

Princess Sumaya University for Technology

King Hussein School for Computing Sciences

**?  
Social Media Platform**

**Prepared By:**

Mutlaq Alsadeed  
Yazan Farrah

**Supervised By:**

Dr. Ammar Odeh

Project Submitted in partial fulfillment for the degree of Bachelor of Science in Computer Science

Summer-2022

Declaration of Originality

This document has been written entirely by the undersigned team members of the project. The source of every quoted text is clearly cited and there is no ambiguity in where the quoted text begins and ends. The source of any illustration, image or table that is not the work of the team members is also clearly cited. We are aware that using non-original text or material or paraphrasing or modifying it without proper citation is a violation of the university’s regulations and is subject to legal actions.

Names and Signatures of team members:

Mutlaq Alsadeed

Yazan Farrah

Acknowledgments

We would like to thank Dr. Ammar Odeh for being the most creative, supportive, and most intellectual real teacher we have ever had.

The idea itself was originated from Mutlaq Alsadeed and was discussed with Yazan Farrah to enhance the idea, otherwise no one helped us.

Summary

For us this is not a simple graduation project, for us this is a real business opportunity that we are going to pursue.

As anime fans ourselves we know firsthand how this idea can benefit the fanbase community and the content creating community both finically and socially.

Our motivation is finical and belief in the idea because of the excitement it gives to all the anime fans, on the other hand one of the main reasons we are perusing this project is that we did extensive research on every anime related application and we found out that no one came close to making this idea.

Our mythology lies in finding the mistakes in popular applications and correcting them in our application, and providing the precis and needed features in such applications and using them to our advantage in our application.

List of Abbreviations (incomplete)

List the abbreviations you have used in your project if there are any and what they stand for.

**UML**: Unified Modeling Language.

**AMV**: Anime Music Video.

**Mad House:** Anime making studio based in Japan-Tokyo.

**Ufotable:** Anime making studio based in Japan-Suginami City.

**Kyoto Animation:** Anime making studio based in Japan-Uji.

Table of Contents

The table of contents should be automatically generated by going to: *Insert >> Index And Tables>> Table of Contents.* Choose *Classic* as the format of the table and set the number of levels to be 3.

In order for the table of contents to be generated correctly:

* Each chapter title should be formatted using the style “Chapter”.
* Each section title should be formatted using the style “Section”.
* Each subsection title should be formatted using the style “Subsection”.

The table of contents should replace all of the text that is in this page.

Table of Figures

The table of Figures should be automatically generated by going to: *Insert >> Index And Tables>> Table of Figures.* Choose *Classic* as the format of the table and set the tab leader to dots.

In order for the table of figures to be generated correctly, the label of each figure should be formatted using the style “Figure”.

The table of figures should replace all of the text that is in this page.

**Table of Tables**

The table of tables should be automatically generated by going to: *Insert >> Index And Tables>> Table of Figures.* Choose *Table* as the caption, *Classic* as the format of the table and set the tab leader to dots.

In order for the table of contents to be generated correctly, the label of each table should be formatted using the style “Table”.

The table of tables should replace all of the text that is in this page.

Chapter 1  
Introduction

* 1. Overview

We’re going to explain the basic approach for this project in a list of reasons.

* + 1. New Era

Especially for this project we took our skills further and exceeded our limitations and expectations, so we specifically used technologies that are more advanced than the common technologies used to be in university graduation projects, the whole point of this approach is to relief ourselves for the long run; especially because this graduation project will be turned into a real business in the future.

* + 1. The Community

As we’ve already discussed this project will directly help introverted Anime fans and AMV editors both financially and socially, as well as nowadays people on social media tend to enjoy the type of content that we plan on embracing on our application.

Judging from the researches[[1]](#endnote-1) that we have read it shows that a platform such as the one we’re planning making will be very helpful and needed for such a large fanbase (900+ [[2]](#endnote-2)million and growing exponentially).

* 1. Problem Statement

We listed the reasons as subsections ordered by importance.

* + 1. Fulfillment

Study shows that many introverts tend to relate to Japanese media (e. g. Manga, Anime),

And this study done in many foreign countries including the U.K, U.S.A and recently

the Gulf countries.

around 900 million [[3]](#endnote-3)people watch anime. And it is increasing exponentially day by day,

especially in this period of time.

So, we aim fulfill these people by making an application that will help them make new

connections and communicate with people from all around the world sharing similar

interests.

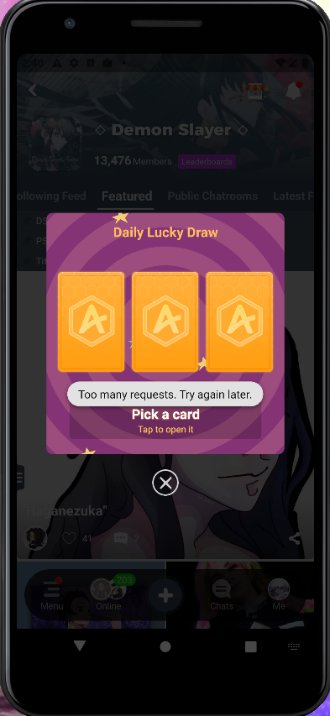
* + 1. Money Making Machines

In recent years the most popular anime studios (e. g. Mad House, Ufotable, Kyoto Animation) have created a new marketing scheme that is adaptable to new trends which is whenever that they plan to animate new Mangas, they work on a short exciting trailer and pay anime related pages on popular social medias to post a meme (Funny picture / video) or an AMV, now these pages usually relay on making the AMV on a computer with paid software for editing and spending around

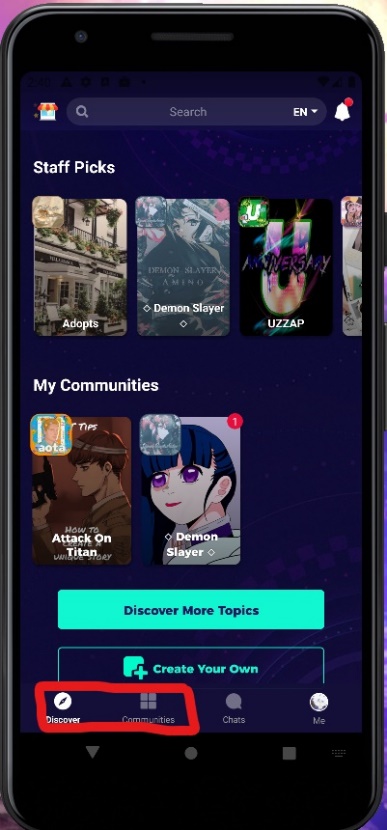
(1-5+) hours on making the AMV [[4]](#endnote-4)even more exciting.

* 1. Related Work
     1. Amino

Amino has a difficult UI to use design which makes the whole application very unappealing to any user, however Amino’s vision was to create a community-based application mimicking reddit, although Amino was successful only in a handful of communities and to be precise it found success in Anime related communities.

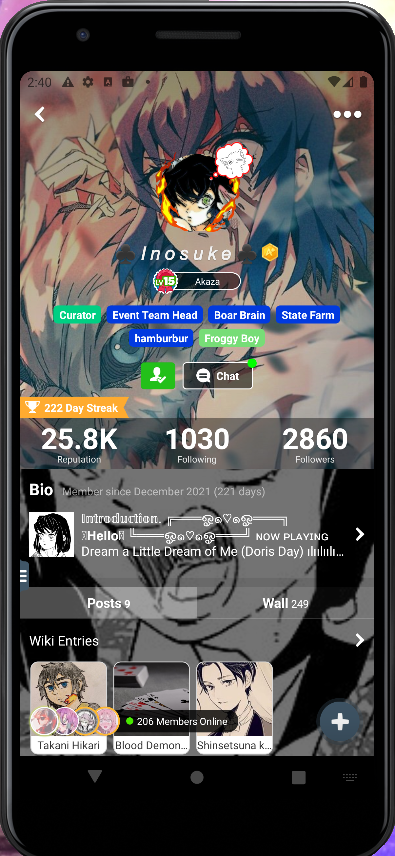


This is a bug that occurs when you try to claim a coin, which happens regularly.



These two buttons have the same functionality which is showcasing which communities are you in, the discover button has “Staff picked” tab with some communities and a search bar with no real recommended communities,

The communities button has your communities and a search bar.



This is an example of a level 15 user; this is supposed to be his profile page but as anyone can see the data is showing in a clumsy way by stuffing every piece of information about the account at once which makes it very confusing and unappealing for the eye.

Now the question is why would anyone leave Amino and come to our application?

The answer is very simple they won’t…

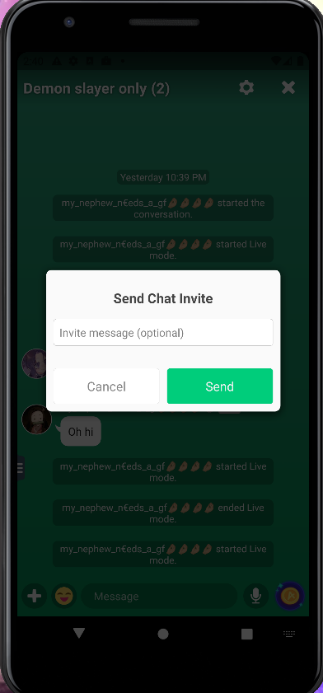
From a user’s perspective no one cares if they have two applications similar to each other, however they definitely will have a favorite application, and this is our goal.

What would make our application better?

If anyone tries to use Amino, they will have a difficult time navigating the application, since it is a teasing task to learn how to use such an application (Amino) so we plan on fixing the problem.

Amino has a circular design, which means you can loop inside the application which makes it very confusing for the user.

It is very obvious that you will need to provide an easy and smooth chatting between the community members, which in Amino it’s a terrible experience.



In this picture a dialog appears before attempting to chat with anyone which makes a bad UX, this picture is inside of a community chat, the only problem with it is extreme latency.



This is an example of a personal chat with a single user, also has latency but also with minor bug like the large space between the chat bar and the texts, and profile gifs not always rendering, sometimes you have to refresh a couple of times before it displays correctly.

Fixing the problem

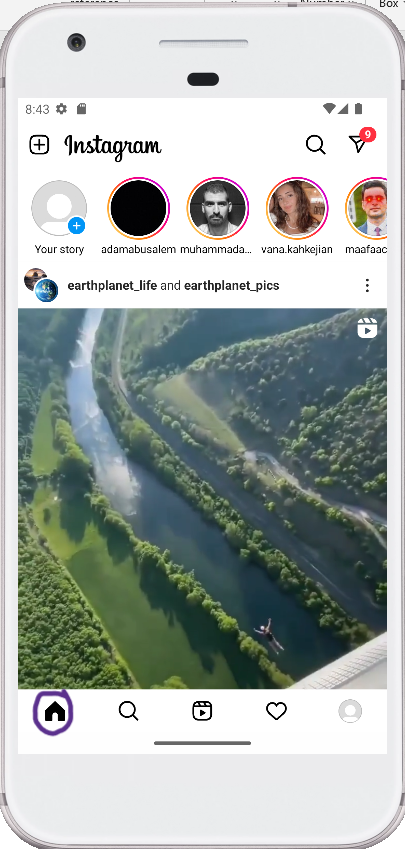
Two university students cannot possibly outclass a whole company with many resources and tons of employees, however a bigger company can!

We plan on using Instagram’s methods for implementing overlapping features.

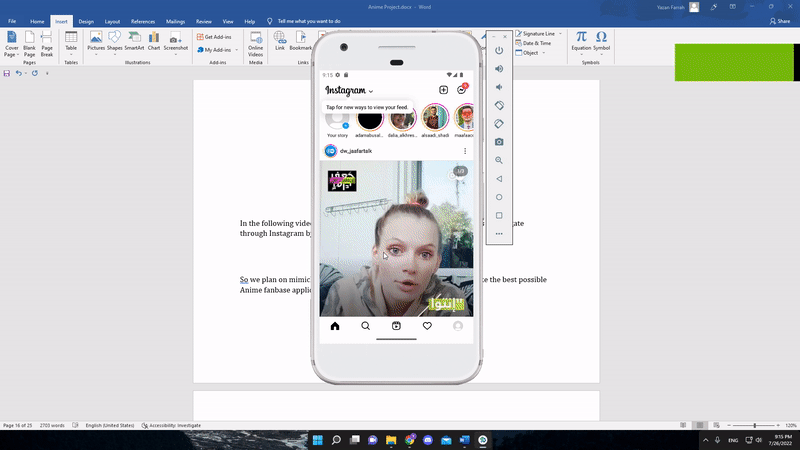
* + 1. Instagram

With over 12 years of experience and hundreds of millions of dollars sunk into making the perfect UI/UX design, Instagram successfully changed the way people think when using applications.

For example, you cannot possibly have a hard time navigating through the application or difficulty using a feature, looking at the home page (in the figure below) here which you can easily know that you are in the home page at the moment by looking at the highlighted button in the navigation bar below, it makes the user’s experience so simple and appealing.



In the following video we’ll be showing a short video on how simple it is to navigate through Instagram by sending a message to one of the user’s friends and then navigate back to the home page using the navigation bar.



We’re not going to get into every little detail about Instagram on here but we are going to be mimicking the way Instagram implement features to make the best possible Anime fanbase application.

* 1. Document Outline
* We following a book-like chapter naming and listing for the entire document
* The lists are always ordered by importance
* We specify each heading as appropriate (Sections, Paragraphs, subsections etc.)

Describe how this documentation is structured and what will be discussed in each of the following chapters.

Chapter 2  
Project Plan

* 2.1 Project Deliverables

List and describe the deliverables of the system. Examples of deliverables include: source code, documentation files, executables, datasets, databases, etc.

* 2.2 Project Tasks

Subdivide the project into high level tasks and provide a timeline for the completion of each task. Tasks can be:

* Analysis
  + requirement identification (e.g: fact finding methods, outputs, inputs, processes, performance, security levels, scalability, etc. )
  + requirement modeling,
  + development strategy (e.g. hardware, software tools, programming language use, etc.)
  + any other relevant analysis task (e.g. distribute a survey, interview, etc.)
* Design
  + Reports layouts
  + User interfaces
  + Database Design
* Implementation
  + Coding (e.g. develop a web-service)
  + Testing
  + Documentation (e.g. prepare a requirements document),

Format the information you provide into a table that includes the following for each task: a task number, a name and description, a time duration, and what dependencies need to be completed before the task begins.

Provide also charts like Gantt Charts and PERT Charts to illustrate the timeline of the project.

* 2.3 Roles and Responsibilities

Clarification: each role/responsibility isn’t exclusive to a one member; meaning that we will be working hand-to-hand on everything, however we’ll be classifying roles/responsibilities based on the primary member

|  |  |
| --- | --- |
| Project Management | Mutlaq Alsadeed |
| API Planning and Building | Mutlaq Alsadeed |
| Technology Research | Mutlaq Alsadeed |
| Admin panel design and implementation | Mutlaq Alsadeed |
| Backend Design | Mutlaq Alsadeed |
| Database Design | Mutlaq Alsadeed |
| AWS Functionality | Mutlaq Alsadeed |
| Research | Yazan Farrah |
| UI/UX Research | Yazan Farrah |
| Mobile App Design and Implementation | Yazan Farrah |
| Firebase Functionality | Yazan Farrah |
| Feature Planning | Yazan Farrah |
| State Management | Yazan Farrah |

**Project Management:** Dissecting the development process into comprehensive stages and injecting real production knowledge into the application.

**API Planning and Building:** Deciding which architectural concept to apply on the API’s and deciding endpoint functionality.

**Technology Research:** Deciding which technologies/framework best suited for the application

**Admin panel design and implementation:** Designing the admin panel in terms of UI/UX and functionality and implementing it using Angular 14

**Backend Design:** Configuring the API endpoints to be used in the mobile and web application.

**Database Design:** Choosing which DBMS to use and designing the schema.

**AWS Functionality:** Depending on the performance of the whole project we might resort to using AWS for more speed and efficiency.

**Research:** We used various resources for our researches that we linked down in the references section and we put our focus on the statistics, studying what the anime community likes, understanding the shared interests between the Anime fans.

**UI/UX Research:** Knowing what colors to use, widgets, animation so we present them in a comfortable way for our users making sure they enjoy spending more and more time on our application.

**Mobile App Design and Implementation:** Here we used an open-source UI software development kit (Flutter) since we can develop cross platform applications for Android, iOS and the web from a single codebase.

* 2.4 Risk Assessment

For each task, describe any associated risks that may prevent completing them. Indicate how probable the risk is, its impact on the system and how you plan to cope with it if it appears.

* 2.5 Cost Estimation

MongoDB:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cluster Tier** | **Storage** | **Ram** | **vCPUs** | **Base Price** |
| M50 | 160 GB | 32 | 8 | $2.00/hr |

Email Service:

|  |  |
| --- | --- |
| **Nodemailer** | **SendGrid** |
| $820/year | $30/10K emails |

Hosting:

Web /API Hosting: $10/month

Play Store (Android) Mobile Application Hosting: $25 (Lifetime)

App Store (iOS) Mobile Application Hosting: $100/year

Cloudinary Hosting: 90/month

* 2.6 Project Management Tools

Version control system: Git/GitHub

Communication: Discord

Chapter 3  
Requirements Specification

* 3.1 Stakeholders

|  |  |  |  |
| --- | --- | --- | --- |
| **Admin** | **AMV editors** | **Anime fans** | **Observer** |
| The admin will be able to delete/ suspend user’s actions etc... | Editors who would be able to cut anime videos, combine clips, custom video, etc... | Add or interact with posts, chat with users, etc... | A person who would be monitoring whether there are inappropriate content/ posts and can delete them or ban users |

* 3.2 Platform Requirements

Specify the *software* and *hardware* requirements for running the system. Clearly mention which requirements are a must and which are only recommended.

For systems that are made of sub-systems (e.g. a client side and a server side), make sure to list the requirements for each sub-system separately. For example, the client may need a browser to use the system, whereas the server may require different (more demanding) software and hardware requirements in order to respond to the client requests.

* 3.3Functional Requirements

Provide a detailed list of all the functional requirements of the system. For each requirement, specify exactly what the input, output, processes and main constraints are. Mark also each requirement as either recommended or essential.

Use a table that contains a numbered list of the requirements and their related information. This will facilitate understanding the requirements and referencing them in the proceeding sections and chapters.

* 3.4 Non-Functional Requirements

Examples of non-functional requirements include: requirements related to performance, storage limits, code quality, documentation, accessibility, security, reliability, scalability, portability, user interface ease-of-use, etc.

Provide an example for each non-functional requirement in your system

* 3.5Other Requirements

Include here any requirements that may not directly fall under any of the sections before. Examples include restrictions on which APIs can be used, data transmission protocols, data storage formats, etc.

Chapter 4  
System Design

* 4.1 Logical Model Design

In this section, provide both a high level and low level design of the system you will develop. You will follow in your design either the structured approach or the Object Oriented approach. Here is a list of the diagrams you need to provide in either case:

Object Oriented Approach:

* Use Case Diagrams.
* Object Diagrams.
* Package and Class Diagrams.
* Component Diagram.
* Deployment Diagram.
* Activity Diagram.
* Sequence Diagram.
* State Transition Diagram.

In both approaches, you need to describe the design of the data in the system. Use Entity Relationship Diagrams (ERD) and provide the detailed database schema.

Describe also any design choices that are related to the user interface. Describe the different screens (or web-pages), and how the flow moves between them. Use appropriate diagrams like trees to describe the structure of web-pages (for example).

Make sure to organize this chapter into sections and subsections in a manner that is appropriate to the provided information. Make sure also to stick the formatting used in the other chapters.

* 4.2 Physical Model Design

System Design must include the following:

- Reports Design

- User Interface Design

- Database Design: Database design must be presented in the form of normalized database (third normal form). This includes stored and transmitted data.

References

1. Schwartz, H., Eichstaedt, J., Kern, M., Dziurzynski, L., Ramones, S., Agrawal, M., Shah, A., Kosinski, M., Stillwell, D., Seligman, M. and Ungar, L., 2022. *Personality, Gender, and Age in the Language of social media: The Open-Vocabulary Approach*. [online] Available at: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0073791>. [↑](#endnote-ref-1)
2. <https://myotakuworld.com/how-many-people-watch-anime/> [↑](#endnote-ref-2)
3. <https://fictionhorizon.com/how-many-people-watch-anime/> [↑](#endnote-ref-3)
4. https://www.animemusicvideos.org/forum/viewtopic.php?t=92080/ [↑](#endnote-ref-4)