UNIVERSITY OF BIRMINGHAM

Videosharetube
UML Documentation

March 27, 2020

Authors
Group 42

Name	Student No.	Email
Josh Alexander	1762887	jta787@student.bham.ac.uk
Devon Brazier	1532879	dxb579@student.bham.ac.uk
Simon Hyslop	2091389	sch989@student.bham.ac.uk
Ben Irving	1243239	bxi239@student.bham.ac.uk
Angus Mills	2075583	axm1594@student.bham.ac.uk

Supervisors
Shuo Wang

University of Birmingham
Birmingham, B15 2TT

Contents

Contribution

Josh Alexander: 20%

Devon Brazier: 20%

Simon Hyslop: 20%

Ben Irving: 20%

Angus Mills: 20%

1 Scope

Videosharetube is a mobile application which enables members of the public to view and share videos online. As well as watching videos uploaded by other members, users also have access to a library of professionally created content – including popular TV shows, sports events, and e-learning opportunities. Advertisers can also use the platform to promote their services, and the revenue from this allows the basic Videosharetube service to be provided free-of-charge to end users.

1.1 Users

There are 3 types of external user to the platform:

- Customer User - upload free content to the platform; watch free and paid content
- Video Company - upload paid content to the platform
- Advertiser - create adverts on the platform (subject to membership fee)

There is also an internal user type:

- Moderator - check videos for compliance (where these fail automated checks)

Users of the platform (both Customer Users and Video Companies) can upload content, and view videos. Videos uploaded by Video Companies are paid content, and cost £1 per hour to view.

Advertisers require a subscription to use the platform, and in return can have their video advertisements seen by app users. This revenue model is what funds the Videosharetube platform.

The use of an automoderator system ensures that videos uploaded to the platform are suitable, before they are published. Compliance checks are conducted through this automated moderation process. Content which fails compliance is manually checked by a Moderator user.

1.2 Assumptions

For any video content uploaded to the platform:

- The file format and quality of all videos is the same, and can be directly used for playback on the app
- The user uploading owns all copyright to their video, and we do not need to validate this

For collecting payment from users:

- Payments received by credit/debit card or direct debit will have a 100% success rate, i.e. we do not need to consider payments which fail or are charged back/refunded

When considering how the platform will be hosted:

- We do not consider the length of videos, or storage space they will take up

2 Requirements

2.1 Functional Requirements

1. User Account

   1.1. Users can sign up to Videosharetube to create/customize profiles, post videos.

   1.2. Users can comment on videos

   1.3. Users can make videos public or private and share at their choosing.

   1.4. Users can follow professional channels and stations.

   1.5. Users shall be able to customize their levels of security and privacy.

   1.6. Users can skip and filter adverts that they see.

   1.7. Users can rank adverts and their providers.

   1.8. Users can pay with debit and credit card or by direct debit.

   1.9. The user must be able to search for videos and/or stations.

2. Business Accounts

   2.1. Companies can access Videosharetube by a pay as you go model of £1 an hour.

   2.2. Business and companies can advertise on the platform for a fee of £300 upfront and £500 monthly.

3. Moderator Account

   3.1. Manual moderators must moderate videos/profiles that fail automated moderation.

4. System

   4.1. The system must have an automated moderation system to reduce abuse of the service.

   4.2. Finance and payment solutions will be provided by third party.

   4.3. System must display advertisements on screen, via push-notifications and emails.

2.2 Non-functional Requirements

Product Requirements

5. Accessibility

    5.1. The system must be stable and accessible and available for 24/7 up-time.

    5.2. The system should have a high performance but not compromise stability and security.

6. Reliability

    6.1. The system must be scalable.

7. Usability

    7.1. The system will have a profile subsystem for recommendations.

    7.2. The subsystem will use profile information to provide recommendations.

    7.3. The system will display users personal videos on their profile page depending on users privacy settings.

8. Security

    8.1. The system must be secure.

Organisational Requirements

9. Compliance

    9.1. The system must be compliant with users data, such as card details.

10. Ethics

    10.1. The system should promote ethical use of the application.

    10.2. The system may use gamification techniques to minimize addiction of the application.

3 Use Case Modelling

3.1 Use Case Diagram

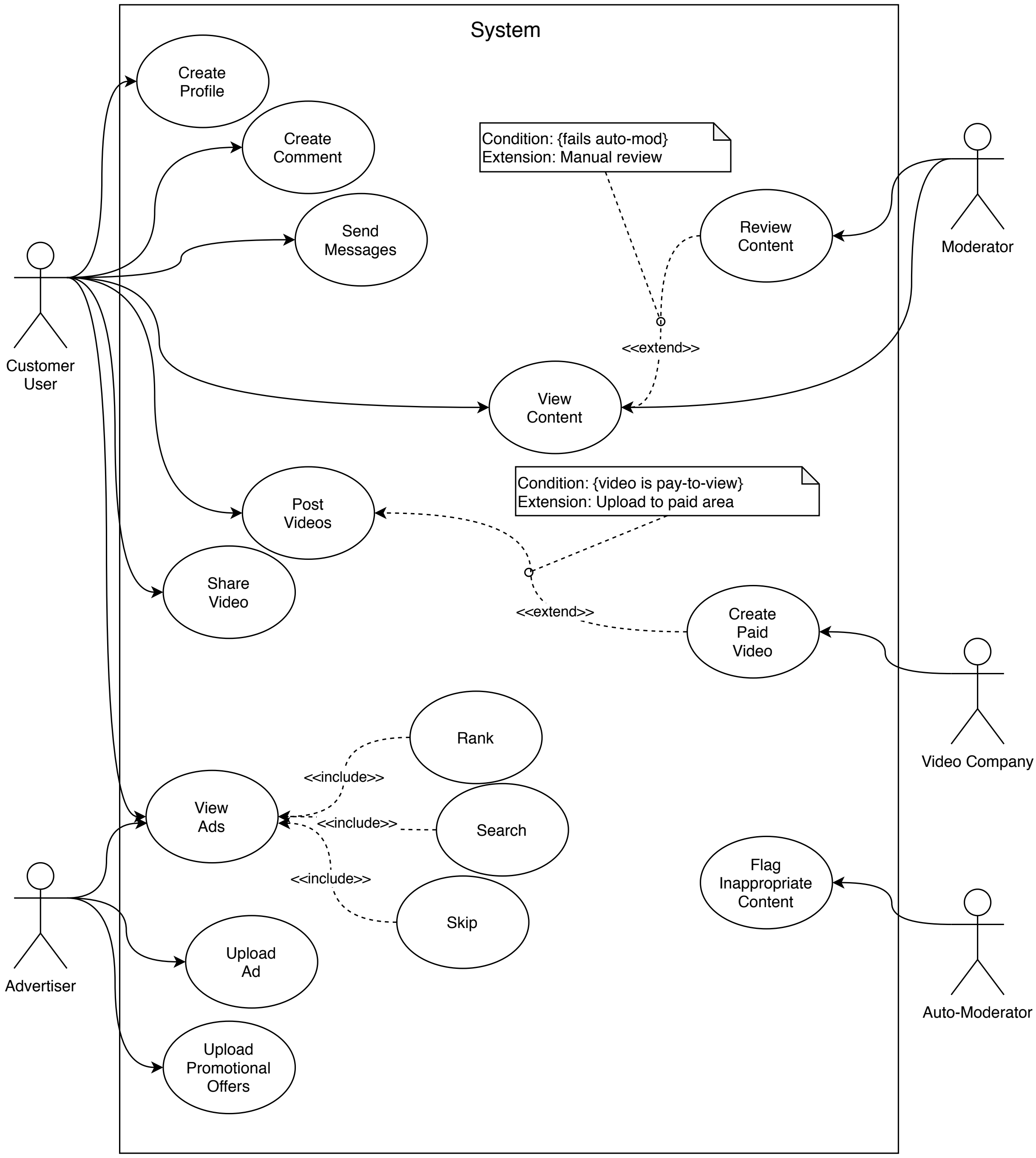

Figure 1: Use Case Diagram

3.2 Use Cases

- James - Posting a Video
- Susan - Viewing an Advert
- Brian - Manual Moderation

3.2.1 James

Scenario: A Customer User (James) wishes to upload a video of his dog doing a trick onto Videosharetube so that everyone can see it. James logs into his Videosharetube account and navigates to the upload video section. He then selects the video file and tags it with "funny" and "dog" with the description "Spot does a jump". He then waits for the video to upload. Once uploaded he receives a confirmation message and is able to view his post.

Description: Customer Users and Video Companies both need to be able to post videos on the platform.
Actor(s): Customer Users, Video Companies.

Pre-conditions:

- Users and video companies have access to the platform.
- The actors must be logged in.
- The actors must have prepared video file to upload in mp4 on their home system.

Main Flow Scenario

- The user logs in to the system by entering their username and password, they will then be directed to the home screen. Here they can then select the post video option.
- The user then selects the video file that they wish to upload when prompted.
- The user can then add a description, tags and select if they want it to be public or private.
  - If the user sets the video as public the system will allow any other user to view it.
  - If the user sets the video as private only users with a direct link and password to the video will be able to view it.
- For Video Companies, they can create a paid video, and upload it in a similar manner except it will only upload to the paid area for viewers.

Post-conditions

- For Customer Users, free videos are accessible to be searched and viewed on the platform by anyone if public, or only by a private link.
- For Video Companies the video is uploaded to the paid area and can be searched and viewed by paying users.
- In both cases they receive a confirmation that the video was uploaded successfully.

Alternative Flow Scenario:

- The video upload fails and as such the video is not posted.
- The Customer User/Video Company will then get an error message detailing why the video did not upload and the option to retry.

3.2.2 Susan

Scenario: Advertisements will be periodically served to viewers. When a Customer User requests to watch a video, an advert will sometimes show before the requested video starts playing.

Description: A user Susan watches an advert before seeing a video.

- Customer Users will view ads that are posted by advertisers.
- Advertisers can also view the ads that they have posted.

Actor(s): Customer User, Advertiser.

Pre-conditions:

- Advertisers will have posted the advert.
- The advert will be within the Customer User's advert filter settings.

Main Flow Scenario

- The user selects to watch a public video from the home screen or a search.
- Customer User on the system will see the ad, either prior to the video they have selected or in a static format adjacent to the video.
- The ad will continue to play or present until the user leaves that page, the advert finishes, or the user chooses to skip the ad.
- To be able to skip the ad, the user must have watched a certain amount of the ad.

Post-conditions

- The user will be prompted to rank the ad they have seen, so that the system can tailor future adverts to that user based on preference.
- If the user chooses to skip the ad, the ad playback should cease and not restart
- Advertisers will be informed of the advert's viewing statistics.

Alternative Flow Scenario:

- If the user type is an Advertiser then they are able to view their own adverts that they have posted on the system.
- They are then able to edit or delete these adverts, in addition to posting new ones.
- They are able to view statistics of the number of views on their advert as well as the number of skips and rank.

3.2.3 Brian

Scenario: Moderators are able to monitor user content to remove that which is inappropriate.

Description: A user has uploaded a video but it has failed automated moderation checks, so moderator Brian must manually moderate it.
Actor(s): Moderators.

Pre-conditions:

- Actor logged in with moderator account.
- Video has failed auto-mod and is flagged for review - removed from public view awaiting manual review.

Main Flow Scenario

- The moderator will view the flagged content.
- The moderator will decide if the content violates Videosharetube policy or not.
- Either the video is removed, or the video is flagged as safe and returned to public view.
- If the video violates Videosharetube policy, depending on the severity of the offense, the user may be banned, reported to authorities or receive a warning and/ or a strike against their account.

Post-conditions

- The video is returned to public view.
- The video is flagged as safe in order to avoid failing auto-mod again.

Alternative Flow Scenario:

- The video is deemed inappropriate and is deleted.
- The user who uploaded the video has a flag added to their channel to warn of new uploads.
- The user receives a message of the offense and the resulting actions.

4 Activity Diagram

The Advertiser user wishes to make a payment to Videosharetube. First, they view their account and choose to make a payment, the system then shows their payment options, either a one-off payment or a monthly payment. The system then shows their outstanding balance. At this point the user can cancel the payment or confirm. The system then shows the user the payment methods and the user can select credit card or debit. The system then confirms the payment with the bank and if it fails, the system displays an error message, if it is successful the system displays a success message and updates the balance.

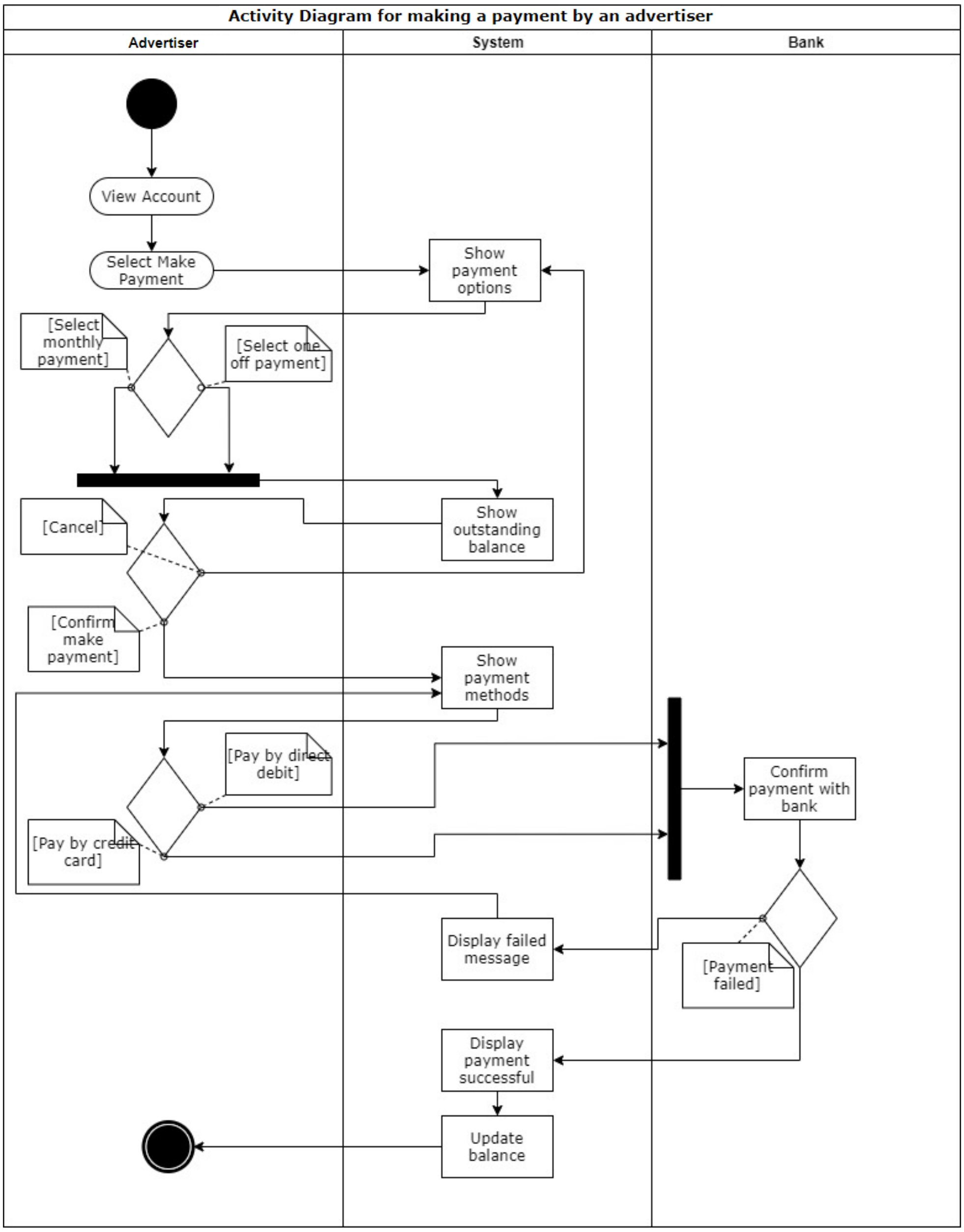

Figure 2: Activity Diagram

5 Class Analysis

5.1 Noun-verb Analysis

Using the use case diagram and the user scenarios, a noun-verb analysis was conducted on the functional requirements of the system to identify possible classes and operations. Where a possible class was identified but deemed irrelevant, it was omitted.

5.1.1 Noun Analysis

Candidate Class	**Use**
User	Class
Username	Attribute of User Class
Password	Attribute of User Class
System Access	Attribute of User Class
Moderator Privileges	Attirbute of User Class
Video Company	Subclass of User
Channel	Attribute of Video Company
Videos Posted	Attribute of Video Company
Advertiser	Subclass of User
Adverts Posted	Attribute of Advertiser
Moderator	Subclass of User
Moderation Queue	Attribute of Moderator
Customer User	Subclass of User
Profile	Attribute of Customer User
Subscriptions	Attribute of Customer User
Premium Account	Attribute of Customer User
Video	Class
Public Visibility	Attribute of Video
Video Title	Attribute of Video
Moderating Status	Attribute of Video
Tags	Attribute of Video
Premium Access	Attribute of Video
Comment	Class
User	Attribute of Comment
Content	Attribute of Comment
Score	Attribute of Comment
Advert	Class
Title	Attribute of Advert
Rank	Attribute of Advert
Advertiser ID	Attribute of Advert
Tags	Attribute of Advert
Language	Attribute of Advert
Auto-Moderator	Class
Flags	Attribute of Auto-Moderator
Recommendation Engine	Class

Table 1: Noun Analysis

5.1.2 Verb Analysis

Verb	**Possible Class Method**
Create Account	User
Delete Account	User
Change Email Address	User
Change Username	User
Compare content to Flags	Auto-Moderator
Send flagged videos to moderator	Auto-Moderator
Approve Video	Auto-Moderator, Moderator
Post Ad	Advertiser
Edit tags on own ads	Advertiser
Follow Profile or Station	Customer User
Post Comment	Customer User
Create Profile	Customer User
Customize own Profile	Customer User
Remove Subscription	Customer User
Make own video public or private	Customer User
Skip Ad	Customer User
Rank Ad	Customer User
Filter Ad	Customer User
View premium videos	Customer User
View Videos	Customer User
Delete own comments	Customer User
Edit own comments	Customer User
View Ad	Customer User, Advertiser
Post Video	Customer User, Video Company
View list of own videos	Customer User, Video Company
Delete own videos	Customer User, Video Company
Edit own videos	Customer User, Video Company
Edit search tags on own video	Customer User, Video Company
View videos that failed auto-moderation	Moderator
Remove videos	Moderator
Recommend Advert	Recommendation Engine
Recommend Video	Recommendation Engine
Email User	Recommendation Engine
Notify User	Recommendation Engine

Table 2: Verb Analysis

5.2 Class-Responsibility Collaboration Cards

From the noun-verb analysis coupled with the use-case scenarios and system requirements, CRC cards were made with the responsibilities and collaborations stated for each class.

User (Abstract)	
Responsibilities	Collaborations
Create Account	Customer User
Delete Account	Advertiser
Change Email Address	Video Company
Change Username	Moderator

Table 3: CRC Card - User

Customer User	
Responsibilities	Collaborations
Follow Profile or Station	Video
Post Comment	Advert
Make own video public or private	Recommendation Engine
Skip Ad	
Rank Ad	
Filter Ad	
View premium videos	
Delete own comments	
Edit own comments	
View Ad	
Post Video	
View list of own videos	
Delete own videos	
Edit own videos	
Edit search tags on own video	

Table 4: CRC Card - Customer User

Video Company	
Responsibilities	Collaborations
Post Video	Video
View list of own videos	
Delete own videos	
Edit own videos	
Edit search tags on own video	

Table 5: CRC Card - Video Company

Advertiser	
Responsibilities	Collaborations
Post Ad	Advert
Edit tags on own ads	
View Ad	

Table 6: CRC Card - Advertiser

Moderator	
Responsibilities	Collaborations
Change moderation status of video	Video
View videos that failed auto-moderation	Auto-Moderator
Remove videos	

Table 7: CRC Card - Moderator

Recommendation Engine	
Responsibilities	Collaborations
Recommend video	Video
Recommend advert	Customer User
Email user	
Notify user	

Table 8: CRC Card - Recommendation Engine

Video	
Responsibilities	Collaborations
Store data for video	Customer User
Store privacy information	Video Company
Store moderation information	Moderator
Store licensing information	Recommendation Engine
	Auto-Moderator

Table 9: CRC Card - Video

Advert	
Responsibilities	Collaborations
Store data for advert	Customer User
Store tags	Advertiser
Store ownership details	Moderator
Store moderation information	Recommendation Engine

Table 10: CRC Card - Advert

Auto-Moderator	
Responsibilities	Collaborations
Compare content to Flags	Moderator
Send flagged videos to moderator	Video
Change moderation status of video	
Analyse video or advert	

Table 11: CRC Card - Auto-Moderator

5.3 First-Cut Class Diagram

Using the noun-verb analysis and CRC cards produced, as first cut class diagram was created.

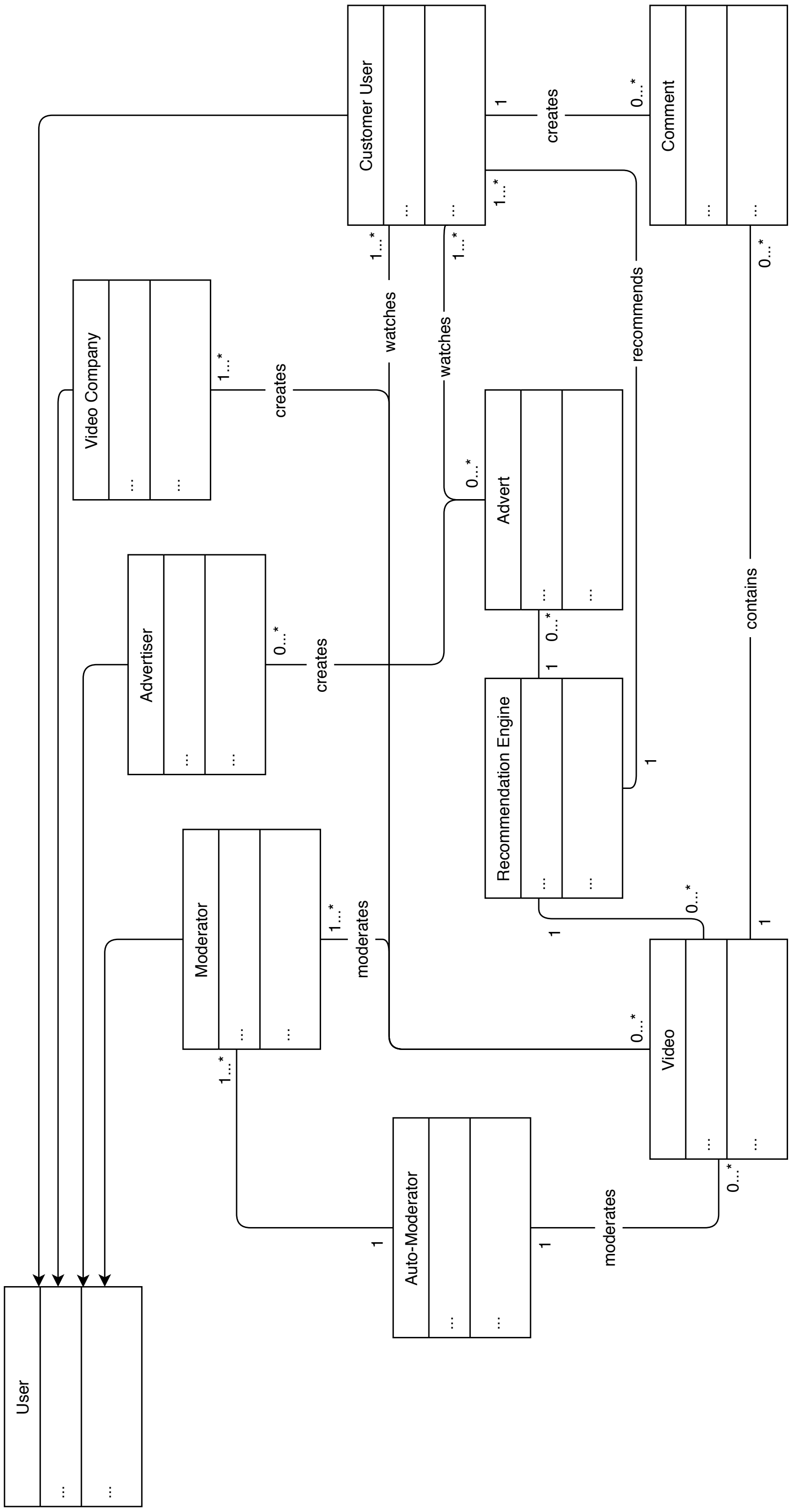

Figure 3: First-Cut Class Diagram

5.4 Full Class Diagram

Following our first cut class diagram and using our CRC cards and noun-verb analysis, a full class diagram was produced detailing all attributes, visibilities and relationships between classes.

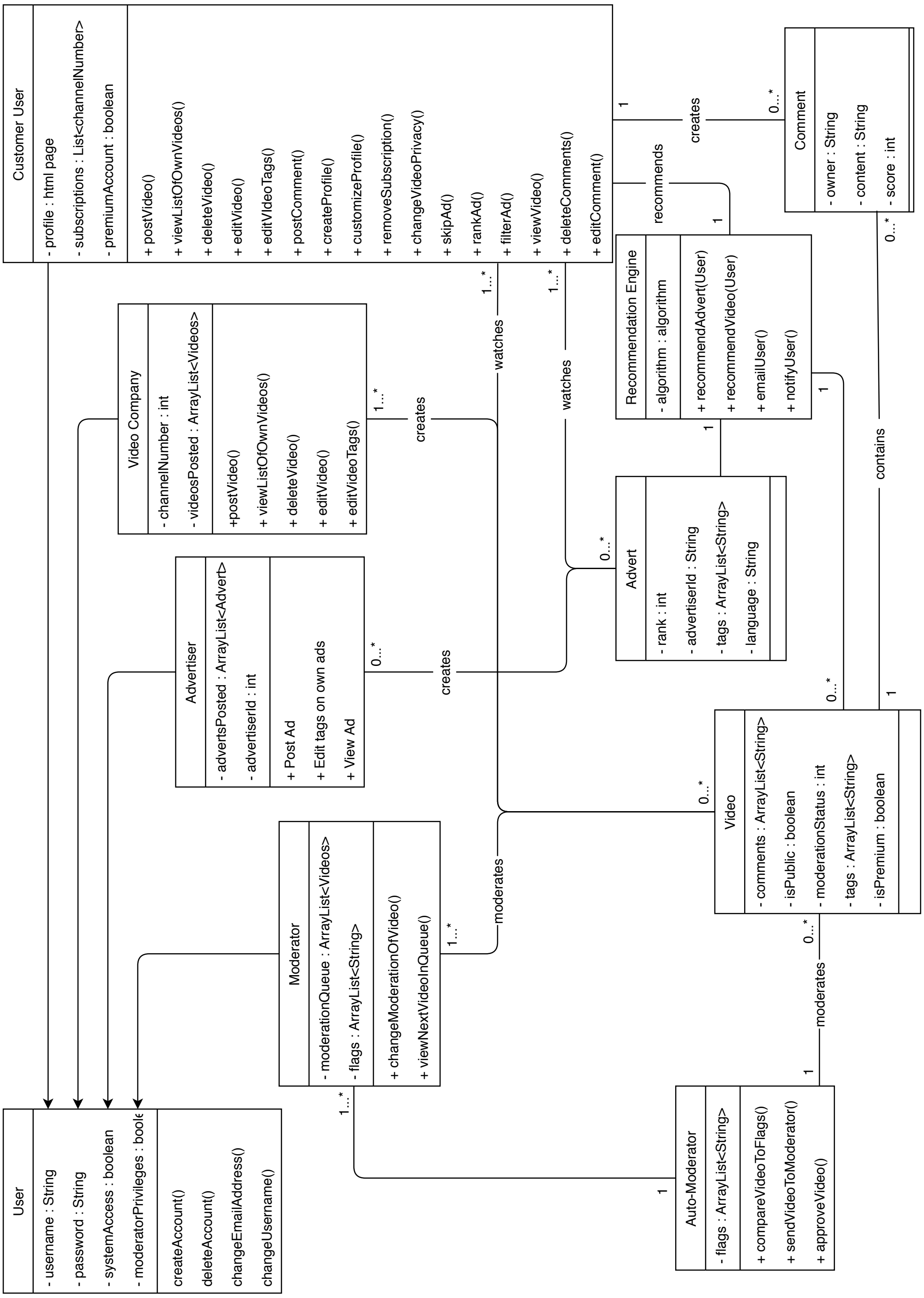

Figure 4: Full Class Diagram

6 Object Diagram

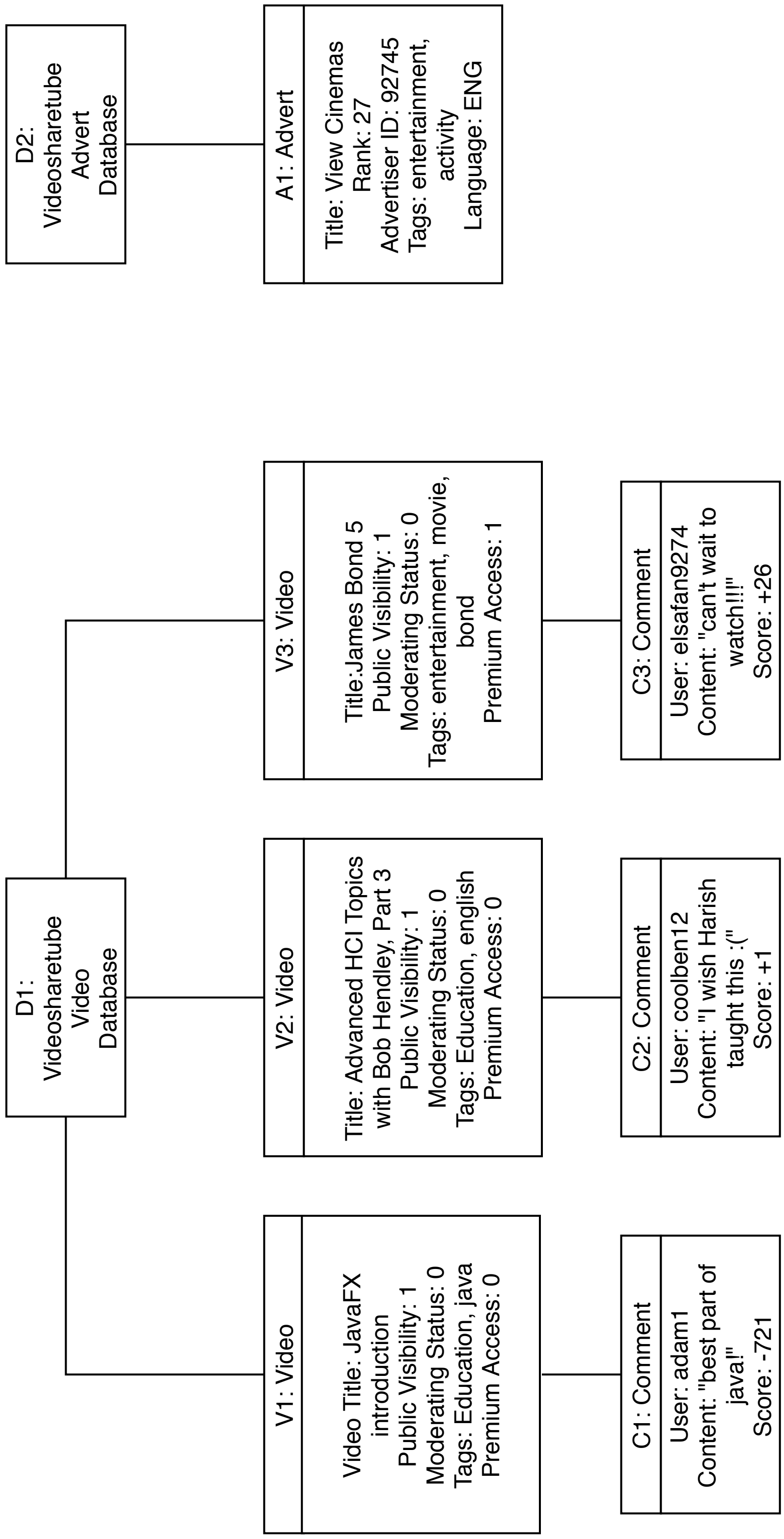

Figure 5: Object Diagram

7 Sequence Diagram

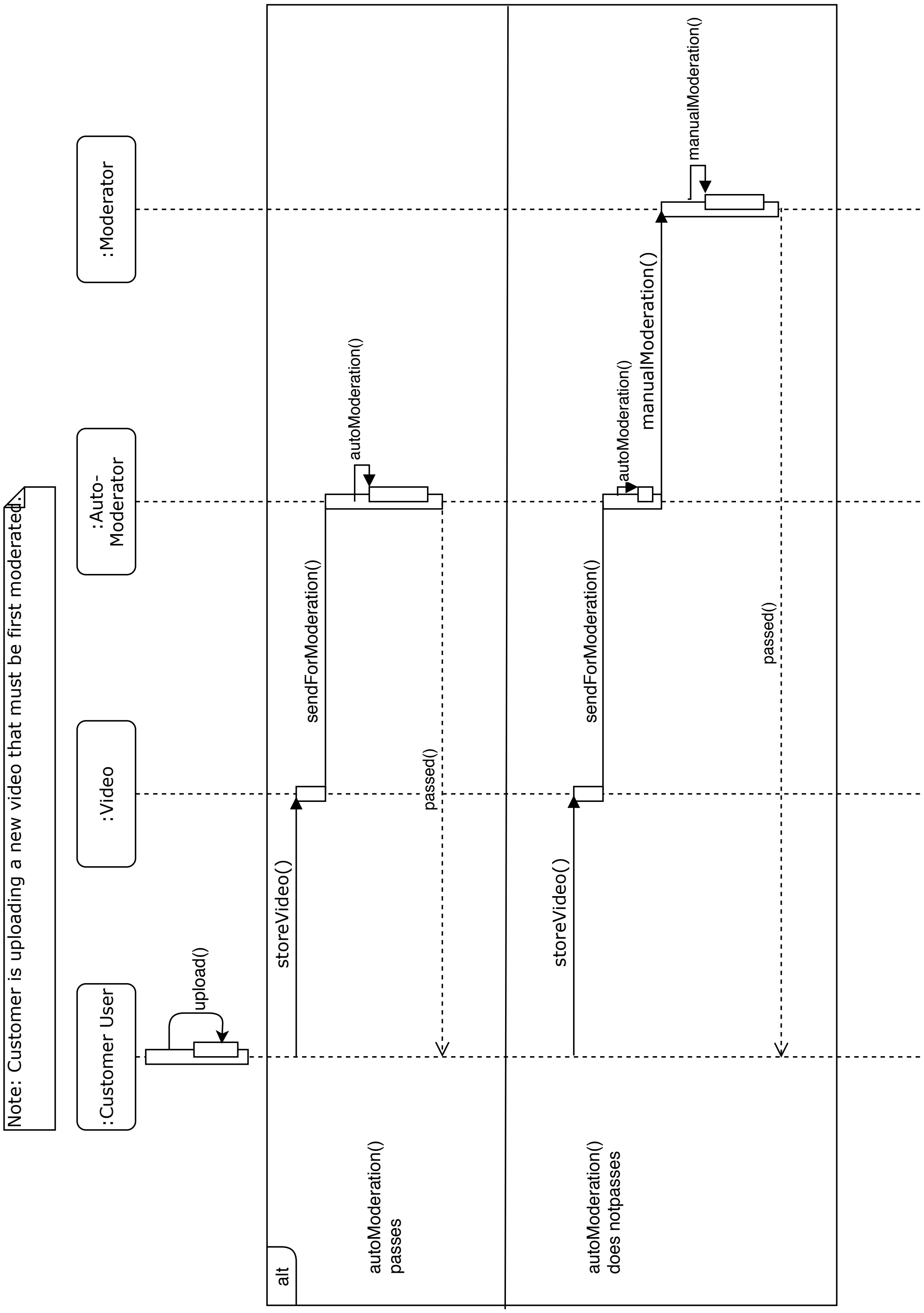

Figure 6: Sequence Diagram

8 State Diagram

This state diagram shows how a user can view, skip, search and rank ads and the various states the system goes through to achieve these. It is assumed that the user has an account and is able to view existing ads.

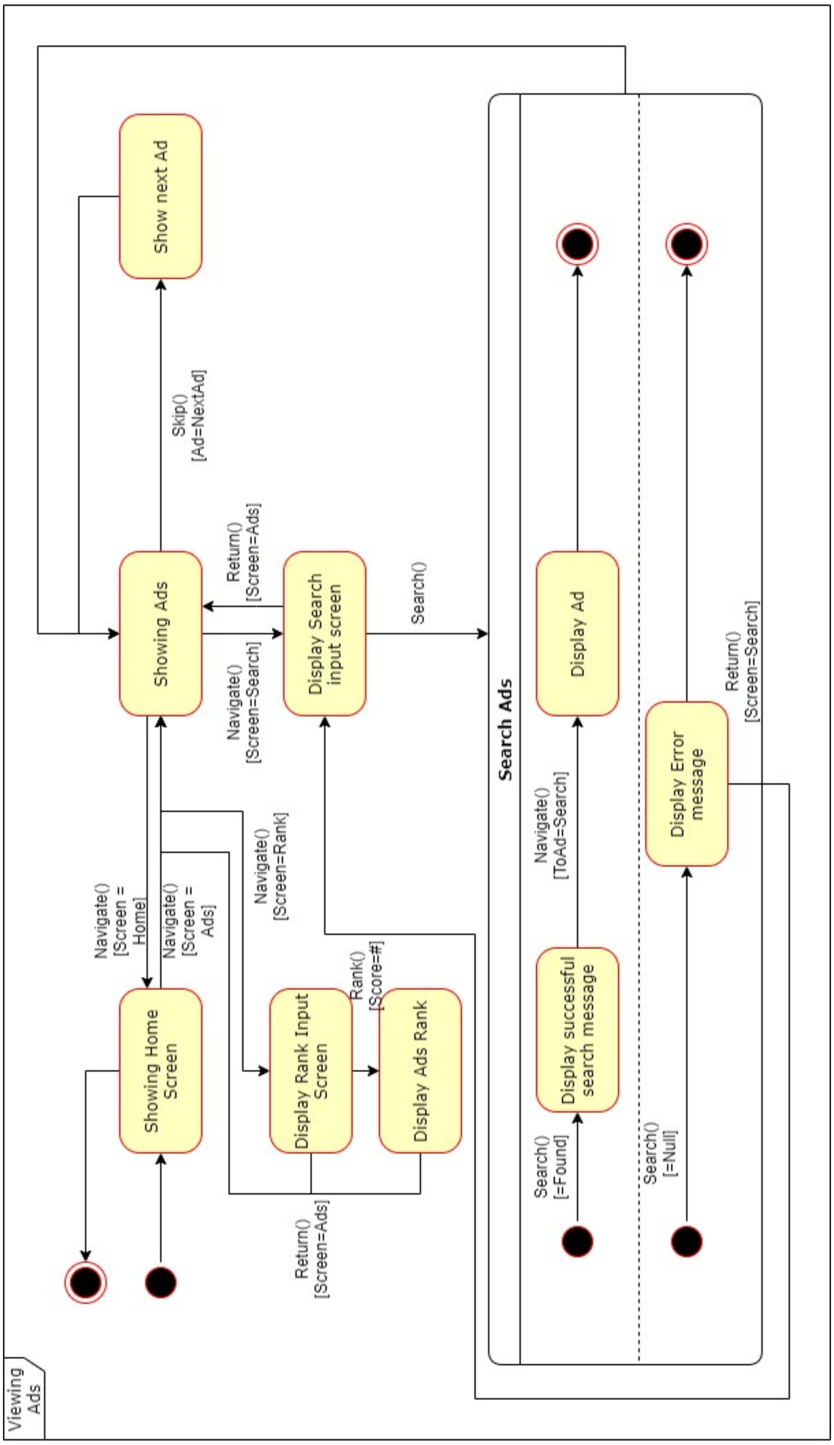

Figure 7: State Diagram

9 Architecture and Deployment

9.1 Architecture 1

9.1.1 Component Diagram

Component diagram describing a potential design/architecture in the form of a three-tier system, comprised of a client, server and database layer.

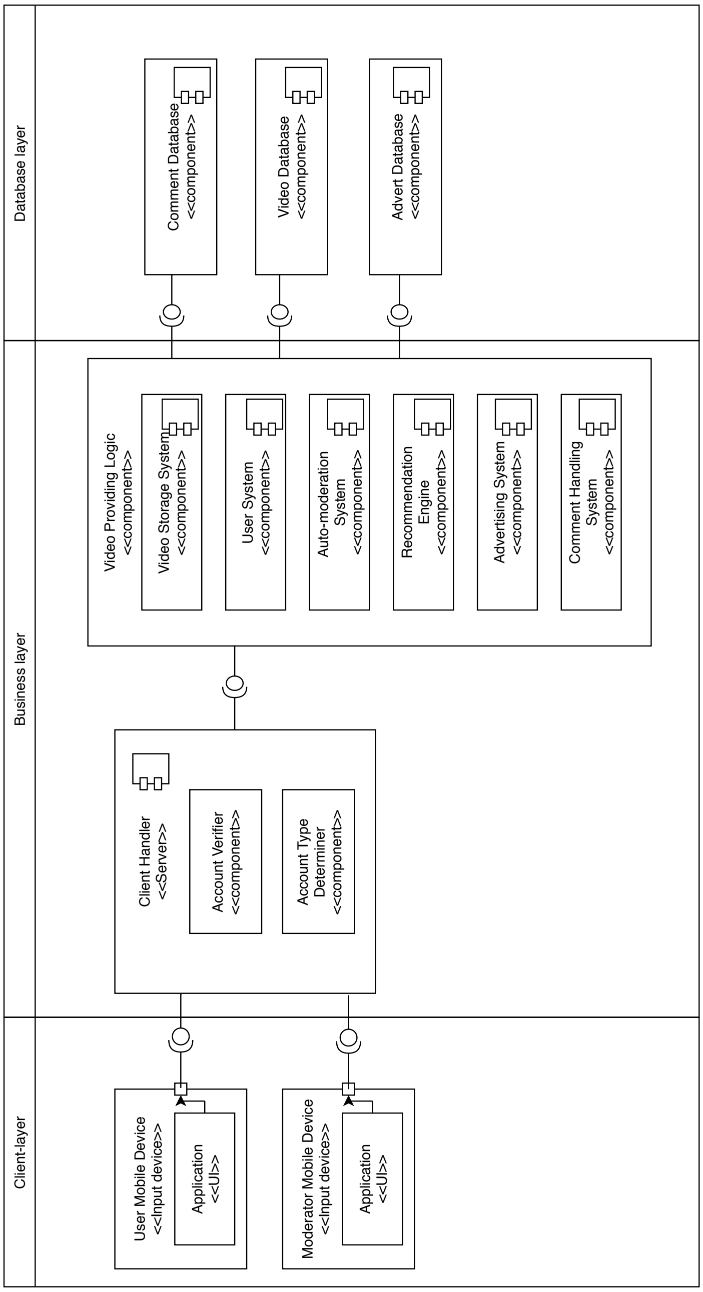
Database layer
Comment Database <<component>>
Video Database <<component>>
Advert Database <<component>>
Business layer
Video Providing Logic <<component>>
Video Storage System <<component>>
User System <<component>>
Auto-moderation System <<component>>
Recommendation Engine <<component>>
Advertising System <<component>>
Comment Handling System <<component>>
Client Handler <<Server>>
Account Verifier <<component>>
Account Type Determiner <<component>>
Client-layer
User Mobile Device <<Input device>>
Application <<UI>>
Moderator Mobile Device <<Input device>>
Application <<UI>>

Figure 8: Architecture 1: Component Diagram

9.1.2 Deployment Diagram

Deployment diagram coupled with figure 8, showing the proposed three-tier architecture.

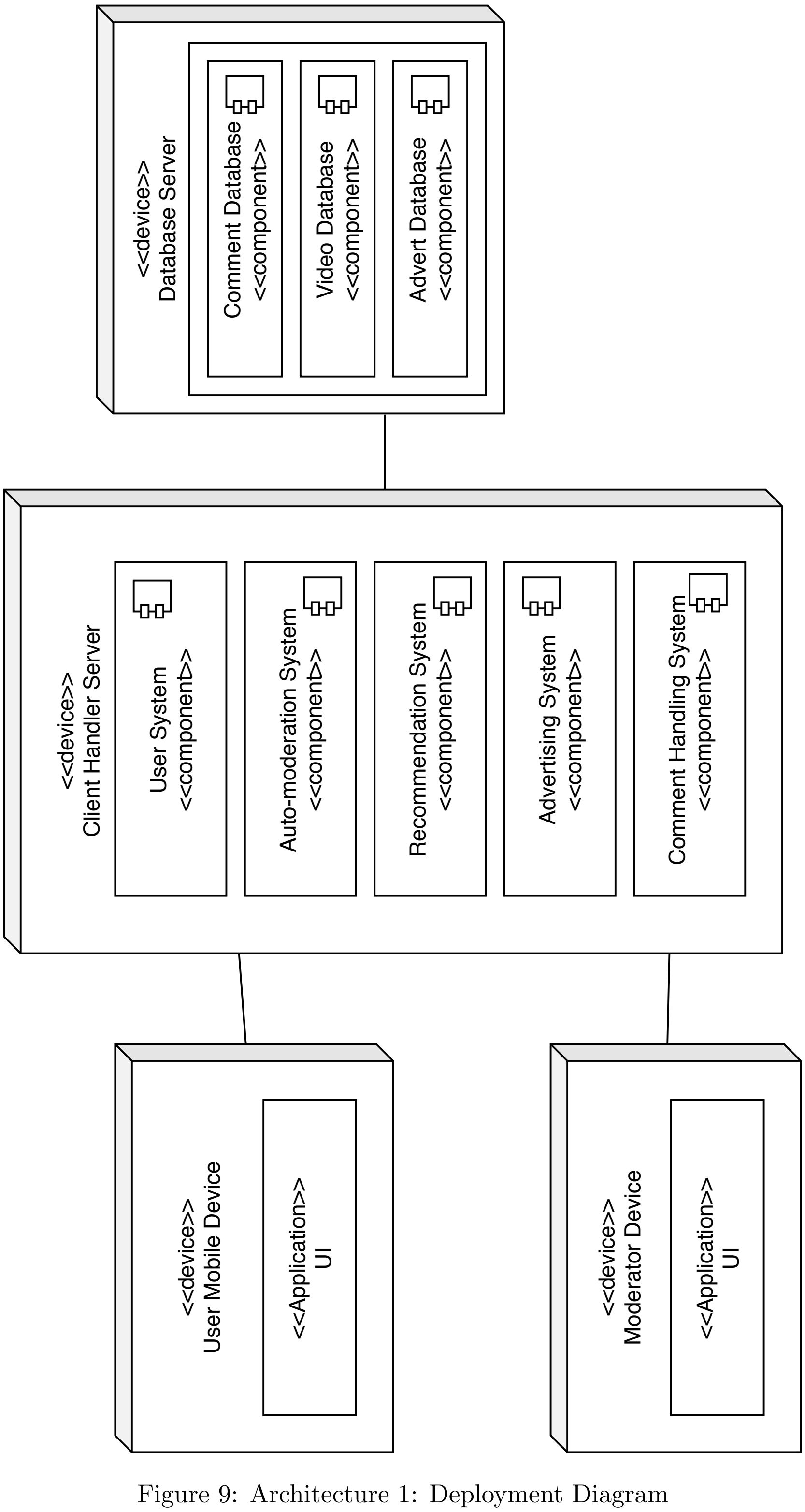

Figure 9: Architecture 1: Deployment Diagram

9.2 Architecture 2

9.2.1 Component Diagram

Component diagram showing a design in the form of a microservices architecture for Videosharetube.

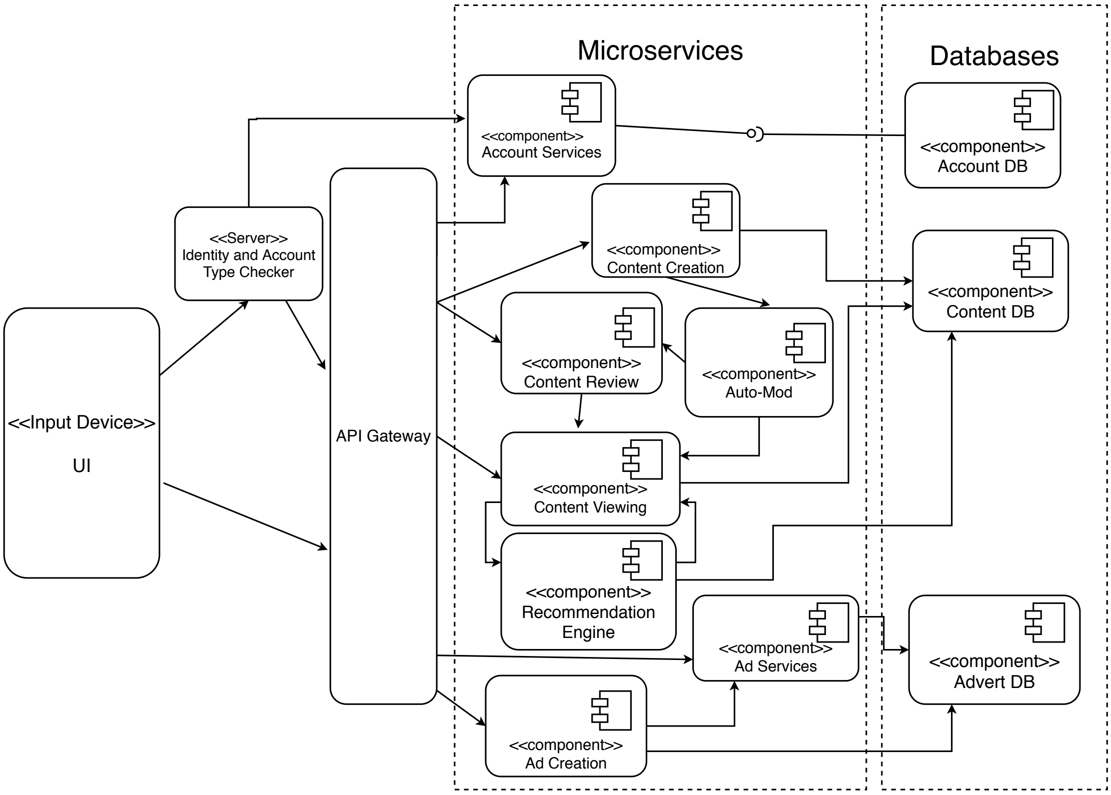

Figure 10: Architecture 2: Component Diagram

9.2.2 Deployment Diagram

Corresponding deployment diagram for figure 10 as above, for the proposed microservices architecture.

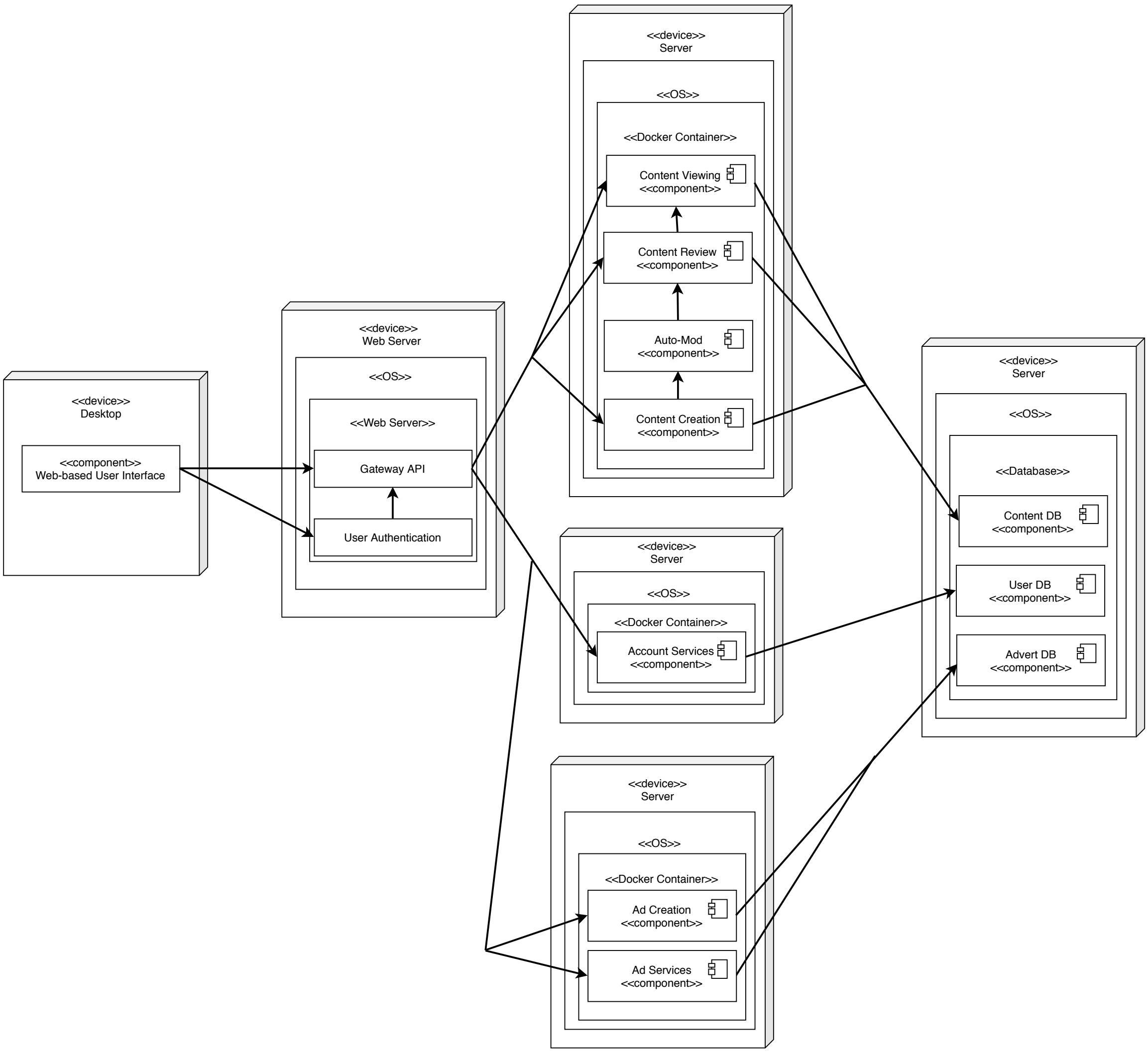

Figure 11: Architecture 2: Deployment Diagram

9.3 Architectural Tradeoff Analysis

In this part we will discuss and compare architecture designs that would best fit the Videosharetube system. First, we look at the quality attributes that define the system and then using the ATAM (architectural analysis method) we use these attributes to compare against the two architectures proposed.

Quality Attributes

The three facets of the system that we deemed to be most significant were performance, reliability, and scalability. Performance was the most significant factor as the core of the system relates to a large number of video content viewing and browsing, meaning that this would require fast responses from the system to allow for users to have a fluid experience. Scalability was also key as the system, relying heavily on user traffic and generated content, can grow and change - meaning that the system must be flexible in multiple ways to accommodate this. If the system could be scaled properly and quickly to meet changing demands, this would prevent a great deal of issues later down the line. Reliability tied in well with scalability, given the unpredictable nature of the system load and growth in content it would undoubtedly have issues with overloading, system failures and outages if not handled properly.

Performance

Users expect the system to respond quickly to each request they make; whether that is loading a video, browsing new posts, or writing a comment. If the system has poor performance, users will be far more likely to abandon the platform and look for alternative services.

Scalability

The content of the system is user generated and subject to exponential growth under certain circumstances. Each user can generate content, then as popularity increases more users will join and create further content. This in turn will attract more businesses to create adverts for the platform to capitalise on the high amounts of traffic and video companies will be attracted to the platform for similar reasons. In order to deal with this sudden growth and changes in traffic / content and number of users, the system architecture must allow for scalability – the ability to change the size and scope of the system to meet changing demands.

Reliability

With a large system, it is inevitable that the system will break at some point. A reason for this could be hardware failure, power outages, or system overload. All of these are impossible to avoid for a product, and so the system's reliability must be accounted for in 2 key ways:

- Can the users access the system at all times?
- Is data loss minimised in the event of system failure?

In addition, the likelihood of any of these crashes should be minimized where possible, and ideally the user will not notice. In an ideal world, the system will perform consistently for the users, both with quick page load times and with no unexpected results.

Candidate Architectures and Comparison

We have considered two different architectures for the platform. In terms of performance, each of these has its advantages and drawbacks.

At a high level, the way a client connects to the system is different depending on the architecture choice:
See figures 8 & 10.

Three-tier: Single point of contact for the client (Business Layer), which handles all requests and passes them back to the user.

Microservices: Client connects through the API gateway, which refers the request to the relevant part(s) of the system, which each in turn pass this back to the user.

Performance

Each server on which part of the architecture runs will have its own resources and limitations in terms of processing power, network connection speed, etc. With microservices, one part of the system (e.g. Content Viewing) could be operating at full capacity, while another part (e.g. Ad Creation) may have a lot of spare capacity. Since these are containerized, it is not possible to pool the resources and share them between services. This affects performance, since a content viewing service operating at full server capacity will start to see reduced performance, and the time taken to serve a user will increase.

With three-tier architecture, on the other hand, all Video-Providing Logic can run on the same server. Resources can therefore be pooled together - such that one part of the system being at low usage leaves system resources available for a busier part of the system. This flexibility leads to increased performance.

This consideration aside, it is worth noting that a microservices architecture is composed of many smaller elements, rather than one large system. This makes it more dynamic, and if utilised correctly this attribute could be used to improve performance.

This links to the scalability of the platform, which we have analysed separately. Modern cloud-hosting services such as AWS allow companies to easily create (spin up) new servers on demand. Such a service could be used to increase capacity in parts of the system where it is required, with only a short delay for the capacity to go live. This is possible since each server in the microservices architecture is relatively small, compared to the larger single server necessary to operate a three-tier architecture.

Considering both architectures side-by-side, a key benefit of Three-tier is the ability for resources to be shared between different parts of the system. Microservices does not pool resources in this way, however the dynamic nature of the system and the flexibility which results from this allows for the system to be optimised in such a way that this architecture is likely to offer the best possible performance overall.

Scalability

There are three types of scalability that relate strongly to the current system: administrative scalability, increasing the number of users; functional scalability, increasing the scope of specific system functions without debilitating general functionality; and load scalability, a systems ability to react to varying levels of traffic and / or loads. The three-tier architecture favours a strict division between client, business and database layers, therefore this will help across general scalability of the application. For example, the user database can be altered in isolation without affecting the account services on the business or client layers, because they are separated in design. Thus, increasing the capacity, functionality and size of specific components in a system would be made easier as opposed to having all systems components intertwined into the same location.

However, a microservice design would allow for even greater subdivision amongst the processes within an application. In short, this means that it would allow for greater specificity when looking to upscale specific parts of the application. For example, if the recommendation engine needed to become more complex and upscaled to incorporate more complex datasets and outputs as a consequence of higher user traffic, this could be done easily as it is separate from the rest of the system in the design. Furthermore, if necessary, the entire service could be taken offline for maintenance without disrupting the other business services surrounding it, unlike the three-tier design which would require the whole business layer to be shut down.

Another benefit of this subdivision between services in a microservice architecture, is that components can be reused in other parts of the system easily. This is because each service will be able to work independently of its surroundings and could be integrated into a 3rd party much easier when needed to increase other aspects of the system. Be it for functional, administrative, or load scalability, the ability to reuse useful services throughout the system will help speed up the process in scaling a product. Based on the greater ability to compartmentalize services and functionality, it is clear the microservice architecture allows for greater flexibility when scaling the system across all facets of scalability discussed.

Reliability

Reliability is a strong point of microservices, due to services being split. In the event of one service failing, the product as a whole may continue to work at only a minor cost to performance while the service is fixed. This does not completely guarantee a lack of failure, however. If one service is having issues, other services that depend on it may develop their own issues as a result. This can be time-consuming to debug, and wastes developer resources. Data loss is prevented easily by holding data at several copies of the microservice architecture, and so in the event of failure data can be copied from another server easily.

Three-tier systems have no advantage for reliability, due to the centralised nature of the 3 tiers. This is due to the single point of contact in the business layer. In the event that a developer pushes a buggy update, the entire system may act in an unintended manner. This centralised nature also means that system-wide failure is much more likely, and so measures to prevent or minimize the effect must be carefully considered while designing the system to avoid an event where no users are able to access the system.

After consideration, from a reliability standpoint microservices are the clear winner. With additional planning and preventative measures, three-tier is still a suitable method however requires significantly more work.

Conclusion

Over the three quality attributes discussed, a microservices architecture design was unanimously found to be the better option for the Videosharetube system. The segmentation of microservice functionality allows for more dynamic and flexible options for the system which lends itself well to performance, scalability, reliability demands. The three-tiered design did have its benefits, such as the ability to pool resources and share them amongst the same layer, but the microservice design showed further benefits that outweigh this shortcoming.

10 MSc Only

10.1 Digital Addiction and Gamification

Social media and video platforms have attracted a great deal of interest over the past few decades, growing to millions of users in just a few years. Many researchers have been interested in the addictive nature of these websites and content generation; in particular, video sharing platforms like YouTube that offer a continuous stream of videos and content have been seen to potentially lead to addictive behaviours in users [1], [2]. This was found using the Usage and Gratification Theory, which is an audience based model looking at why people seek out media. Specifically, the theory breaks this motivation into separate facets and this allows researchers to look at how these motivations affect addictive behaviour [3] [4]. For example, when users go onto sites such as YouTube, they are seeking specific gratifications such as education, entertainment, or escapism.

In an empirical study [5], Klobas saw that users going onto YouTube for entertainment were more likely to demonstrate compulsive watching behaviours than those seeking information or education. The user-generated aspect of the system in YouTube and Videosharetube also lends to the addictive model, as this creates a communal system within the platform and users are further motivated by interpersonal interactions and reputation [2] [6]. The habitual behaviour to continuous use of a platform is referred to in some studies as stickiness and is largely influenced by how gratifications interplay with the availability and accessibility of the media [7], [8]. While it is unfeasible to filter for the motivations users feel when seeking out the Videosharetube platform, it is possible to curb this addictive reaction by altering the accessibility of the platform implicitly and thus disrupting the continuous stream of content [1] [7].

Even though addiction models are rooted in behavioural stimulus-reward models, perhaps the best solution to this lies in the same idea. Gamification takes advantage of the same principles that led to the current problem, using already present or manufactured motivations to create behavioural change. Looking at the health applications that try to curb negative behaviours (such as smoking), the results are promising, if a little premature [9] [10]. However, making the users aware of their viewing habits and incentivising them to curb compulsive watching establishes a feedback loop that has shown to provide significant changes in behavioural interventions [11]. Therefore, this form of intervention was used to tackle addiction to the platform as it will decrease addictive behaviours in users without directly affecting the content or user traffic that is necessary to the system.

10.2 Compliance Requirements

1. Videosharetube MUST operate within its specific purpose (sharing videos online among adult users).
2. Video content MUST be in line within the law that the user is viewing it in.
3. The system MUST use gamification to reduce digital addiction to Videosharetube.

10.3 Monitoring Requirements

1. Videosharetube system MUST record each Customer Users time spent on the platform.
2. The system MUST notify the user when they are spending too long on the platform.
3. The system must keep track of a user's points.
4. Gamification techniques MUST overall reduce addiction to the platform.
5. Customer Users MUST generate points within the app to redeem for free use of the platform.
6. Customer Users SHOULD also be able to complete educational challenges, such as watching a series on addiction, to accumulate more points.
7. Customer Users MUST be able to sign up for a daily email which educates about addiction to online platforms for extra points.

10.4 Trade-Offs

- The less addiction that Videosharetube promotes for its platform, the ad spaces become less valuable, as the incentivisation not being on the platform through gamification causes less user time online.
- Videosharetube will be promoting a good relationship with online platforms, promoting the brand.
- The use of gamification will encourage users to monitor their usage of Videosharetube. On the other hand, depending on the implementation this could work the other way, causing an overall increase in time spent online per user.
- Examples of gamification could be through the use of day streaks of spending (for example) an hour or less a day on consecutive days, which could result in free use of the platform as opposed to the normal £1 an hour. Streaks would generate points to redeem for time credit.

10.5 Sequence Diagrams

The points system shows the points the user have earned whilst being logged off and they can then redeem these points for free premium viewing. If they do not have enough points they get an error message.

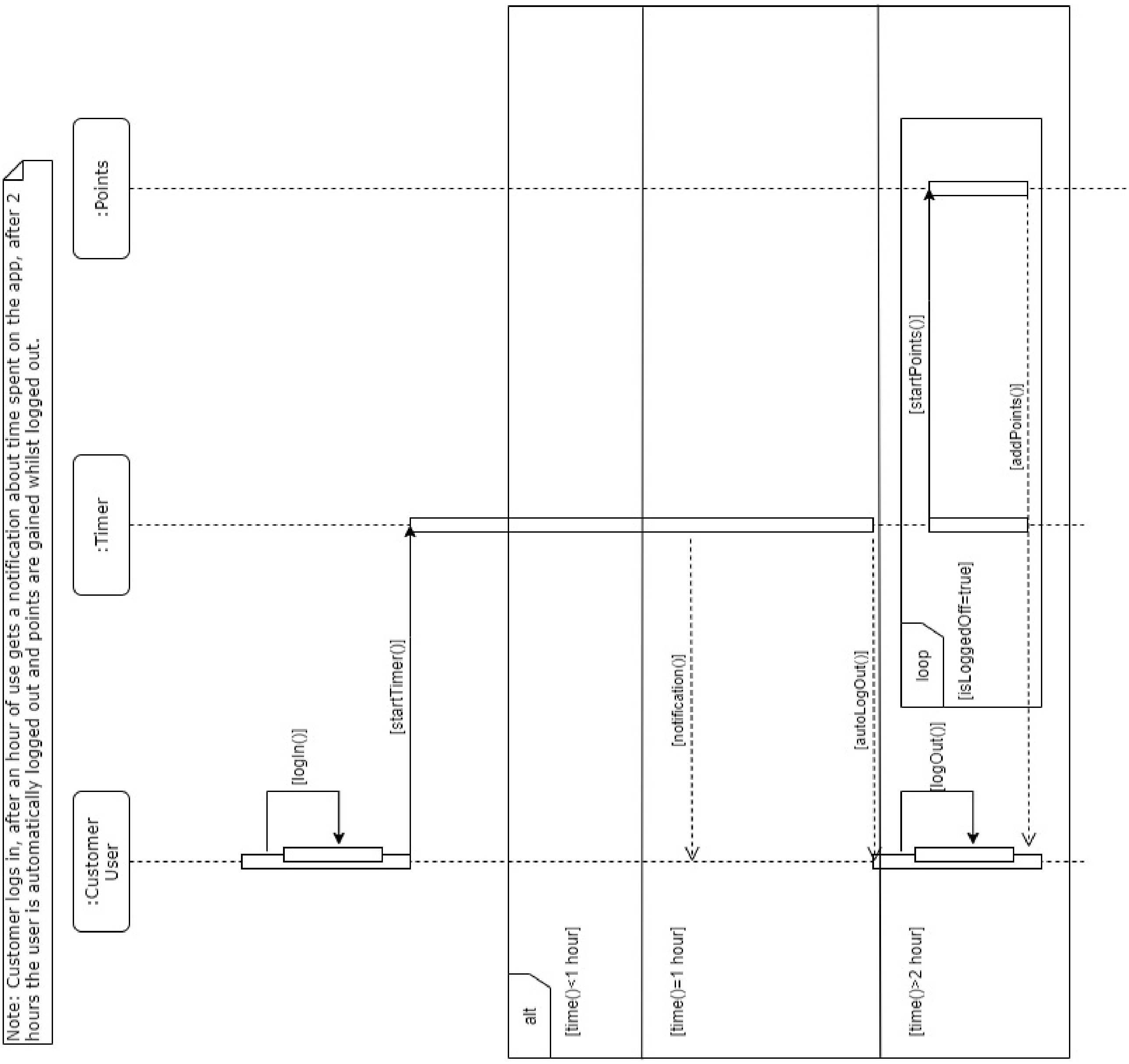

Figure 12: Compliance Sequence Diagram

10.6 State Diagrams

The points system shows the points the user have earned whilst being logged off and they can then redeem these points for free premium viewing. If they do not have enough points they get an error message.

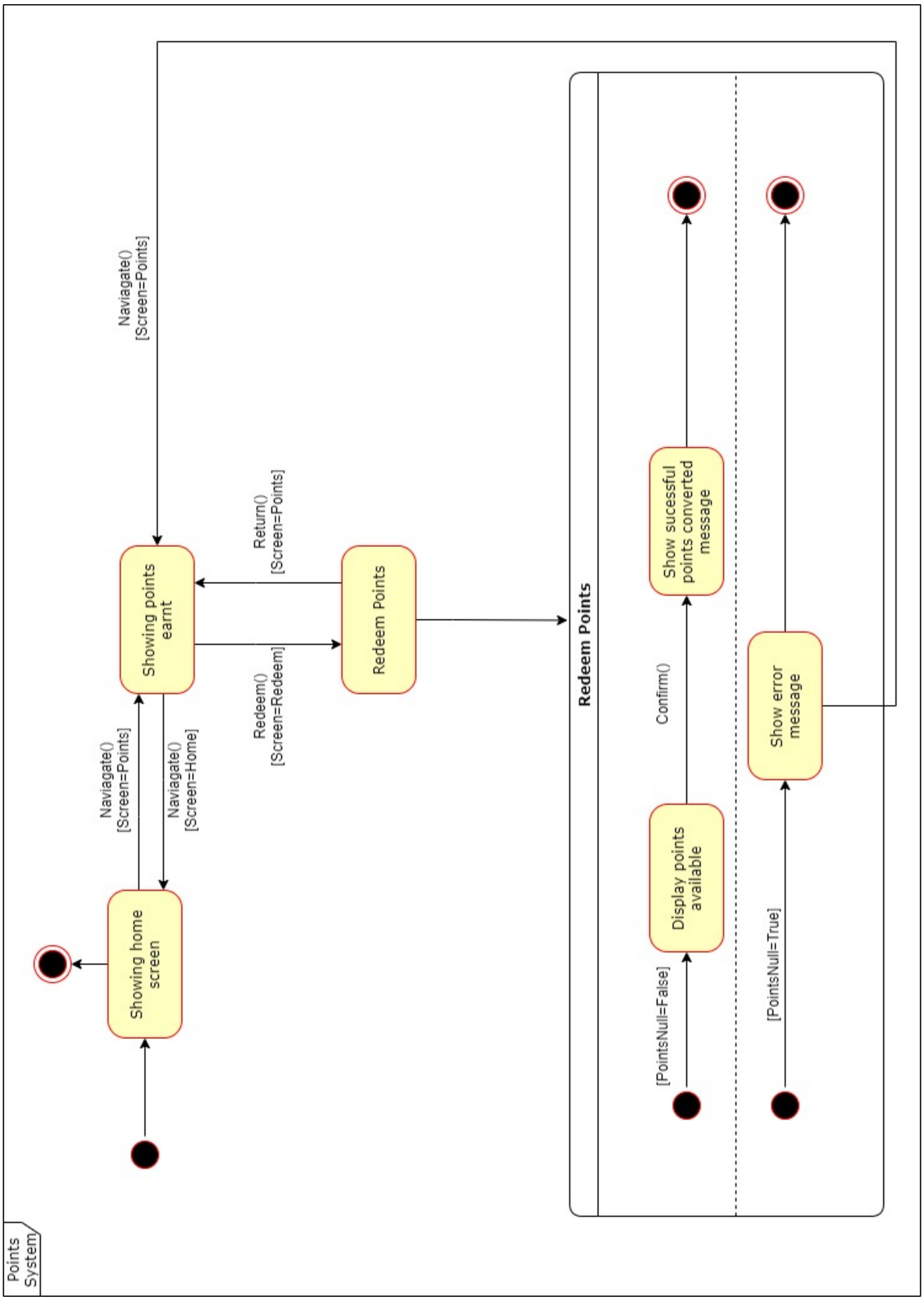

Figure 13: Compliance State Diagram

The timer system starts timing when the user logs off and adds points for every minute they are logged off. Once they log in, the system displays the points earned and also starts a new timer. If the user is on for more than an hour, they get a warning notification. If the user is on for two hours they are automatically logged off and the timer starts again.

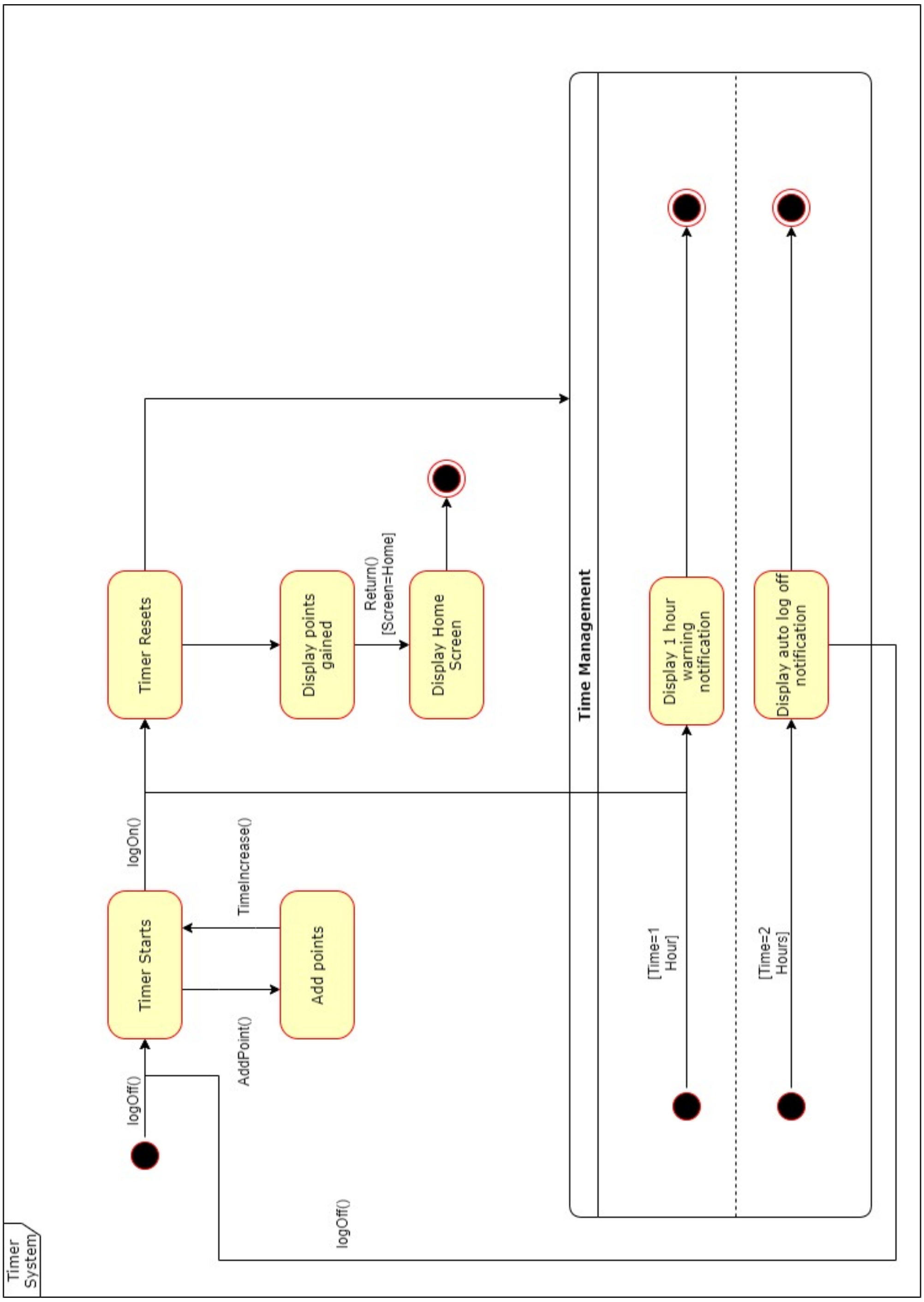

Figure 14: Compliance State Diagram

10.7 Class Analysis Discussion

The addition of new requirements to monitor and prevent digital addiction creates further problems for the system. The system now incorporates more classes to fulfil these new requirements. The system now monitors a user's time spent on the platform, giving points to users for time not spent on the platform and allows these points to be exchanged for free use of a video company's content. The system now also creates educational challenges and allows users to be emailed about addiction.

The points system creates much more interconnection on the system as it is linked to the time analysis, educational challenges and the redeem points functions. Furthermore, the redeem points class also connects with users to change their account balance, as does the time monitoring class.

These new classes increase coupling in the system, both between themselves and with the user. The new classes also affect the cohesion of the system as the points system is less focused and therefore less tight. In order to reduce these adverse effects on the system, we could use an interface for the monitoring system to achieve abstraction and reduce the coupling. This will help to reduce the maintenance cost of the system modification by reducing the number of connections and will also help to allow for future maintenance and scalability at a reduced cost.

10.8 Architecture Discussion

Considering figure 10, it was previously discussed how a microservices based architecture was preferential for this system to a classic three-tier design. Performance, scalability and reliability were the main attributes that were used to assess whether each architecture was appropriate for Videosharetube.

The above solution for gamification and addiction aversion for the video sharing platform included explicitly giving accounts/users redeemable points in return for premium viewership on the platform. This would translate into free services for the user in return for a healthy relationship with the platform. Furthermore, users would be notified on-screen by the platform, prompted to take a break, once they had been active for one continuous hour. This tactic is similar to how Nintendo manage digital addiction, with prompts on games such as Wii Sports and Wii Play.

With respect to architectural decisions and design for Videosharetube, since a microservices design (figure 10) was chosen a Monitoring Engine module could be appended to the microservices system. This module would need to communicate with Account Services, Content Viewing and potentially Recommendation Engine.

Considering the primary non-functional requirements that were used in the ATAM section, they will be re-evaluated with the new monitoring module considered in the microservices architecture.

Performance

There are trade-offs for performance in monitoring users activity as it will require more server processing power to keep track of the extra information such as: time logged in, points on account, time elapsed since last login. This cost would come with the benefit of managing users addiction and any cost could be managed over time as its requirements (processing or space) would scale linearly with number of users as the implementation is relatively simple.

Scalability

Monitoring would offer some benefits for load scaling as the number of addicted users would be reduced and therefore not be putting the hardware and systems under extra strain. On the other hand, monitoring would be changing the fundamental attitude towards Videosharetube as revenue is generated from subscriptions and adverts. The generation of profit is a primary financial requirement and monitoring users to reduce addiction would reduce the overall revenue of ads. Therefore, there would need to be a balance between the two to find a sweet spot between administrative scaling and monitoring users to maximise both growth and minimise addiction to the platform.

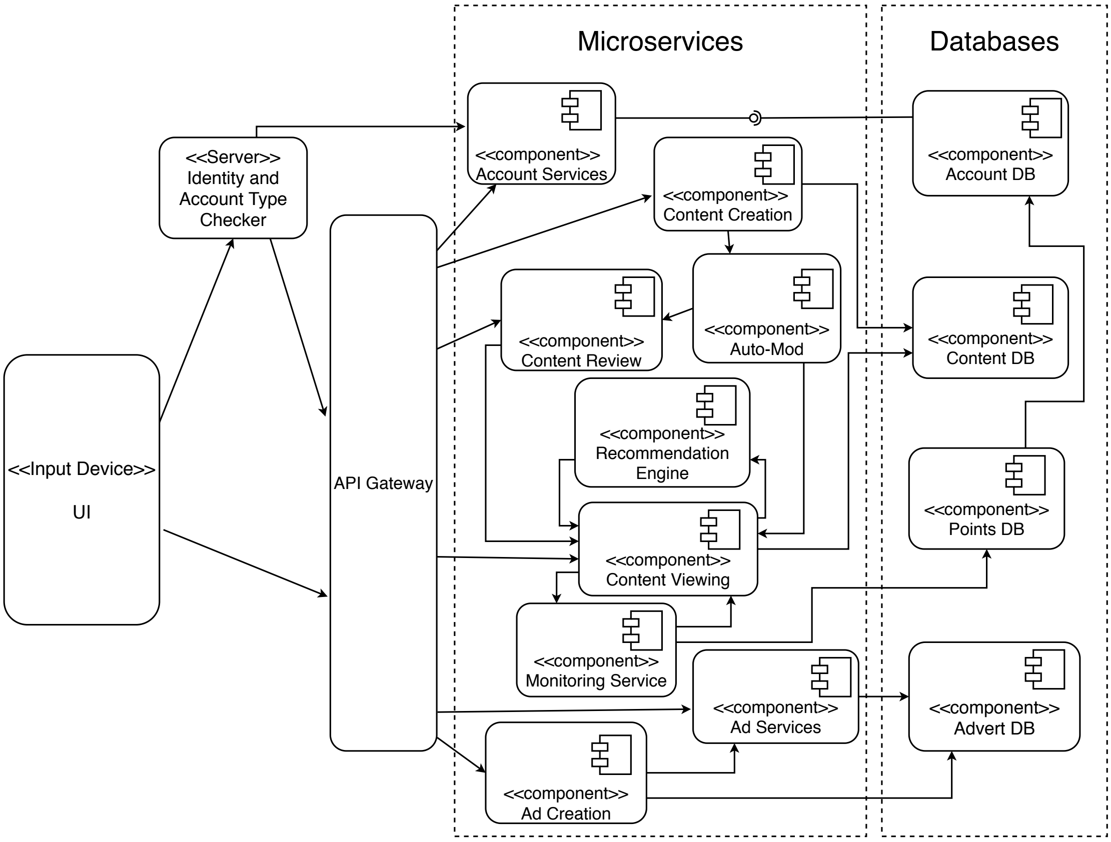
Microservices
Databases
<<Server>> Identity and Account Type Checker
<<Input Device>> UI
API Gateway
<<component>> Account Services
<<component>> Content Creation
<<component>> Auto-Mod
<<component>> Content Review
<<component>> Recommendation Engine
<<component>> Content Viewing
<<component>> Monitoring Service
<<component>> Ad Services
<<component>> Ad Creation
<<component>> Account DB
<<component>> Content DB
<<component>> Points DB
<<component>> Advert DB

Figure 15: Compliance State Diagram

Reliability

Given that microservices was elected as the preferential architectural design, reliability is not an issue for the addition of the monitoring system. This is due to the compartmentalisation of services in the design, where other services such as video viewing or video creation can continue without interruption even if the monitoring system fails. Furthermore, since the monitoring system comes later in the control flow than more primary systems like video viewing, there is a very narrow scope for the overall service to be become dysfunctional should the monitoring system fail. One example could be that the number of account points is irretrievable and therefore a user may be unable to redeem them for premium videos, but scenario will not cripple the platform function as a whole.

11 Appendix

Agenda with Minutes (Meeting 1, Week 4)

1. Date & Time Of Meeting
   - Friday 7th February 2020, 12pm-2pm
   - Location: Library meeting room 149
2. Attendance
   - All
3. Key Discussion Points
   - Set date and time for regular meetings
   - Workspace – set up Google drive?
   - Use Overleaf for final report
   - Could set our own deadlines for completing parts of the project
   - Deadline: Friday 27th March (week 11) -¿ gives us 7 weeks time to work on project
4. Decisions
   - Weekly meetings to be Fridays 12-2pm
   - Use Google drive initially for organising files - set up during meeting:
   - Overleaf seems like good idea for final report, as long as not painful to set up
   - Aim to complete 20
5. Actions
   - SIMON: Q1. Write summary, detailing assumptions made on scope and take into account views of stakeholders (600 words)
   - DEVON: Q2. State the functional and non-functional requirements of our system
   - DEVON: Set up Overleaf for report
   - ANGUS, BEN, JOSH: Q3. Document analysis and design of system using UML for: Q3.1. Use case diagram; Q3.2. Non-trivial use cases for the diagram

Agenda with Minutes (Meeting 2, Week 5)

1. Date & Time Of Meeting
   - Friday 14th February 2020, 1pm-2pm
   - Location: Library meeting room 106
2. Attendance
   - All
3. Key Discussion Points
   - Action points from last week
   - How is workload so far
   - Tasks for upcoming week
   - Next meeting same time?
4. Actions

- SIMON: Finalise summary doc
- SIMON + DEVON: Compare summary and funct/non-funct requirements -¿ can potentially restrict project spec to remove unnecessary work
- BEN, ANGUS, JOSH: Finish up use case diagram, add lines/connectors
- ANGUS: Final non-trivial use case diagram
- ANGUS: Q3.3: Object diagram
- JOSH: Q3.5: Activity diagram
- BEN, DEVON, SIMON: Q3.4: Class analysis

Agenda with Minutes (Meeting 3, Week 6)

1. Date & Time Of Meeting
   - Friday 21st February 2020, 1pm-2pm
   - Location: Library meeting room 106
2. Attendance
   - All
3. Key Discussion Points
   - Action points from last week
   - How is workload so far
   - Tasks for upcoming week
   - Next meeting same time?
4. Actions
   - Finishing up from last week:
   - SIMON + DEVON: 2nd version of class diagram
   - DEVON: CRC
   - SIMON: Finalise summary doc
   - New for this week:
   - DEVON: Update latex: - 3rd use case, updated activity diagram
   - DEVON: Sequence diagram
   - JOSH: State diagram
   - BEN, ANGUS, SIMON: Component Diagram and a Deployment Diagram for the two candidate architectures

Agenda with Minutes (Meeting 4, Week 7)

1. Date & Time Of Meeting
   - Friday 28th February 2020, 1pm-2pm
   - Location: Library meeting room 222
2. Attendance
   - All but Josh (absent due to illness)
3. Key Discussion Points

- Action points from last week
- How is workload so far
- Tasks for upcoming week

4. Decisions

   - Larger chunks of work for next sections of project, so agreed to assign tasks into subgroups to be completed over next 2 weeks

5. Actions

   - Larger tasks, so assigning these for following 2 weeks:
   - BEN, ANGUS, SIMON: Component Diagram and a Deployment Diagram for the two candidate architectures -¿ meeting Mon 15:15-17:00
   - BEN, ANGUS, SIMON: Evaluate using ATAM
   - DEVON, JOSH: MSc Advanced Questions

Agenda with Minutes (Meeting 5, Week 9)

1. Date & Time Of Meeting

   - Friday 28th February 2020, 1pm-2pm
   - Location: Library meeting room 222

2. Attendance

   - All

3. Key Discussion Points

   - Action points from last week
   - How is workload so far
   - Tasks for upcoming week

4. Actions

   - ANGUS: re-work/expand user scenarios
   - BEN: update noun/verb analysis to include Recommendation Engine
   - SIMON: (once Ben updates noun/verb analysis) CRC card for Recommendation Engine
   - ANGUS, BEN, SIMON: perform ATAM analysis
   - DEVON: re-visit architecture/component diagram
   - JOSH: last 2 bullet points under ‘advanced questions’

Agenda with Minutes (Meeting 6, Week 10)

1. Date & Time Of Meeting

   - Friday 20th March 4-5pm
   - Location: Online video conference

2. Attendance

   - All

3. Key Discussion Points

- Coordinating work during covid-19 and working remotely
- Completing final sections of projects
- Uploading Component Deployment Diagrams to Overleaf

4. Decisions

   - Assign tasks for each team member to work on remotely

5. Actions

   - ANGUS: re-work/expand user case diagram
   - BEN: update noun/verb analysis to include Recommendation Engine
   - SIMON: (once Ben updates noun/verb analysis) CRC card for Recommendation Engine
   - ANGUS, BEN, SIMON: perform ATAM analysis
   - DEVON: re-visit architecture/component diagram

Agenda with Minutes (Meeting 7, Week 11)

1. Date & Time Of Meeting

   - Thursday 26th March 3-4pm
   - Location: Online video conference

2. Attendance

   - All

3. Key Discussion Points

   - Final steps to complete report

4. Decisions

   - Each look over a share of the report. Compare it to the project spec and check it meets the requirements.

5. Actions

   - Assigned sections for review:
   - SIMON: Use Case Modelling (p.6-9), Object Diagram (p.17), Previous Meeting Minutes
   - JOSH: Class Analysis (p.11-16)
   - ANGUS: Scope, requirements, Activity diagram, Sequence diagram, State diagram
   - DEVON: Architecture and Deployment (p.20-26)
   - BEN: MSc Section (p.27-31)

References

[1] P. Haridakis and G. Hanson, "Social interaction and co-viewing with youtube: Blending mass communication reception and social connection," *Journal of Broadcasting & Electronic Media*, vol. 53, no. 2, pp. 317–335, 2009.

[2] J. Balakrishnan and M. D. Griffiths, "Social media addiction: What is the role of content in youtube?" *Journal of Behavioral Addictions*, vol. 6, no. 3, pp. 364–377, 2017.

[3] J. Raacke and J. Bonds-Raacke, "Myspace and facebook: Applying the uses and gratifications theory to exploring friend-networking sites," *Cyberpsychology & behavior*, vol. 11, no. 2, pp. 169–174, 2008.

[4] G. Shao, "Understanding the appeal of user-generated media: a uses and gratification perspective," *Internet research*, 2009.

[5] J. E. Klobas, T. J. McGill, S. Moghavvemi, and T. Paramanathan, "Compulsive youtube usage: A comparison of use motivation and personality effects," *Computers in Human Behavior*, vol. 87, pp. 129–139, 2018.

[6] C. L. Lortie and M. J. Guitton, "Internet addiction assessment tools: Dimensional structure and methodological status," *Addiction*, vol. 108, no. 7, pp. 1207–1216, 2013.

[7] J.-H. Wu, S.-C. Wang, and H.-H. Tsai, "Falling in love with online games: The uses and gratifications perspective," *Computers in Human Behavior*, vol. 26, no. 6, pp. 1862–1871, 2010.

[8] H.-S. Chiang and K.-L. Hsiao, "Youtube stickiness: the needs, personal, and environmental perspective," *Internet Research*, 2015.

[9] D. Johnson, S. Deterding, K.-A. Kuhn, A. Staneva, S. Stoyanov, and L. Hides, "Gamification for health and wellbeing: A systematic review of the literature," *Internet interventions*, vol. 6, pp. 89–106, 2016.

[10] E. A. Edwards, J. Lumsden, C. Rivas, L. Steed, L. Edwards, A. Thiyagarajan, R. Sohanpal, H. Caton, C. Griffiths, M. Munafò *et al.*, "Gamification for health promotion: systematic review of behaviour change techniques in smartphone apps," *BMJ open*, vol. 6, no. 10, p. e012447, 2016.

[11] C. J. Greaves, K. E. Sheppard, C. Abraham, W. Hardeman, M. Roden, P. H. Evans, and P. Schwarz, "Systematic review of reviews of intervention components associated with increased effectiveness in dietary and physical activity interventions," *BMC public health*, vol. 11, no. 1, pp. 1–12, 2011.