

G-Linux - Online Game Store

Faculty of Information Technology - MUST University
Fall 2021/2022

Team Members (Team 117)

89553 - Nour Gaser Algendi
89555 - Wafaa Wael
89671 - Ahmed Khaled Mohamed Ismael Kira
89634 - Barakat Ramadan Barakat
77697 - Ammar Elbanna



Under the supervision of Dr. Laila Abd EL-Latif



1. Introduction	2
1.1 Overview	2
1.2 Document Conventions and Language	2
1.3 Goals	3
1.4 Project Purpose	3
1.5 Project Scope	3
1.6 References	3
2. Overall Description	4
2.1 Requirements Gathering Methods	4
2.1.1 Requirements Gathering Methods' Main Components	4
2.1.2 Method 1 - Brainstorming	4
2.1.3 Method 2 - Market Studies	5
2.1.4 Method 3 - Questionnaires	6
2.2 System Requirements	10
3. Models and Diagrams	12
3.1 Context Diagram	13
3.2 DFD Diagram	14
3.2.1 DFD L1	14
3.2.2 DFD L2	15
4. System Design	16
4.1 Database Design (ERD)	16
4.2 Design Interface	17
4.2.1 Game Developer Interface	17
4.2.2 User Interface	20
5. Appendix	23
5.1 Questionnaire Responses	23
5.2 Collaboration and Planning Diagrams	24
5.2.1 Project's Phase II Plan	24
5.2.2 Project's Software Requirements Plan	25

1. Introduction

Linux is becoming more and more mainstream nowadays, and given that a large portion of PC owners play video games, it's only logical that Linux should make it easy for its users to purchase, install, run, and play video games. Moreover, it should facilitate tools for game developers to develop and publish Linux-native versions of their games. There's a lack of game publishers, game stores, and investment in the gaming scene from software developers, despite the increase in demand; G-Linux is here to solve that.

1.1 Overview

G-Linux is an open-source online game store where people go to find and enjoy their favorite video games on their Linux devices. G-Linux will be available through all the Linux package managers, and will provide the necessary tools for game developers from all over the world to publish and distribute their video games.

1.2 Document Conventions and Language

Here's a list of word definitions for words which will be frequently used throughout this document:

- Linux: the operating system.
- Compatibility software, Wine: *Wine*¹ is a compatibility tool for Linux, it's used to run software which supposedly only runs on Windows by translating system calls into equivalent POSIX calls used by Linux; in simpler words, though the performance of apps run through Wine is worse than real native apps, Wine is not an emulator, it converts internal Windows commands to commands your Linux system can understand and execute.
- Dependency list: on Linux, just like on any OS, when installing a piece of software, most of the time it needs other software to be installed in order for it to function (drivers, compatibility software, specific tools or runtime environments, etc..). This is what we typically mean by a dependency list.
- Distro: a Linux distribution (Ubuntu, Mint, Fedora, Arch, etc..)

¹ <https://www.winehq.org/>

1.3 Goals

1. Create a platform where Linux users can purchase, group, and launch their favorite video games.
2. Make the process of publishing video games on Linux easier for game developers.
3. Take the initiative and start encouraging game developers and video-games-related software developers to develop native versions of their software for the Linux platform; ultimately growing the Linux user base and empowering its video games capabilities.

1.4 Project Purpose

G-Linux's vision is a completely satisfying, fully-functional Linux gaming experience. How it contributes to that will be through ***analysis*** of the current problems that face Linux users when purchasing and launching video games, then through a ***design*** for a system which attempts to deploy a maintainable solution to these problems. G-Linux's purpose is beyond its very own software, it aims to be a pioneer amongst the free and open-source communities, taking the initiative to make Linux gaming more accessible to all Linux users, encouraging game developers to develop and build native version of their games for Linux, and perhaps eventually growing Linux's user base through users which previously didn't use it due to its shortcomings when it comes to video games.

1.5 Project Scope

This project embraces Linux's open-source nature to create a new game store/platform where Linux users can purchase, install, and manage their video games and game launchers through a user-friendly interface which hides all the compatibility layers underneath. Furthermore, it encourages that compatibility solutions be temporary and that software developers distribute native versions of their software for the Linux platform, which is why this project, its documentation (including this document), its codebase, its database of software dependencies, and all its non-sensitive data will be open-source, contributable to by the community, and we encourage anyone to fork and improve over our efforts to work towards our common vision (mentioned in section 1.3).

1.6 References

- This document's structure is loosely based on the *IEEE Guide to Software Requirements Specifications*².
- The statistical sample used in the questionnaires (refer to section 2.1.4) consists mostly from Linux users which were found on Reddit³.

² Stellman, Andrew & Greene, Jennifer (2005). *Applied software project management*. O'Reilly Media, Inc. p. 110

³ https://www.reddit.com/r/linux_gaming/comments/r391kl/university_project_help/

2. Overall Description

This section of the document should serve as the overall description of the project. That includes the system's description, its requirements specification, and how information was collected to formulate said requirements. <This text should be updated after this chapter is complete.>

2.1 Requirements Gathering Methods

2.1.1 Requirements Gathering Methods' Main Components

Each method we use will be defined through **three components**:

- The **goal** of this particular method; what it aims to achieve.
- The **process** -- how this method was carried out.
- The **results and the conclusion**.

The methods we will use are three: brainstorming, market studies, and questionnaires to the system's target users (game developers and gamers).

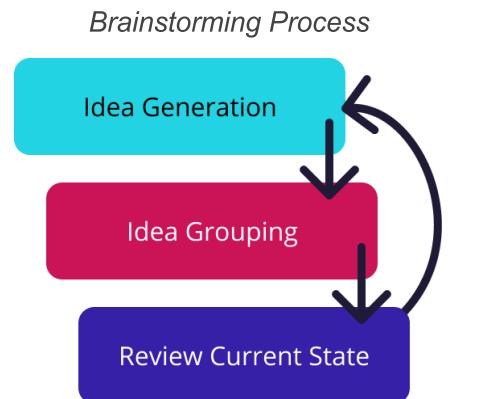
The appendix will include the outputs of methods which produce results such as filled out forms, etc..

2.1.2 Method 1 - Brainstorming

- A. Goal:** brainstorming sessions should contribute to the conceptualization of our product, defining its core functional requirements, and bringing the whole project team up to speed on what this project is about and their roles in the upcoming phases of the project; the result should be an initial idea of what our system should do, and who its users are.

- B. Process:** consists of three stages:

- Idea generation: this happens during meeting sessions where the whole project team is present and free to present their ideas and discuss them, the team leads making notes throughout the session for later analysis.
- Idea grouping: the team leads then group the ideas into their respective categories (user requirements, system constraints, etc..)



- 
- Reviewing current state: the team leads review what our initial system's concept looks like so far and decide whether further brainstorming sessions are needed or not.

C. Results and conclusion: ours is an open-source game platform native to Linux, G-Linux should aim to make installing games and launching them easier for all Linux users of all distros, all the while grouping the user's favorite games and game launchers in one library, where games are easily launchable regardless of compatibility issues, making use of a comprehensive database of software dependencies (for temporary compatibility solutions, like *Wine*, for non-Linux-native games), which we collect from our users given their consent and guaranteed anonymity. It should serve as a platform where Linux gamers can game, chat, hangout, and share their experiences together. Furthermore, G-Linux should make the process of publishing Linux-native video games on its store easy and non-infuriating for game developers; it should provide them with the necessary tools to maintain their already published video games, allowing them to track their sales, highlight their games on the store's homepage, and deliver updates and DLCs.

The next section will explore these features more by looking at current software in the market with similar features, what they do well, and what can be improved.

2.1.3 Method 2 - Market Studies

A. Goal: the goal of this information gathering method is to study the current gaming scene on Linux and study the pros and cons of the available gaming options. Conclusions made will be either validated or invalidated using the questionnaires from the next section, then the information will be used to strictly define the software's functional requirements.

B. Process: a quick research will tell you that the largest PC game stores/game launchers available are ***Steam***, ***Epic Games Store***, ***Riot Games***, and ***Battle.net***. In addition to these, there's a software called ***Lutris*** which is an open-source platform specific to Linux. Our process will be to list the pros and cons of each of these, to reach a conclusion on what features are missing from the market and which are fulfilled (and to which degree).

- Lutris: this is the most famous gaming platform for Linux, and it's the closest thing to G-Linux available; Lutris is an open-source platform which helps you download and install games (or game launchers) through their massive dependency list. All you have to do is search for the video game title you want, and if it's available, you can click one button and Lutris will do the rest, that's how it's supposed to be at least. Sadly, Lutris' UI is unattractive, unintuitive, complex, and often involves a lot of tinkering and configuration to get games running, which puts it well beyond the hands of the average gamer's computer skills. Lutris also lacks basic features like chatting and general quality-of-life

gaming features, which makes for a gaming experience where you have many, many different apps running to accomplish tasks which should optimally be in one place. Note: unlike the rest of this list, Lutris is **not** a game store, in fact, it helps the user to install game stores which aren't available on Linux through compatibility solutions, namely Wine; Lutris merely semi-automates configuring Wine.

- Steam: by far, Steam and *Valve* - its owner company - are one of the only companies working towards compatibility solutions for Linux (see *ProtonDB*⁴), and actively develop native games for Linux. The biggest con for Steam, is that their library of games mostly consists of Valve's own video games, many small indie games, and only occasional big titles; most of which being not exclusive to Steam. However, Steam does offer solid community tools like forums, game ratings, chatting, status updates, etc..
- Epic Games Store: not available for Linux along with all of its video games; must be installed through Wine (with or without the help of Lutris).
- Riot Games: not available for Linux along with all of its video games; must be installed through Wine (with or without the help of Lutris).
- Battle.net: not available for Linux along with all of its video games; must be installed through Wine (with or without the help of Lutris).

C. Results and conclusion: most of the largest game stores aren't even available for Linux, with Steam being the only large store available, and most of its games being run through several performance-killing compatibility layers; Linux users have to resort to Lutris (or Wine directly) to run most of their games, having to settle for a more humble gaming experience with less features, unsettling UI, worse performance than Windows, and an overall lesser gaming experience.

G-Linux should aim to incorporate Lutris' core functionalities into a more elegant, user-friendly UI, inclusive of much-needed community features like game reviews, chatting, friends lists, status updates, livestreaming, etc.. G-Linux should also build a new comprehensive dependency list for video games which do not run natively on Linux, in order to provide an experience where users can run all their games from the same place, making use of other users' troubleshooting experiences.

The next section should validate or invalidate these features, as well as append to them through Linux gamers' feedback based on their current experiences.

2.1.4 Method 3 - Questionnaires

A. Goal: to validate or invalidate the features list provided from the previous information gathering methods, based on user feedback. This is the last information gathering

⁴ <https://www.protondb.com/>

technique which will be used; the results of this questionnaire should be information accurate enough to formulate a comprehensive list of software requirements.

B. Process: two questionnaire forms are created: one for the normal **gamer Linux users**, and one for **game developers**. The questionnaires' questions fall into one of these three categories:

- **Target validation:** these consist of multiple screening questions which make sure that whoever's filling the questionnaire is our desired target. These questions may be repeated multiple times in order to filter out spam responses through contradictory answers.

Example: "*are you a game developer?*"

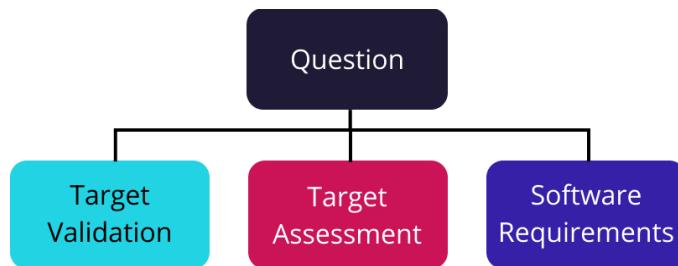
- **Target assessment:**

these aim to assess our level of interest in this target; how much experience do they have and how valuable their feedback can be.

Example: "*how many years have you been a game developer?*"

- **Software requirements:** questions related to G-Linux and its features; answers to these questions are what we are looking for and analysis will be done on these answers.

Example: "*Which of these features do you think are currently unsatisfied in the market?*"



In the interest of keeping this document short and organized you can view the two questionnaire forms *here*⁵, and *here*⁶.

⁵ User questionnaire: <https://docs.google.com/forms/d/1pdBvRUZYGu-3WIwwTHgX6rrgTNHPTYGna6gXM0IkPwk>

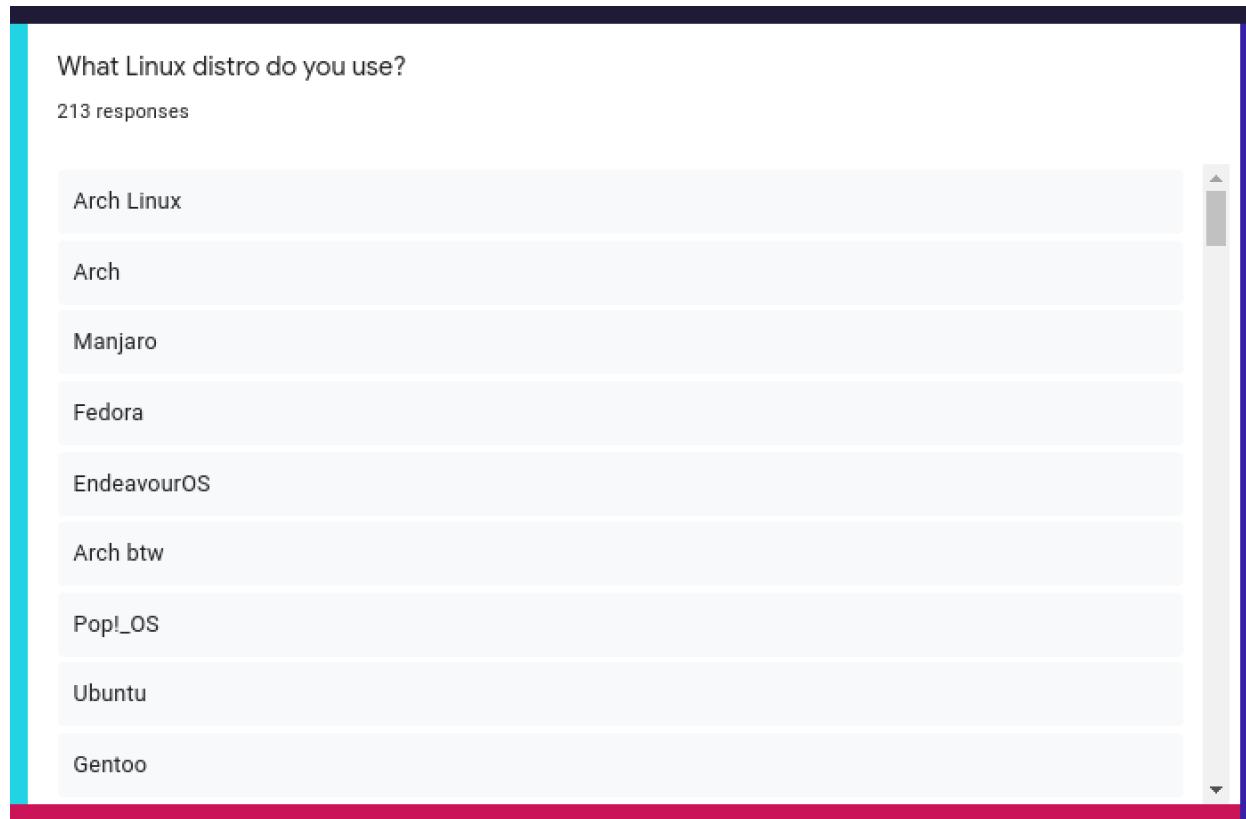
⁶ Game developer questionnaire:

https://docs.google.com/forms/d/1CjCeA_OjEV4sjk1XKShtJCAGcBYAr6dSJVP7A_Rs64

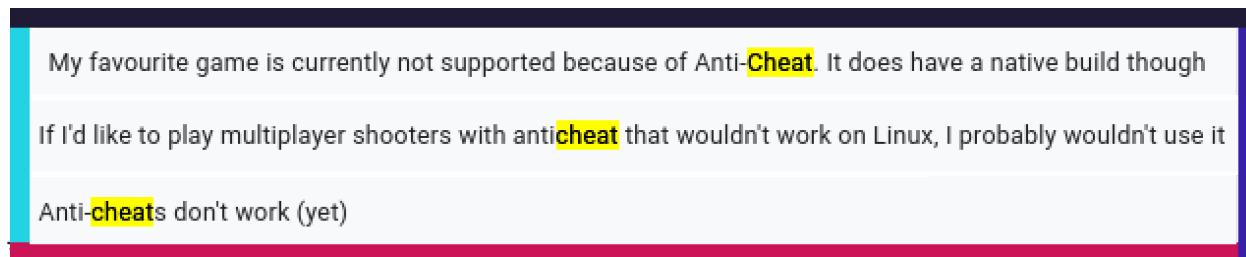
Results and conclusion: in the interest of keeping this document short and organized, you can view the questionnaire's responses statistics *here*⁷, and *here*⁸.

The responses to the questionnaires more or less validate all the features that had been listed from the previous information gathering methods, however key new issues can be highlighted, as well as features that users feel are missing from their current experiences:

- Linux gamers use all kinds of different distros; trying to reproduce steps of installing a certain game might cause unexpected behaviours.



- Anti-Cheat softwares used by many games don't currently work on Linux.



be concluded from looking at the statistics.

- Microsoft's GamePass only works on Windows.

⁷ User: <https://docs.google.com/forms/d/1pdBvRUZYGu-3WIwwTHqX6rrgTNHPTYGna6gXM0IkPwk/viewanalytics>

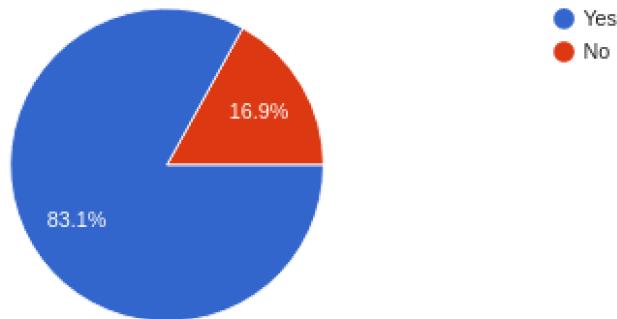
⁸ Developer: https://docs.google.com/forms/d/1CjCeAOjEV4sjk1XKShtjCAGcBYAr6dSJvp7A_Rs64/viewanalytics

- There are performance issues compared to gaming on Windows. (This is because of compatibility software).
- *Sandboxing*⁹ features are rare/non-existent (to stop game launchers from collecting user data).
- Lutris does not provide game-specific troubleshooting steps for getting a game to work.
- Game developers find it to be too much effort to create *portable binaries*¹⁰.
- Game developers find that it does not pay off creating Linux-native versions of their games because of the small market share.
- Many more conclusions can be made from further analysing the 200+ responses of the questionnaires.

The questionnaires also conclude that a large percentage of users would be willing to consent to sharing their system's information for use in our dependency database; we can use that information to predict how easily a user can install a specific game on their system by comparing their system information to those of previous users with similar configuration who successfully installed the same game. We can also process the data to automatically generate troubleshooting steps, and perhaps even automate these steps in order to get as many games running without the user needing to interfere or configure anything.

Would you be willing to share your system's information when using the software? (It'll be used to provide a better experience for other users of similar system configurations)

177 responses



We can conclude that G-Linux should aim to implement all the features from the previous information gathering methods, as well as fix as many issues as possible that face the Linux gaming community as seen in the questionnaires' analysis.

⁹ Sandboxing: isolating the software from its environment so it cannot collect data about the system or the user.

¹⁰ Portable binaires: executables which would run on most/all operating systems.



2.2 System Requirements

We can deduce most of our system's requirements from the 3 requirements gathering methods used in the previous sections:

Requirement ID	Requirement Statement	Non/Functional	Comments
REQ01	The system allows users to browse, search, and view available video games.	Functional	Users don't have to be logged in. All video games have dedicated pages.
REQ02	Users can register new accounts into the system.	Functional	Available on Linux only.
REQ03	When viewing a video game's page, the system shows an overall percentage likelihood that a certain game will work properly if installed on Linux.	Non-functional	
REQ04	Users can consent to sharing their system's information with us.	Functional	System information will be stored in our database for video game compatibility analysis.
REQ05	Users can purchase video games.	Functional	Must be logged in.
REQ06	Users must pay when adding games that are published directly on the system to their libraries.	Functional	
REQ07	Users can install free video game launchers.	Functional	Like Battle.net, Epic Games Store, etc.
REQ08	Users can view their library section which contains all their video games.	Functional	
REQ09	Users can search for	Functional	



	each, add as friends, and chat with each other.		
REQ10	The system must provide troubleshooting information for all available video games.	Functional	Can be collected from users.
REQ11	The system must display technical game compatibility issues.	Non-functional	Known graphical issues, anti-cheat support, etc.
REQ12	Game developers must register on the system before publishing games.	Functional	Same goes for game studios.
REQ13	Game developers can publish their games on the system.	Functional	
REQ14	Game developers must register their games into the system first for first-time publishes (full information like price, genre, images, size, minimum-requirements, and description).	Functional	
REQ15	Comprehensive sales and performance reports shall be provided for game developers.	Functional	
REQ16	Game developers' games must be natively built for Linux.	Functional	

***It is clear that G-Linux is a software that can only be developed incrementally through agile methods; this requirements list is likely to go through many iterations, updates, and improvements until a comprehensive list is acquired.**



3. Models and Diagrams

This section will contain some useful UML diagrams, specifically, the system's context diagram, and its DFD diagram. <this section should be updated when this chapter is complete.>

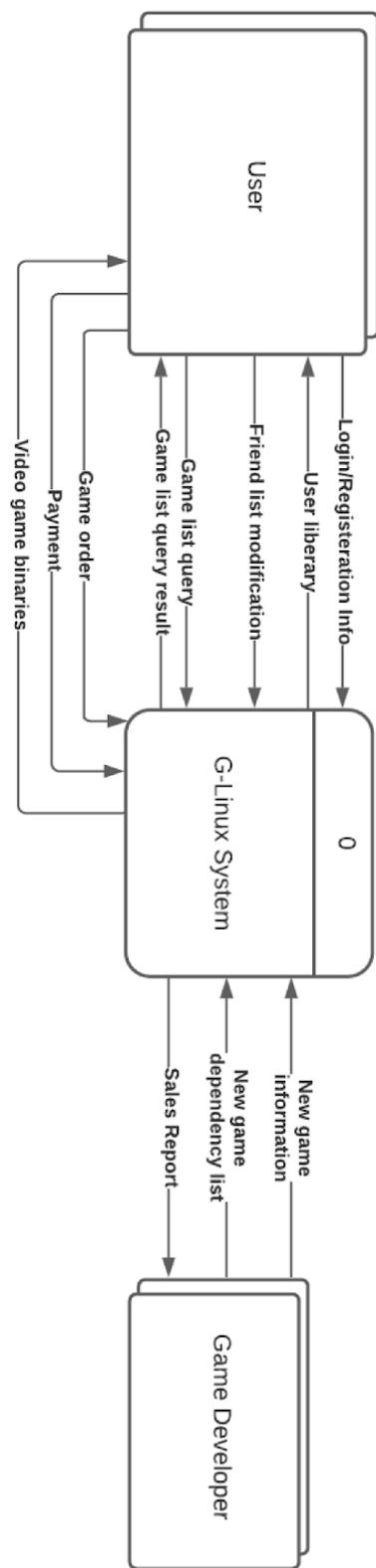
you can view these diagrams digitally [here](#)¹¹

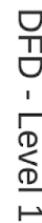
this page left empty intentionally

¹¹ <https://linky.design/phase2-dfd>

3.1 Context Diagram

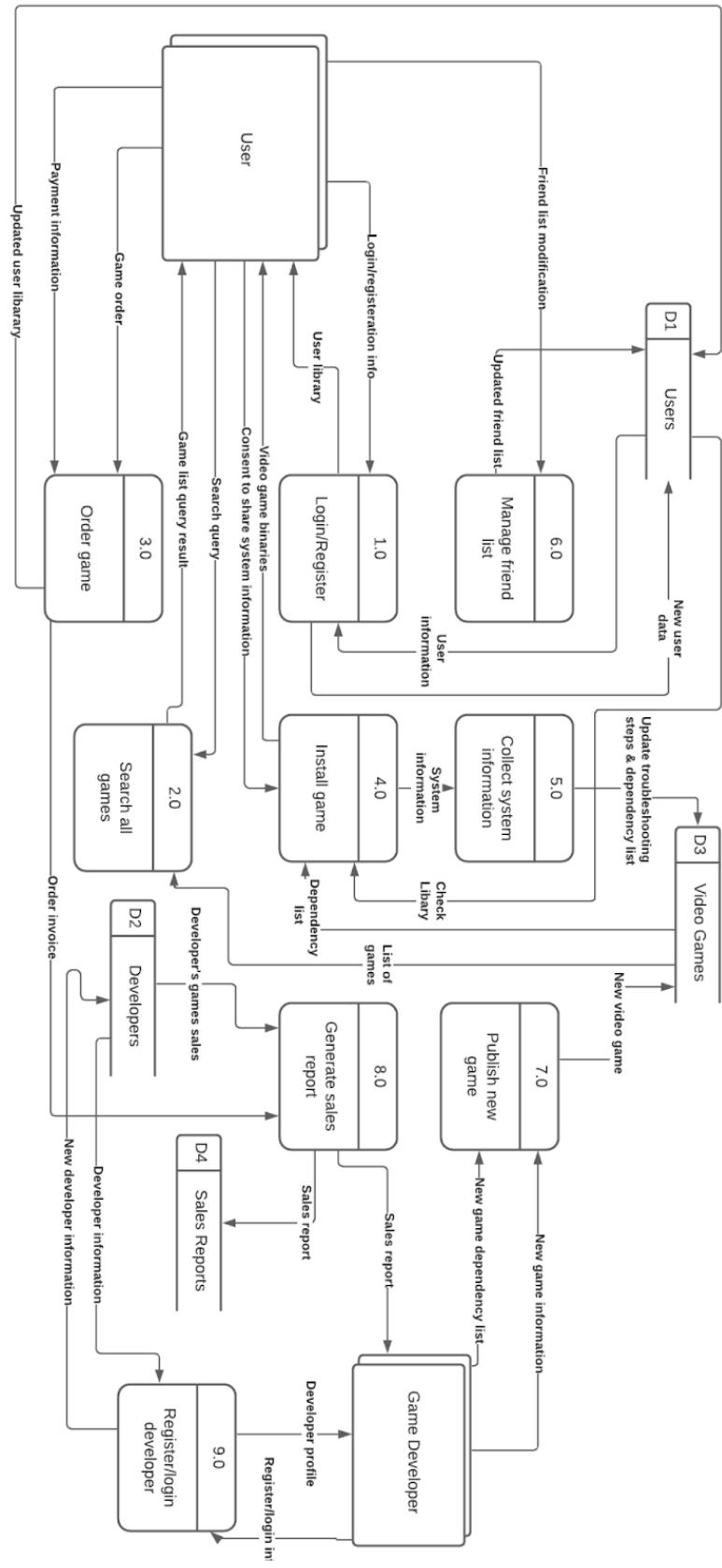
Context Diagram





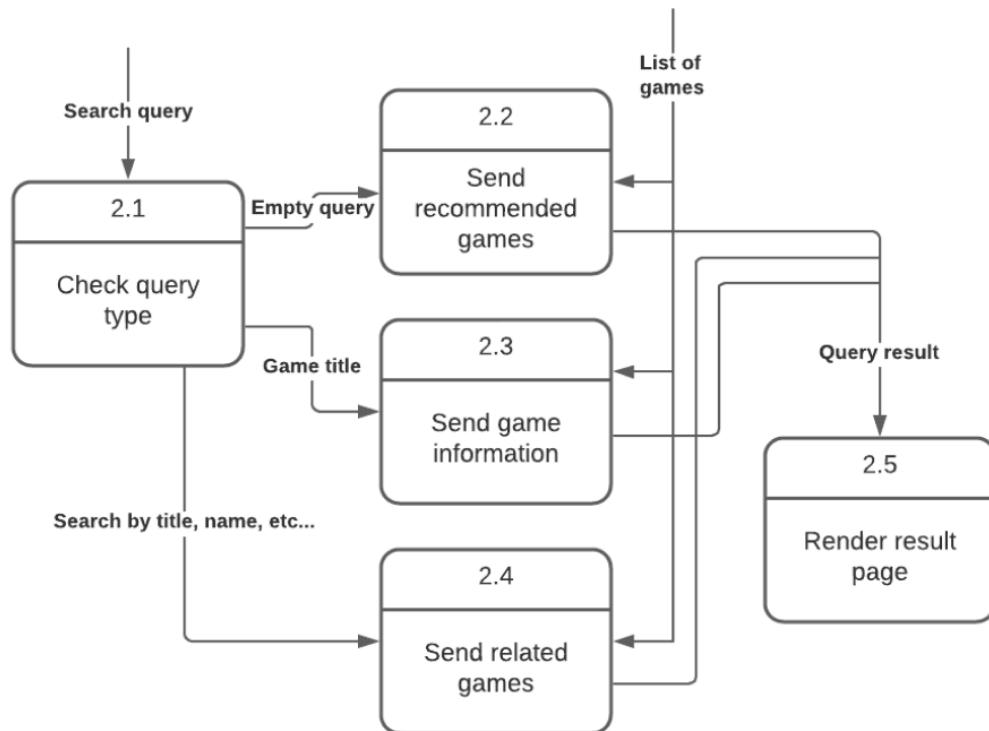
3.2 DFD Diagram

3.2.1 DFD L1

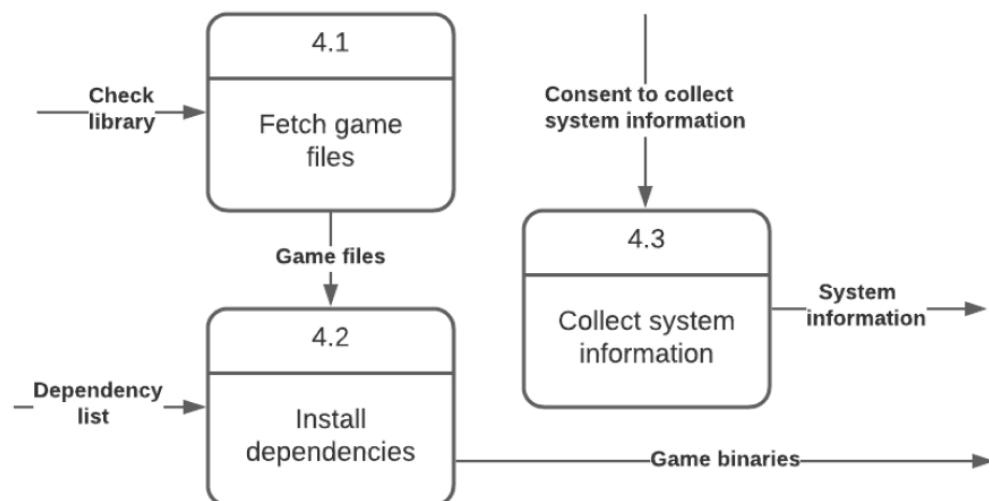


3.2.2 DFD L2

DFD - Level 2 (Process 2.0)



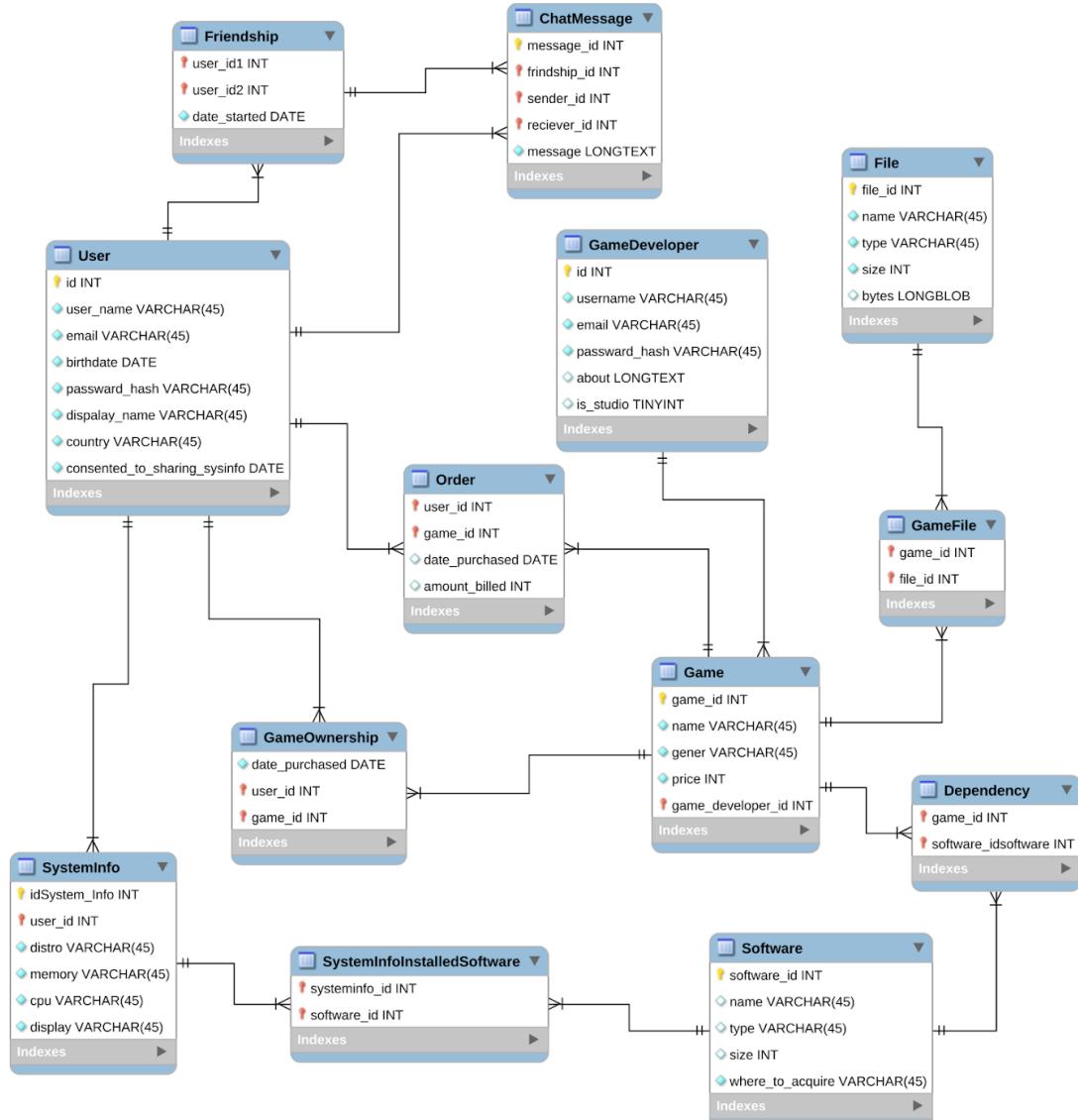
DFD - Level 2 (Process 4.0)



4. System Design

This section will be used to discuss the system's database and interface design.

4.1 Database Design (ERD)





4.2 Design Interface

4.2.1 Game Developer Interface

G-Linux
Developers

New Developer

First Name Last Name

E-Mail Address Country

Username Display Name

Password Confirm Password

Birthdate • You must be at least 16 years old.

Indie Dev. or Game Studio? Indie Game Studio

About

I agree to G-Linux's [terms of service](#).



G-Linux Developers

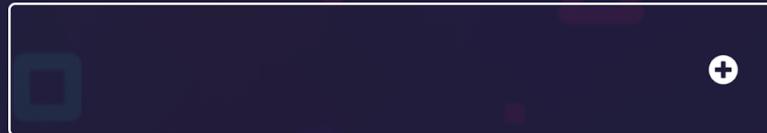
Register New Game

Title

Price

 Genre

Media



About

Linux Native? Yes NoI have read G-Linux's [rules and guidelines](#) of publishing video games.

Back G-Linux Developers Register New Game Upload Game Files

Click or drag here

G-Linux
Developers

Register New Game

Title

Price

Genre

Media



Preview 



About

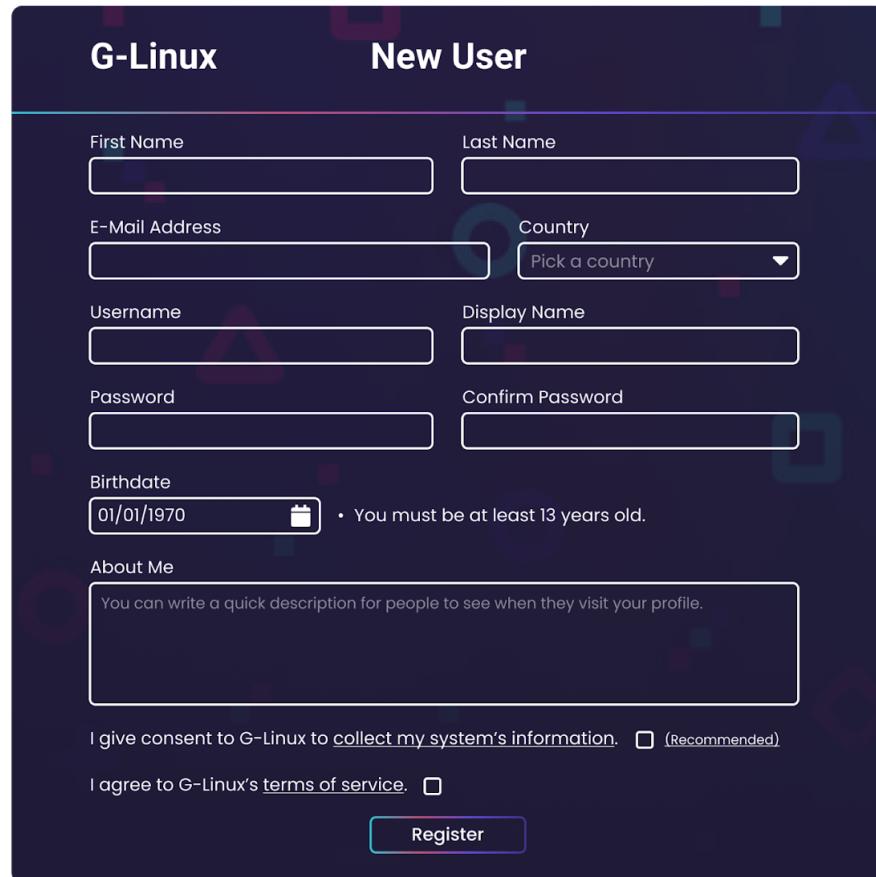
Fall Guys: Ultimate Knockout flings hordes of contestants together online in a wild dash through round after round of escalating chaos until one victor remains! Battle bizarre obstacles, shove through unruly competitors, and overcome the unbending laws of physics as you stumble towards greatness. Leave your dignity at the door and prepare for hilarious failure in your quest to claim the crown!

Linux Native? Yes No

I have read G-Linux's [rules and guidelines](#) of publishing video games.

Submit for Review

4.2.2 User Interface



The screenshot shows the 'New User' registration form for G-Linux. The form is titled 'New User' and includes fields for First Name, Last Name, E-Mail Address, Username, Display Name, Password, Confirm Password, Birthdate (set to 01/01/1970), and a note about being at least 13 years old. There is also a 'About Me' text area and two checkboxes for consent and terms of service, followed by a 'Register' button.

G-Linux **New User**

First Name Last Name

E-Mail Address Country

Username Display Name

Password Confirm Password

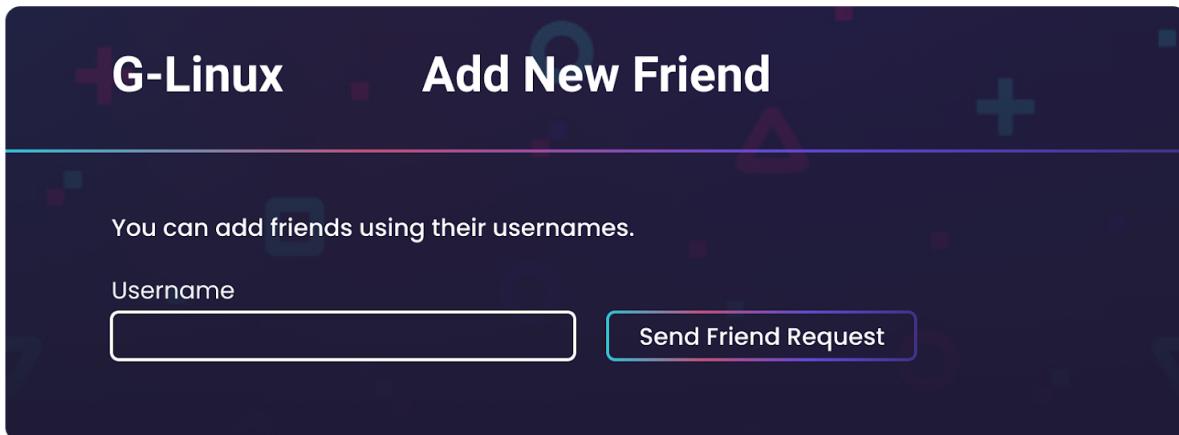
Birthdate • You must be at least 13 years old.

About Me
You can write a quick description for people to see when they visit your profile.

I give consent to G-Linux to collect my system's information. (Recommended)

I agree to G-Linux's [terms of service](#).

Register



The screenshot shows the 'Add New Friend' form for G-Linux. It features a note about adding friends using usernames, a 'Username' input field, and a 'Send Friend Request' button.

G-Linux **Add New Friend**

You can add friends using their usernames.

Username **Send Friend Request**



G-Linux

New Order

Title

Fall Guys: Ultimate Knockout

Price

9.99\$

Genre

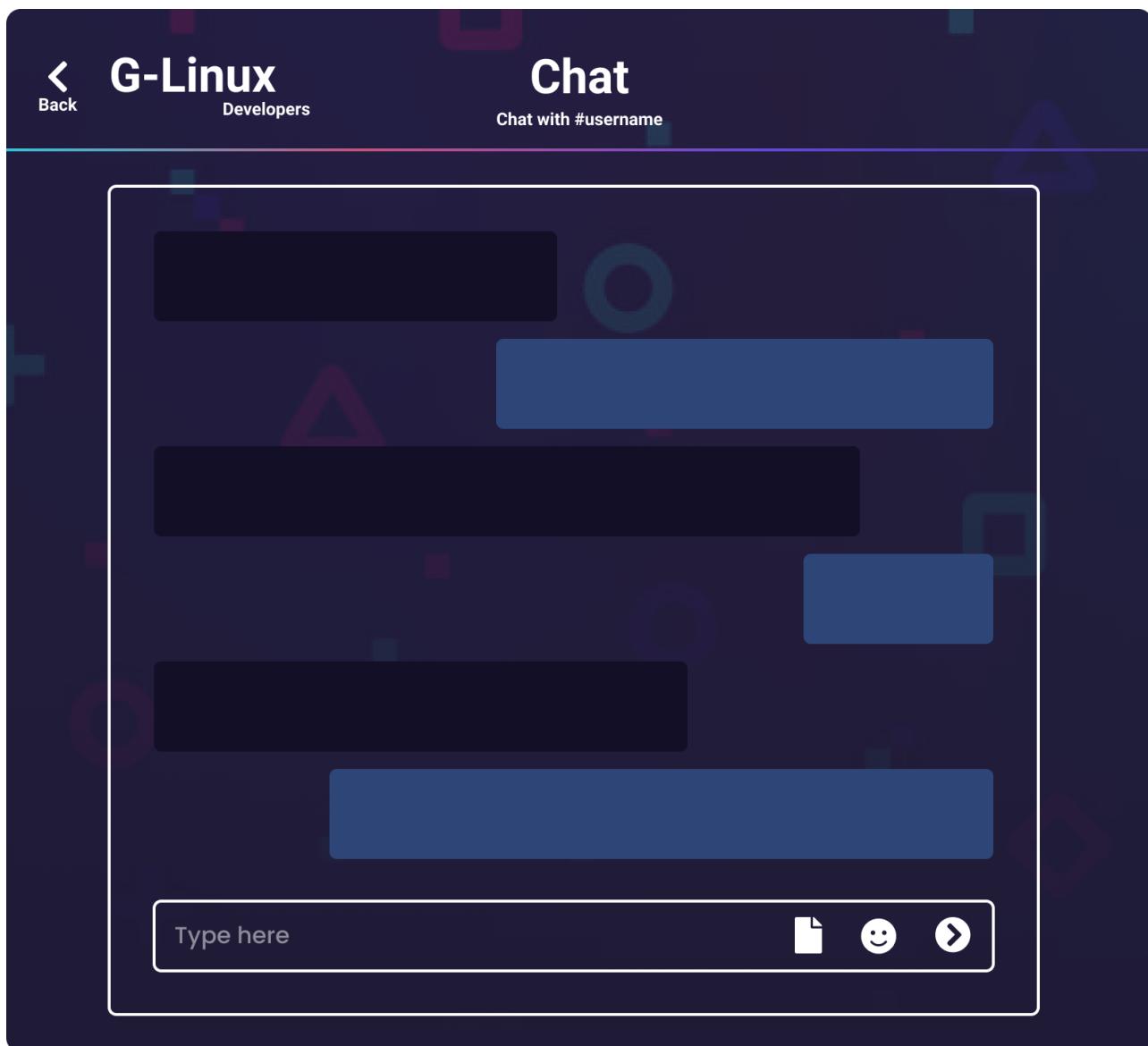
Arcade, Multiplayer

Media

▶ Preview

>

Checkout >





5. Appendix

This section will be used to contain miscellaneous topics which we couldn't fit into any other chapter or subchapter.

5.1 Questionnaire Responses

The responses to the questionnaires turned out to be 250+ responses; in the interest of keeping this document organized, please refer to [these documents](#)¹² to view all of the individual responses.

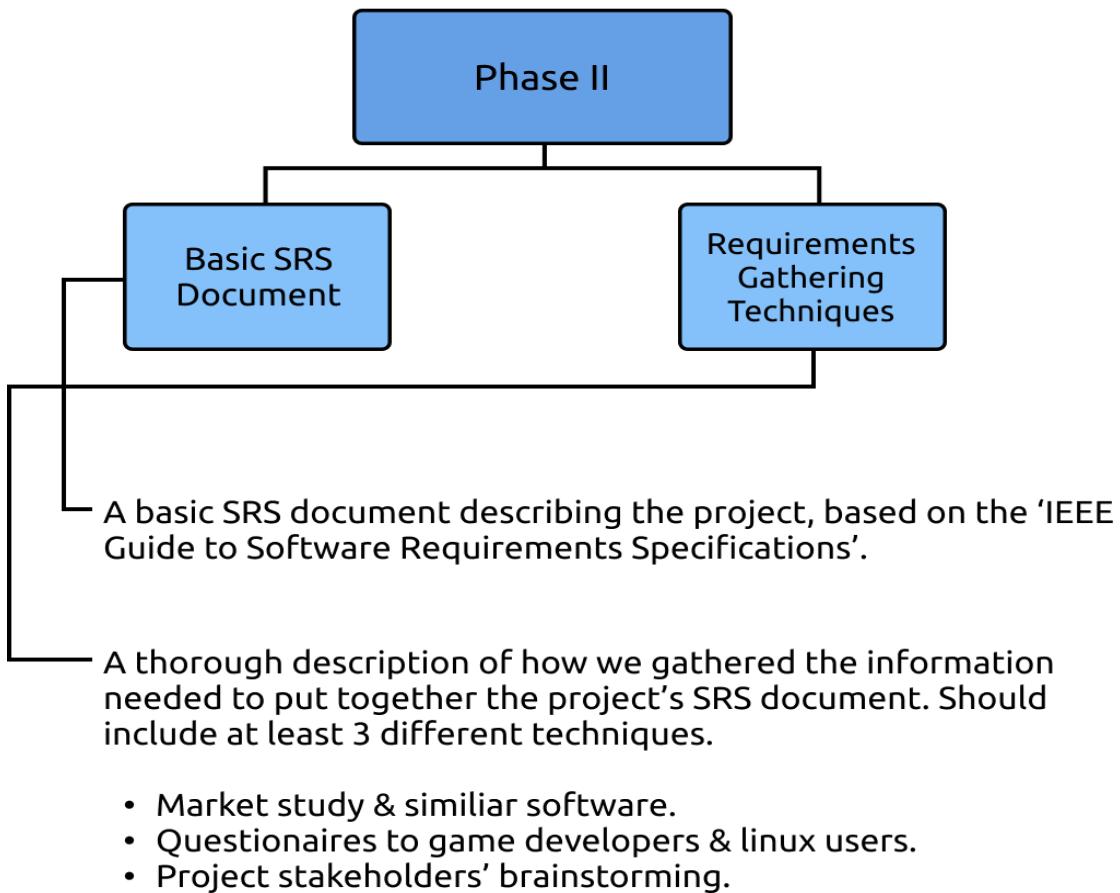
¹² Link: <https://drive.google.com/drive/u/1/folders/1PZMtSPpkA8T4aRp-vPFDjIRwkT2gxYxZ>



5.2 Collaboration and Planning Diagrams

5.2.1 Project's Phase II Plan

Systems Analysis & Design Project | G-Linux Phase II: SRS & Requirements Gathering



5.2.2 Project's Software Requirements Plan

