Project Proposal for L5 Practical Project

Htoo Pyae Linn
MCC Yangon Campus

Table of Contents

Computing Artifact

Background of the existing system

YouTube is an American social media and online video sharing platform owned by Google. YouTube was founded on February 14, 2005, by Chad Hurley, Jawed Karim, and Steve Chen, who were former employees of PayPal. Headquartered in San Bruno, California. On November 13, 2006, YouTube was purchased by Google for US$1.65 billion (equivalent to $2.39 billion in 2024). Google expanded YouTube's business model of generating revenue from advertisements alone, to offer paid content such as movies and exclusive content explicitly produced for YouTube. It also offers YouTube Premium, a paid subscription option for watching content without ads. Since its purchase by Google, YouTube has expanded beyond the core website into mobile apps, network television, and the ability to link with other platforms.

Problems of the existing system

Because of the premium version from YouTube, users who use the free version have to watch ads, sometimes 2 to 3 ads per 15 minutes video. These can lead to distraction to users for long–term content and educational videos, breaking user’s concentration.

YouTube has a download feature but it is time limited; downloaded videos can only be available for a specific amount of time, needing to be online to redownload the video. This can be inconvenient for users who live in areas where there is slow or no internet connection.

YouTube is for entertainment purposes but sometimes users use it for productivity such as learning. In such cases when users want to take notes at a specific time of the videos, users have to use an external application to achieve the behavior which makes the user experience bad.

Scope of the proposed system

PlexiPlay will be developed by identifying the pain points of YouTube; presenting the analysis of use cases and class diagrams. We will define the functional areas, methods or framework that will be used in the project. Project, Risk and Configuration management will also be shown in the following chapters. The implemented system will be tested with unit and integration tests. We will also release the user's manual guide.

The new system will mainly solve the problems of the existing system, then will have improvements in UI/UX also. Firstly, We will not show any ads in videos, making the user experience while watching video seamless and smooth. Second, we will remove download duration, allowing the users to keep the video in local storage forever. The last one will be adding the ability to take notes at a specific time of the video so users can be more productive while watching learning or educational videos. The core features of PlexiPlay are the basic features of video social media apps like uploading, downloading video, commenting the video and reply to the comments. Since we remove ads completely, we have to change the way of monetization. PlexiPlay will be open-sourced and will add the donation feature where users or developers who find the app useful can donate to the application. Since the system is open-sourced, users and developers can request, contribute and submit new features to the system.

Advantages of the proposed system

There are the advantages from the purposed system -

- There are no ads so users can watch the videos seamlessly, making users concentrate more on the content.
- Simple UI/UX allows users to focus more on the content, understand the functionalities of the system intuitively.
- Remove the download time limit so users can keep the video forever which is more convenient for the users who have limited or no internet connectivity.
- Added the ability to take notes at specific times of the videos so users can be more productive by not switching apps to get this behavior.
- Without ads, PlexiPlay runs faster and consumes less system and data resources, so it will work well in lower-end devices.
- Ensures better privacy to users by not tracking user activity to show ads.
- Open-source means more secure and trustworthy, developers but not limited to users can inspect the code and contribute to the system.

There are some minor advantages of the proposed system such as showing the uploading, downloading progress, ability to comment to video, reply to the comments.

Aims and Objectives of the project

Aims

To provide a smooth experience by not having ads, also ensure better users' privacy, enhance offline usage and integrate note–taking features.

Objectives

- To develop ad-free video platform that allows uninterrupted experience
- To allow users to download content for lifetime without limitations
- To integrate note-taking feature for productivity for content creators and students
- To develop a simple UI/UX so every users with any technical level can use easily
- To ensure users' privacy by not integrating ads and tracking users' activity
- To optimize system performance so users can enjoy videos smooth and responsive and also allow users with low-ends devices to use
- To implement donation based model for monetization instead of ads that tracks users
- To make a project open-source and allow community and developers to request, contribute features Functional scope of the proposed system

Functional Scopes of the Proposed System

Video Operations

- Video upload/download with progress
- Local storage management for offline access
- Video streaming with playback controls

Content Organization

- Set video category and tag
- Search/Filter videos
- Playlist management

Community Feature

- User registration
- User profile
- Comment/reply to video content
- Social sharing

Note Taking Feature

- Create note at specific timestamp
- Share note to other users

Open-Source Development

- Version control and release management
- Allow community contribution
- Documentation and issue tracking

System Design Architecture

Framework

We will use the Flutter application framework to build Android and iOS applications for PlexiPlay. Flutter is an open source framework for building beautiful, natively compiled, multi-platform applications from a single codebase.

Object-Oriented Analysis and Design

Object-oriented analysis and design (OOAD) is an approach to analyzing and designing a computer-based system by applying an object-oriented mindset and using visual modeling throughout the software development process. It consists of object-oriented analysis (OOA) and object-oriented design (OOD) – each producing a model of the system via object-oriented modeling (OOM). Proponents contend that the models should be continuously refined and evolved, in an iterative process, driven by key factors like risk and business value.

Design Supporting Tools

We will use Lucidchart to draw UML, class diagrams, use case diagrams and flowcharts.

System Development Architecture

UI Design

We will use Flutter widgets and Material 3 design system to build PlexiPlay user interface.

Development tools and Platform

We will use IntelliJ IDEA Ultimate IDE to develop Flutter applications with Dart programming language. This IDE can also be used with Java, Kotlin, C, C++ and many other languages. IntelliJ IDEA can run on Windows, macOS and Linux OS but we will primarily build on macOS to be able to build iOS applications. Since Flutter uses Dart, we will be using Dart as a primary programming language but we will also need Kotlin for Android and Swift for iOS too for platform specific logic like video processing, capturing battery parcentage.

Database

We will use the PostgreSQL database which is also known as Postgres, is a free and open-source relational database management system (RDBMS) emphasizing extensibility and SQL compliance. PostgreSQL features transactions with atomicity, consistency, isolation, durability (ACID) properties, automatically updatable views, materialized views, triggers, foreign keys, and stored procedures. It is supported on all major operating systems, including Windows, Linux, macOS, FreeBSD, and OpenBSD, and handles a range of workloads from single machines to data warehouses, data lakes, or web services with many concurrent users.

Backend as a Service

Being a mobile developer with a little backend knowledge, we will use Supabase backend to handle auth, database and other service side logic. Supabase is a BaaS which offers PostgreSQL database with realtime, edge functions to handle server side logic and authentication services to handle users.