, please don’t hesitate to reach out. We’ll be in touch soon with more information on how you can get involved in testing.

Thank you for your continued support. We’re excited to move into this next phase and look forward to working with you to make this project a success.

Best regards,

[Your Full Name]

[Your Job Title]

[Company Name]

[Contact Information]

**8)** **Link to video**

*Paste here the link to your video.*

<https://youtu.be/WqIzi6tYoDw>

**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) - UPDATE?**

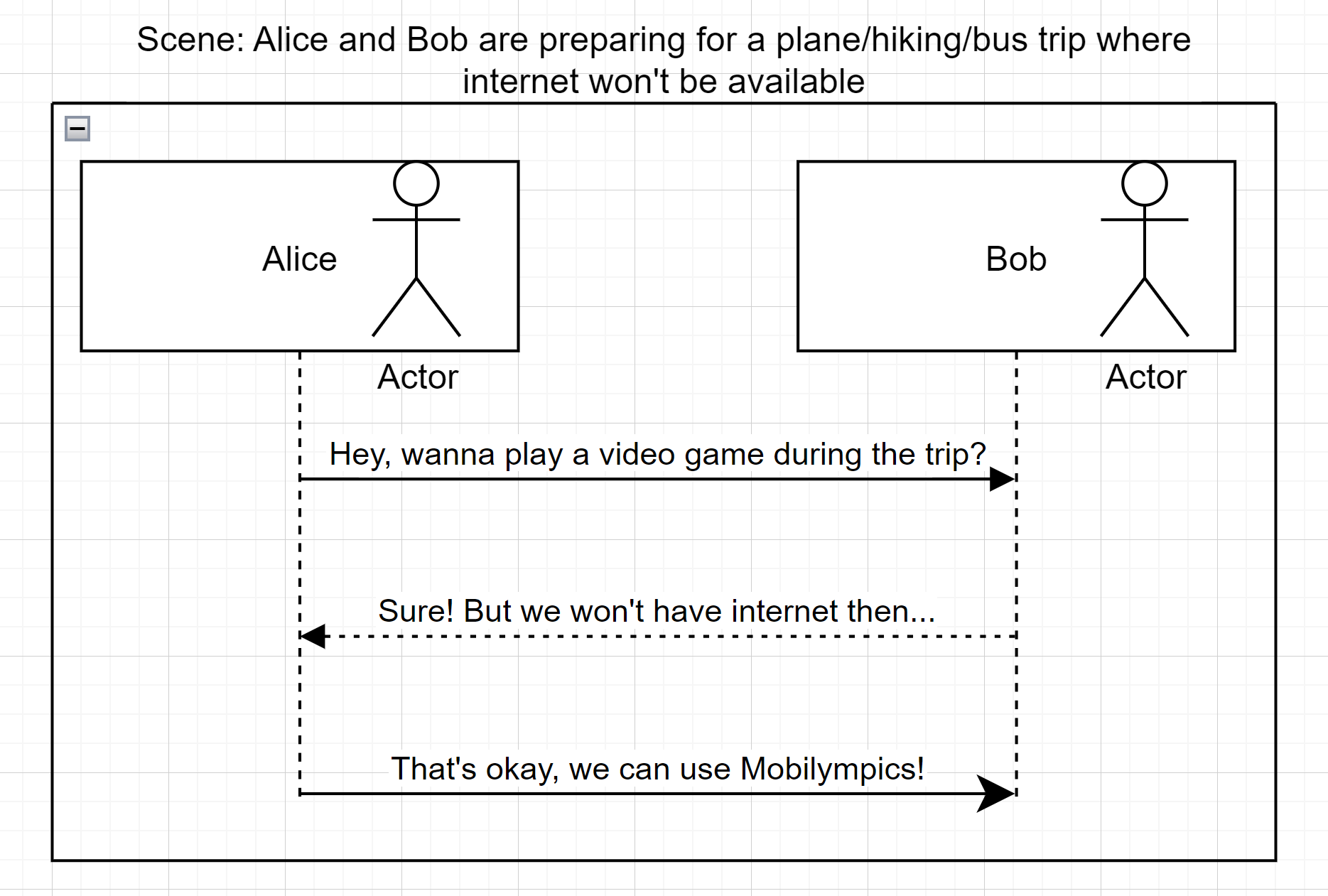
1. Main menu - high
2. Lobby info dialogue - high
3. Lobby class - high
4. Leaderboard - low
5. Lobby host activity - high
6. Find lobby activity - high
7. Options activity - medium
8. Tic Tac Toe - medium
9. Checkers - medium
10. Chess - medium
11. Dots and Boxes - medium
12. Mancala - medium

# **3.** **Non-functional Requirements (10 points) - UPDATE?**

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.*

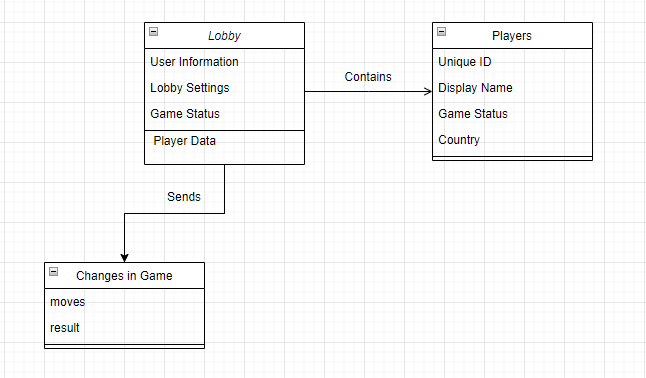


Hiking, airplane rides, power/internet outages, Idk man have you ever been bored?

# **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) - THOUGHTS ON UDPATE?**

*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 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.