

REPORT

On

Case Study: Snapchat

of

Software Engineering

(CS330)

By

Panth Savant (22000400) & Pearl Agnihotri (22000405)

Third Year, Semester 5

Course Incharge: Prof. Darshan Parmar



Department of Computer Science and Engineering

School of Engineering and Technology

Navrachana University, Vadodara

Autumn Semester

(2024 – 2025)

Table of Contents

I Project Description	01
1. Project Overview	01
2. The Purpose of the Project	01
2a. The User Business or Background	01
2b. Goals of Project	02
2c. Measurement	03
3. The Scope of the Work	04
3a. The Current Situation	04
3b. The Context of the Work	05
3c. Work Partitioning	07
3d. Competing Products	09
4. Product Scenarios	10
4a. Product Scenario List	10
4b. Individual Product Scenarios	10
5. Stakeholders	11
5a. The Client	11
5b. The Customer	12
5c. Hands-On Users of the Product	12
5d. Priorities Assigned to Users	14
5e. User Participation	15
5f. Maintenance Users and Service Technicians	16
5g. Other Stakeholders	16
6. Mandated Constraints	18
6a. Solution Constraints	18
6b. Implementation Environment of the Current System	19
6c. Partner or Collaborative Applications	21

6d. Off-the-Shelf Software	22
6e. Anticipated Workplace Environment	23
6f. Schedule Constraints	24
6g. Budget Constraints	25
7. Naming Convention and Definitions	26
7a. Definitions of Key Terms	25
7b. UML and Other Notation Used in This Document	27
7c. Data Dictionary for Any Included Models	28
8. Relevant Facts and Assumptions	29
8a. Facts	29
8b. Assumptions	30
II Requirements	34
9. Product Use Cases	34
9a. Use Case Diagrams	34
9b. Product Use Case List	39
9c. Individual Product Use Cases	39
10. Functional Requirements	51
11. Data Requirements	56
12. Performance Requirements	58
12a. Speed and Latency Requirements	58
12b. Precision or Accuracy Requirements	61
12c. Capacity Requirements	63
13. Dependability Requirements	64
13a. Reliability Requirements	64
13b. Availability Requirements	65
13c. Robustness or Fault-Tolerance Requirements	66
13d. Safety-Critical Requirements	67

14. Maintainability and Supportability Requirements	69
14a. Maintenance Requirements	69
14b. Supportability Requirements	70
14c. Adaptability Requirements	71
14d. Scalability or Extensibility Requirements	72
14e. Longevity Requirements	73
15. Security Requirements	75
15a. Access Requirements	75
15b. Integrity Requirements	76
15c. Privacy Requirements	77
15d. Audit Requirements	78
15e. Immunity Requirements	79
16. Usability and Humanity Requirements	79
16a. Ease of Use Requirements	79
16b. Personalization and Internationalization Requirements	81
16c. Learning Requirements	82
16d. Understandability and Politeness Requirements	84
16e. Accessibility Requirements	85
16f. User Documentation Requirements	86
16g. Training Requirements	87
17. Look and Feel Requirements	87
17a. Appearance Requirements	87
17b. Style Requirements	88
18. Operational and Environmental Requirements	89
18a. Expected Physical Environment	89
18b. Requirements for Interfacing with Adjacent Systems	90
18c. Productization Requirements	91

18d. Release Requirements	92
19. Cultural and Political Requirements	93
19a. Cultural Requirements	93
19b. Political Requirements	93
20. Legal Requirements	94
20a. Compliance Requirements	94
20b. Standards Requirements	94
III Design	95
21. System Design	95
21a. Design Goals	95
22. Current Software Architecture	96
23. Proposed Software Architecture	103
23a. Overview	103
23b. Class Diagrams	105
23c. Dynamic Model	111
23d. Subsystem Decomposition	113
23e. Hardware / Software Mapping	115
23f. Data Dictionary	122
23g. Persistent Data Management	124
23h. Access Control and Security	126
23i. Global Software Control	129
23j. Boundary Conditions	131
24. Subsystem Services	133
25. User Interface	136
26. Object Design	138
26a. Object Design Trade-Offs	138
26b. Interface Documentation guidelines	141

26c. Packages	144
26d. Class Interfaces	146
IV Test Plans	149
27. Features to be tested/ not to be tested	149
28. Pass/ Fail Criteria	156
29. Approach	162
30. Suspension and Resumption	164
31. Testing Materials (Hardware/ Software Requirements)	166
32. Test Cases	168
33. Testing Schedule	173
V Project Issues	176
34. Open Issues	176
35. Off-the-Shelf Solutions	177
35a. Ready-Made Products	177
35b. Reusable Components	179
35c. Products That Can Be Copied	180
36. New Problems	181
36a. Effects on the Current Environment	181
36b. Effects on the Installed Systems	184
36c. Potential User Problems	185
36d. Limitations in the Anticipated Implementation Environment That May Inhibit the New Product	187
36e. Follow-Up Problems	189
37. Tasks	190
37a. Project Planning	190
37b. Planning of the Development Phases	192
38. Migration to the New Product	194
38a. Requirements for Migration to the New Product	194

38b. Data That Has to Be Modified or Translated for the New System	196
39. Risks	197
40. Costs	198
41. Waiting Room	201
42. Ideas for Solutions	202
43. Project Retrospective	203
VI Glossary	205
VII References / Bibliography	206

List of Figures

Figure 1: Context Diagram of Snapchat App.....	06
Figure 2: Use Case Diagram of Module 1: Authentication	34
Figure 3: Use Case Diagram of Module 2: Media Capturing and Editing.....	35
Figure 4: Use Case Diagram of Module 3: Sends a Snap.....	36
Figure 5: Use Case Diagram of Module 4: Snap Storage and Management.....	37
Figure 6: Use Case Diagram of Module 5: Chat and Messaging.....	38
Figure 7: Activity Diagram of Module 1: Authentication.....	46
Figure 8: Activity Diagram of Module 2: Media Capturing and Editing.....	47
Figure 9: Activity Diagram of Module 3: Sends a Snap.....	48
Figure 10: Activity Diagram of Module 4: Snap Storage and Management.....	49
Figure 11: Activity Diagram of Module 5: Chat and Messaging.....	50
Figure 12: Entity Relationship (ER) Diagram of Primary Functionality of Snapchat.....	99
Figure 13: Data Flow Diagram (DFD) (Level 0) of Primary Functionality of Snapchat.....	100
Figure 14: Data Flow Diagram (DFD) (Level 1) of Primary Functionality of Snapchat.....	101
Figure 15: Data Flow Diagram (DFD) (Level 2) of Primary Functionality of Snapchat.....	102
Figure 16: Class Diagram of Module 1: Authentication.....	107
Figure 17: Class Diagram of Module 2: Media Capturing and Editing.....	108
Figure 18: Class Diagram of Module 3: Sends a Snap.....	109
Figure 19: Class Diagram of Module 4: Snap Storing and Management.....	110
Figure 20: Class Diagram of Module 5: Chat and Messaging.....	111
Figure 21: Sequence Diagram of Module 1: Authentication.....	118

Figure 22: Sequence Diagram of Module 2: Media Capturing and Editing.....	119
Figure 23: Sequence Diagram of Module 3: Sends a Snap.....	119
Figure 24: Sequence Diagram of Module 4: Snap Storage and Management.....	120
Figure 25: Sequence Diagram of Module 5: Chat and Messaging.....	121

I Project Description

1 Project Overview

Snapchat is a fun and interactive app that lets people share photos, videos, and messages that disappear after being seen. It became popular, especially among younger users, because of its unique feature of messages that vanish. Beyond just messaging, Snapchat allows users to share moments in their daily lives through Stories, which are photos and videos that stay up for 24 hours. The app is also known for its playful filters and lenses that can transform your look or add fun effects to your snaps. With Snapchat, users can also discover exciting content from creators, brands, and the community. It's all about being spontaneous, creative, and staying in the moment.

2 The Purpose of the Project

2a The User Business or Background of the Project Effort

Content

Snapchat is a widely recognized social media platform that primarily serves the purpose for exchanging videos and photographs with close friends and family. It is renowned for being ephemeral, meaning that the content often vanishes after a set amount of time—typically 24 hours.

Intended Work

Snapchat users hope to share intimate moments with friends and family in a less staged and more genuine way with the delivered product. Users are encouraged to be more impulsive and less concerned with maintaining a flawless online profile because of the content's fleeting nature. Snapchat's interactive features, such as Stories, Filters, and Augmented Reality (AR) lenses, allow businesses and content providers to connect with younger audiences while also promoting brand engagement and user participation in novel ways.

Motivation

- **Stress-Free Communication:** The motive of Snapchat's creation was to provide a less formal, less pressure to communicate flawlessly.
- **Ephemeral Content** Unlike the long-lived posts on other social media platforms, the emphasis was on sharing fleeting, unplanned experiences.
- **Special Purpose:** This driving force provided Snapchat with a distinct course, which helped it become successful and well-known.

Considerations

- **Privacy:** With the emphasis on disappearing content, protect user data.
- **UX:** Make sure the application remains easy to use and enjoyable.
- **Content Safety:** Effectively manage and screen out unwanted content.
- **Innovation:** To keep people interested, release new features on a regular basis.
- **Monetization:** Strike a balance between flawless user experience and adverts.
- **Global Adaptation:** Modify the application to accommodate many languages and cultures.
- **Competition:** Recognize and satisfy user demands to stay ahead of the game.
- **Regulation:** Adhere to international privacy laws and guidelines.

2b Goals of the Project

Content

In a world that is excessively polished and enduring, Snapchat wants to be a lighthearted, user-friendly program that allows users to share true, unplanned moments with friends in a private and authentic way.

Motivation

As Snapchat develops, there's a chance that new concepts and features could eclipse the platform's basic intention of encouraging sincere, impromptu communication. It's critical to maintain focus on the original goals and to often remind the team of them in order to avoid this. This guarantees that, despite its growth and evolution, the app will live up to its promise of genuine, private contacts.

Examples

Our goal is to ensure that individuals can effortlessly and rapidly share authentic, unadulterated memories with their peers. Even when we roll out new features and improvements, we want to support users in connecting in a way that feels real and private.

2c Measurement

We need to assess Snapchat's performance in a few crucial areas in order to decide whether it is succeeding in encouraging sincere and impromptu sharing. Initially, it is important to monitor user activity by seeing how frequently users post images and videos on the app or make use of features like Stories and Filters. This aids in our comprehension of whether consumers are actively engaging with the sharing feature that Snapchat seeks to offer.

Subsequently, tracking the quantity of active users on a daily and monthly basis will provide valuable information about the app's attractiveness and its ability to retain users over time. For evaluating the app's continued popularity, this is essential.

Another crucial element is user happiness. We can assess how well Snapchat is fulfilling user expectations and whether it is fostering the kind of genuine connections it claims by obtaining input through surveys and reviews.

Lastly, retention rates can show us if customers are still using the software on a daily basis. Retention rates are high, indicating that users find Snapchat's sharing and connecting strategies valuable and that the service is serving its intended function.

3 The Scope of the Work

3a The Current Situation

Content

Snapchat's business is all about making it easy and fun for people to share quick photos, videos, and messages. Right now, they use a mix of both manual and automated processes to keep everything running smoothly.

For the things they do manually, like content moderation and customer support, real people are involved. Moderators review any content that's been flagged by users to make sure it's appropriate, and support agents help out with any issues users might have, whether that's fixing technical problems or handling account concerns.

On the automated side, Snapchat uses technology to instantly deliver snaps, suggest content in the Discover section, and recommend friends. These automated systems make sure that everything is fast and feels spontaneous, which is a big part of what makes Snapchat fun.

If Snapchat introduces new features or updates, some of these processes might change. They could use more advanced AI to automatically catch inappropriate content, reducing the need for human moderators. They might also add AI tools to help with customer support, making it quicker for users to get help. And with better algorithms, Snapchat could make even smarter content recommendations, keeping users more engaged and entertained.

Overall, any new product developments would aim to make Snapchat more efficient, personalized, and fun for its users.

Motivation

The motivation behind this project is to improve some features of Snapchat which we found a gap to satisfy our needs by analyzing, researching and asking reviews or feedbacks among users. There is a need to add a feature to Snapchat that stops users from taking screenshots or recording the screen while viewing snaps. This would help keep user content more secure and ensure that their privacy is better protected.

3b The Context of the Work

Content

Snapchat is continually evolving to meet user expectations for privacy and security. As part of these efforts, there is a recognized need to address concerns about how user content is managed and protected. Currently, users can take screenshots or record their screens while viewing snaps, which poses a risk to the confidentiality of shared content.

- Improving Security
- Enhancing Privacy Policy

Motivation

The main motive behind these changes is to provide the user their privacy and relief for fear of Data leak. We're focusing on two main areas to make Snapchat better:

- **Improving Security:** We want to make sure the data is safe from hackers and other threats.
- **Enhancing Privacy Policy:** We're giving more control over one's information and making sure we follow the latest privacy laws.

Context Diagram of Snapchat

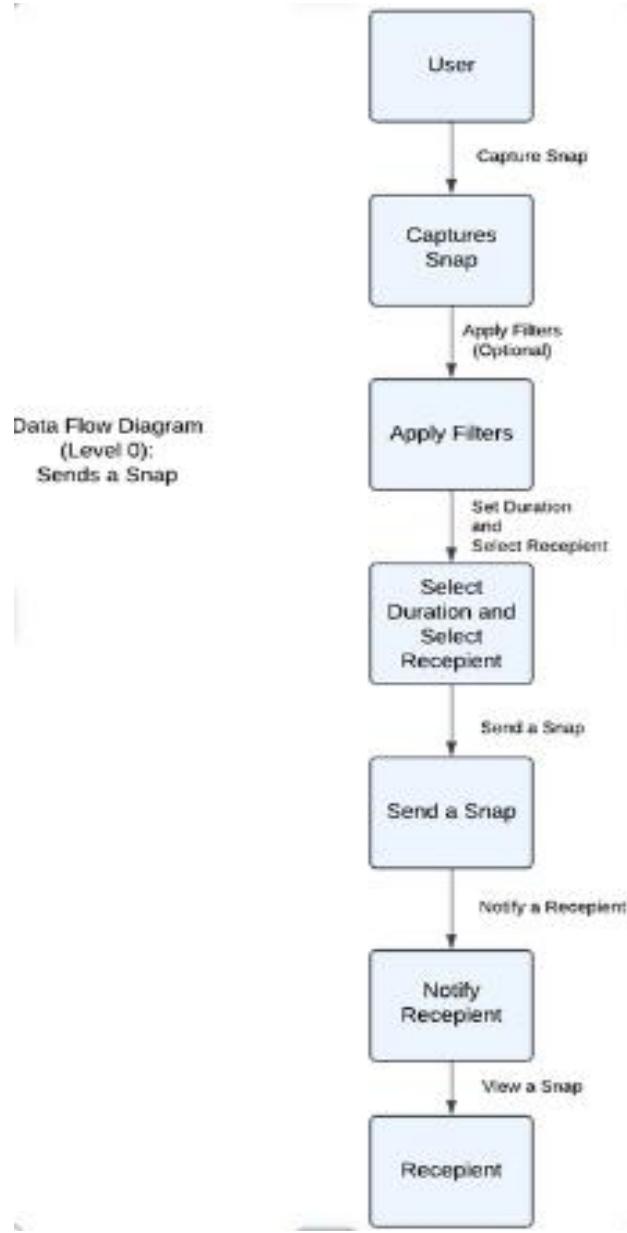


Figure 1: Context Diagram of Snapchat

(Website: <https://lucidchart.com>)

Considerations

1. Consistency with Naming Conventions:

All names used in the context diagram have been carefully matched to the naming conventions outlined in Section 5 of the data dictionary. For example, the term "User Profile" in the context diagram corresponds exactly to the defined term in the data dictionary, ensuring uniformity across documentation.

2. Alignment with Data Dictionary:

The context diagram adheres to the definitions specified in the data dictionary. Terms such as "Authentication Service" and "Data Storage System" are used as defined. This alignment ensures that the diagram accurately reflects the intended components and interactions.

3. Stakeholder Agreement:

The definitions and representations in the context diagram have been reviewed and approved by relevant stakeholders, including the project team and key users. All parties have confirmed that the diagram accurately represents the interfaces and data flows according to their understanding and needs.

3c Work Partitioning

1. Identify Key Actions:

- The important actions that the app needs to respond to, like when a user posts a snap or when there's a system update.
- **Examples:** A user sends a snap, someone logs in, the app detects a potential security issue.

2. Make a List of These Actions:

- **Event Name:** Name each key action or event, like "User Sends Snap" or "User Logs In."
- **Input:** Describe what information Snapchat gets when this event happens (like user data or a photo).

- **Output:** Describe what Snapchat does in response (like sending notifications or saving the snap).
- **Quick Summary:** Optionally, give a brief description of what happens during this event. For example, "When a user sends a snap, Snapchat updates their friends' stories and checks the content for safety."

3. Define What Each Action Means for Snapchat:

- **Concept:** Each event is like a mini job that Snapchat does. We need to understand what each job entails to improve the app.
- **Examples:** When a user posts a snap, Snapchat needs to save the photo, notify friends, and maybe check if the snap follows community guidelines.

4. Reason behind:

- **Purpose:** Breaking down these actions helps us understand what Snapchat needs to do better. It also makes it easier to figure out what new features to add or how to make the app more secure.
- **Goal:** To clearly identify the parts of Snapchat that need work and understand how to improve them.

5. Things to Keep in Mind:

- **Be Stringent:** Make sure all the actions and responses listed match what actually happens in Snapchat.
 - **Double-Check the Work:** Use this list to make sure we're not missing anything important and adjust if needed.
 - **Gathering Requirements:** Use these actions as a guide to figure out exactly what we need to build or improve to make Snapchat better for everyone.
-

3d Competing Products

Existing Alternatives

- **Instagram Stories:** Instagram lets people share photos and videos that disappear after 24 hours, with lots of filters and stickers. However, it doesn't focus as much on privacy or the spontaneous feel that Snapchat offers.
- **WhatsApp Status:** This feature is built into WhatsApp and lets users share disappearing updates with their contacts. It's simple but doesn't have Snapchat's fun and creative tools.
- **Tik-Tok:** Tik-Tok is all about short videos with cool effects, but it doesn't have disappearing messages or the same level of privacy that Snapchat does.

Motivation

These competitors do have some flaws, which are reflected and noticed, so the main reason of motivation are

- **Privacy Concerns:** Some of these platforms have issues with data security, which can make users hesitant to share personal moments.
- **Too Similar:** Many of these apps offer the same features, which can make them feel boring or repetitive.
- **Less Authentic:** Apps like Instagram and TikTok often promote polished, curated content, lacking the genuine, in-the-moment vibe of Snapchat.

Considerations

- Be Different: Snapchat can stand out by focusing on what makes it unique: privacy, disappearing messages, and a more genuine, real-time experience.
- Highlight Unique Features: By emphasizing its distinctive features like fun filters and the snap map, Snapchat can attract users looking for something fresh and exciting.

4 Product Scenarios

- **Scenario 1:** Sarah posts a picture of the sunset to Snapchat with her pals. She applies a filter and gets immediate feedback, so she doesn't worry about it being permanent.
- **Scenario 2:** Using an AR lens, John makes a humorous film that he shows to his friends and appreciates their quick, fleeting reactions to it.
- **Scenario 3:** Emma uses Snapchat to provide updates on her surprise party planning so that friends may follow along and subsequently reminisce about the enjoyable times.

4a Product Scenario List

- 1. Casual Sharing** - Sarah uses filters to share a sunset photo with friends and gets immediate feedback.
- 2. Spontaneous Moments** - John uses an AR lens to make and send a humorous movie, and he enjoys his friends' fleeting reactions.
- 3. Event Documentation** - Emma uses brief photos to inform friends on the party's preparations, preserving a momentous moment in time.

4b Individual Product Scenarios

- **Casual Sharing:** Sarah wants to spend her last moments at the beach with her pals as the sun sets. She opens Snapchat immediately, takes a picture of the scenery, applies a lighthearted filter, and shares it to her Story. Her pals see it right away and respond with remarks and emojis. Because the post is just temporary, Sarah doesn't have to worry about it staying up on her profile forever, which encourages casual and impromptu sharing.

- **Spontaneous Moments:** While hanging out with his pals, John decided to use one of Snapchat's AR lenses to create a humorous movie. After filming the video, he uploads it straight to his group chat. As soon as his pals watch the video, they react by sending in their own amusing movies. The exchange feels less formal and more enjoyable thanks to the temporary messages.
- **Event Documentation:** Emma wants to keep her pals informed about the plans for her surprise birthday celebration. She shares brief shots of the cake and decorations on Snapchat. Her friends experience a sense of participation in the planning as they follow her story. Emma can go back to her Story after the celebration and relive the moments because she knows it was provided in a light, transient way.

5 Stakeholders

5a The Client

Content

Important decision-makers like the CEO or the product development team would be Snapchat's clients. These people are in charge of the app's general strategy and financial commitment.

Motivation

The final decision to approve the finished product rests with the client. Since they are the ones making the development investments in Snapchat, their contentment is vital. In this case, the customer is making sure the software achieves its objectives and keeps its user base interested.

Considerations

The marketing department's representatives might be integrated into the client list if Snapchat were being created for users outside of the company. In such cases, a marketing representative would be designated as the customer to guarantee that the app complies with brand strategy and market demands.

5b The Customer

Content

Customers who download and utilize the Snapchat app are referred to as such by Snapchat. These are usually younger people, maybe in their teens or early 20s, who are searching for an enjoyable and simple way to spend time with friends.

Motivation

Whether Snapchat is worth a user's time and attention is up to them. It's crucial to recognize their needs in order to make sure the app fulfills them, whether it be privacy, instantaneous communication, or interesting features like Stories and filters. To create a product that people would like and keep using, it is essential to comprehend these goals.

5c Hands-On Users of the Product

Content

This section outlines the key stakeholders who will be using Snapchat. For each type of user, we describe:

User name/category: The general group or type of people who will use the app, like teenagers or young adults.

- **User role:** These people mostly use Snapchat to quickly and casually send messages, movies, and photographs with friends.
- **Subject matter experience:** While most users are unfamiliar with social media's business aspects, they are accustomed to interacting with people through related platforms.
- **Technological experience:** Users of Snapchat typically possess a journeyman's degree of technological expertise. They are at ease using social networking apps, smartphones, and simple photo and video editing functions.

Other user characteristics: This section describes specific traits of Snapchat users that influence the app's design and features:

- **Physical abilities/disabilities:** Take into account the app's accessibility features to make sure users with physical limitations can utilize it with ease.
- **Intellectual abilities/disabilities:** Consider the accessibility features of the software to ensure that people with physical constraints may easily use it.
- **Attitude toward technology:** Since most Snapchat users are tech-savvy and willing to try new things, the app should be simple to use yet still creative.
- **Education:** Since students and young professionals make up the majority of users, the app should be suitable for a broad spectrum of educational backgrounds
- **Linguistic skills:** In order to satisfy its worldwide user base, Snapchat ought to offer multilingual support.
- **Age group:** Since the majority of the app's users are between the ages of 13 and 30, it needs to be simple and aesthetically pleasing for them.
- **Gender:** The app should provide a varied range of features and content that appeals to users of all genders, with an inclusive design.

Motivation

Users are searching for an entertaining, interesting method to communicate with peers and share routine occasions. They cherish both the transient character of the content and privacy.

Examples

There is a broad spectrum of backgrounds among Snapchat users. They could be non-techies, young professionals, students, or even casual users of social media. Teenagers hanging out with friends, working parents preserving family memories, and even individuals from various nations and cultures might all be among the app's users. In essence, anyone interested in interacting with people and owning a smartphone could be a prospective Snapchat user.

5d Priorities Assigned to Users

Content

When designing Snapchat, it's important to prioritize different types of users based on their impact on the app's success:

- **Key users:** These are youngsters and teens that make up Snapchat's core user base and are essential to the app's success. Since user interaction fuels the app's growth and popularity, their demands and preferences are given high consideration.
- **Secondary users:** These users still use Snapchat, but their influence on its long-term success is less significant. For example, older adults who might use the app occasionally. While their feedback is valuable, it doesn't outweigh the needs of key users.
- **Unimportant users:** This category consists of infrequent users, those who might abuse the app, and those who don't help the app succeed. When it comes to design choices, their requirements are prioritized the lowest.

By establishing these goals, Snapchat's development team can concentrate on satisfying the demands of its most significant users, assuring the app's continued success.

Motivation

Knowing which users are most critical to Snapchat's success is critical when building the app. For instance, there may be a sharp decline in engagement and popularity if the app's primary user base, which consists primarily of teens and young adults, is unable to locate what they're looking for. As such, design choices ought to be strongly influenced by their needs.

However, some users might use Snapchat infrequently and have minimal effect on the platform's overall performance. These customers won't voice complaints if their demands aren't satisfied and won't make significant contributions to the app's expansion. Consequently, during the design phase, any particular demands from these less important consumers ought to be given less weight.

5e User Participation

Content

It's critical to explain how each type of Snapchat user's input will influence the development of the app. This could include their opinions on the design of the UI, their insights into how they use social media, or recommendations for improving the usability of the app. For instance, it may be necessary for important users, such as teenagers and young adults, to take part in usability testing or new feature testing. To make sure the app fulfills their requirements and expectations, it is crucial that customers take the time to share their experiences and preferences.

Motivation

Users' lack of participation in projects is a common reason of failure, sometimes due to a lack of communication about the significance of their participation. Users frequently give priority to their daily job when given the option to choose between contributing to a new project and their regular chores. It's critical to establish up front just how much user involvement Snapchat needs to grow in order to prevent this. This makes it possible to guarantee that users will set aside the time and energy required to offer insightful and helpful comments.

5f Maintenance Users and Service Technicians

Content

Snapchat's maintenance users are in charge of updating and maintaining the platform. They require particular tools to manage and alter the program, like those for updating features or resolving problems.

Motivation

Determining these maintenance users' traits enables us to make sure their demands are met. We can identify and incorporate crucial criteria that might otherwise go unnoticed if we are aware of who will be maintaining Snapchat and what their responsibilities entail. This facilitates timely upgrades and bug fixes while also keeping the app operating properly.

5g Other Stakeholders

Content

Other people or groups that have an influence on or support Snapchat's growth are listed in this section. Among these parties are:

Examples of stakeholders:

- **Sponsor:** The key investor or financial organization driving Snapchat's development.
- **Testers:** Users who will test Snapchat in order to identify and report any problems.
- **Business analysts:** Experts who research consumer demands and market trends to inform Snapchat's features and approach.
- **Technology experts:** specialists who provide information on the newest technological advancements and Snapchat-specific technical solutions.
- **System designers:** designers who build the application's framework and underlying structure.
- **Marketing experts:** those in charge of advertising Snapchat and creating plans to draw in and keep users.
- **Legal experts:** advisors making sure Snapchat abides by all applicable rules and laws.
- **Domain experts:** people who are extremely knowledgeable about social media and user behavior and who offer insightful commentary.
- **Usability experts:** Experts concentrated on making Snapchat simple and entertaining for consumers.
- **Representatives of external associations:** individuals from groups or regulatory agencies that are part of the industries that Snapchat may impact or influence.

For each type of stakeholder, provide the following information:

- **Identification:** Their name, role, and affiliation.
- **Knowledge Required:** The level of experience or knowledge they offer.
- **Degree of Involvement:** The extent to which they must participate in the project.
- **Degree of Influence:** The extent to which their contributions affect the project.

Motivation

Snapchat's success may be impacted if these stakeholders are not identified and engaged, which could result in requirements being ignored.

6 Mandated Constraints

For Snapchat, there are some important rules we need to follow when designing the app. These rules are set at the start of the project and are crucial for making sure everything works right. Each rule comes with a description of what's needed, why it's important, and how we'll check if we've done it correctly. Just like any other requirements, these rules help us build Snapchat in a way that's secure, easy to use, and compliant with laws. They ensure the app runs smoothly on iOS and Android, keeps user data safe, and meets all the necessary standards from day one.

6a Solution Constraints

Content

Snapchat has to be secure, fast, and easy to use. This means every snap should disappear as expected, messages are private, and the app feels smooth and fun to use, with a design that's fresh but simple.

Motivation

Snapchat users love the quick, private interactions the app offers. Prioritizing security and responsiveness helps build trust, so users feel confident sharing moments without worrying about privacy or laggy performance.

Considerations

Security involves keeping all messages and snaps safe with strong encryption, and real-

time interactions mean the app's backend must be able to keep up with high traffic. The design needs to be intuitive and visual, perfect for Snapchat's style.

Example

For example, when you send a snap, Snapchat ensures it's encrypted for safety and set to disappear, so users can enjoy a fleeting moment of sharing, knowing it won't linger on the platform.

6b Implementation Environment of the Current System

Content

The implementation environment of Snapchat includes the technological and physical setup where the app operates and interacts with various devices and systems. This environment encompasses:

- **Mobile Operating Systems:** Snapchat is primarily used on mobile devices running iOS and Android. These platforms provide the foundational software environment where Snapchat is installed and executed.
- **Smartphone Hardware:** Snapchat needs to function on a wide range of smartphones with varying hardware capabilities, including different camera qualities, processors, memory, and screen resolutions.
- **Network Connectivity:** The app relies heavily on internet connectivity, both Wi-Fi and mobile data (3G, 4G, 5G), to send and receive snaps, messages, and other data in real-time.
- **Cloud Infrastructure:** Snapchat utilizes cloud servers for storing user data, such as photos, videos, and chat histories, as well as for running backend processes like AI filters and content moderation.
- **Third-Party Integrations:** Snapchat integrates with several non-human systems, such as external APIs for augmented reality filters, maps, and other third-party services that enhance the user experience.

Motivation

Understanding the implementation environment is crucial for designing Snapchat to fit seamlessly with the surrounding technology. This environment dictates specific design constraints that the app must adhere to, such as compatibility with different mobile operating systems, performance on various hardware configurations, and smooth operation under varying network conditions. This knowledge ensures the app can effectively interact with all elements of its technological ecosystem, providing a consistent and reliable user experience.

Considerations

When describing the implementation environment for Snapchat, it is important to include all relevant components:

- **Mobile Device Capabilities:** Different devices have different capabilities, and Snapchat must perform well across all supported devices.
- **Network Variability:** The app should be optimized for various network conditions, including low bandwidth scenarios.
- **Security Measures:** The implementation environment also includes security protocols and measures, such as encryption and data protection standards, that are vital for user safety and compliance.
- **Organizational Context:** If Snapchat is integrated into a larger organizational setup or partnered with other companies, an understanding of the organizational context and roles involved is important for smooth operation.

This comprehensive overview helps developers and designers create a version of Snapchat that works efficiently in its intended environment and meets user expectations.

6c Partner or Collaborative Applications

Content

- **Third-Party APIs:** Snapchat works with other companies' APIs to bring cool features to the app, like AR filters, maps, and weather updates. This makes the app more fun and interactive for everyone.
- **Advertising Platforms:** Snapchat partners with ad networks to show ads to users. This helps Snapchat earn money while making sure that user data is handled with care.
- **Payment Gateways:** For things like buying in-app products or using the Snap Store, Snapchat collaborates with payment processors to keep transactions safe and secure.
- **Social Media Integrations:** Snapchat connects with other social media platforms, making it easy for users to share their snaps across different networks and find friends by importing contacts.
- **In-House Tools:** Snapchat uses its own tools for things like analyzing user data, moderating content, and providing customer support to keep everything running smoothly.

Motivation

- **Understanding Integrations:** By knowing how Snapchat works with these different apps and tools, we can spot any challenges early and make sure everything works seamlessly together.
- **Ensuring Security and Privacy:** It's important to keep user data safe and private. By managing these partnerships carefully, Snapchat can ensure it meets all privacy rules and keeps user trust.
- **Maintaining Performance:** We want to make sure these partnerships don't slow down the app. Keeping Snapchat fast and responsive is key to a good user experience.
- **Handling Updates:** Staying prepared for updates from partner apps ensures Snapchat keeps running smoothly, even when changes happen.

Considerations

- **Integration Challenges:** We need to think about how well Snapchat can integrate with these partners and what issues might pop up.
- **Security and Privacy:** It's crucial to make sure any data shared with partners is secure and respects privacy laws.
- **Performance Impact:** We have to consider how these partnerships might affect the app's speed and reliability.
- **Maintenance and Updates:** Planning for regular maintenance and being ready for updates helps Snapchat stay on top of any changes and keep things running smoothly.

6d Off-the-Shelf Software

Content

Snapchat doesn't reinvent the wheel for everything—it uses pre-made tools like cloud storage for handling the huge volume of snaps and third-party filters to add fun effects, which makes sense instead of building all features from scratch.

Motivation

Using off-the-shelf software helps Snapchat save time and focus on what makes the app unique, like creating exclusive filters or new interactive features.

Considerations

Snapchat has to pick reliable software that integrates seamlessly. These tools should be secure, scalable, and fit with Snapchat's fast-paced, visual nature.

Example

Snapchat might use cloud storage to easily store and retrieve snaps, or partner with a third-party provider for special effects, so users get fresh options without waiting on months of development.

6e Anticipated Workplace Environment

Content

Snapchat's work culture is creative, fast, and very team-oriented. Departments like design, engineering, and product management work closely together, sparking ideas and making quick improvements to keep the app exciting.

Motivation

In the fast-moving social media world, it's crucial to test new ideas quickly. A collaborative, flexible work environment helps Snapchat keep up with trends and consistently bring new, fun features to users.

Considerations

It's important to have open communication across departments and foster a space where everyone can share and test ideas without being bogged down by red tape.

Example

For instance, a new filter might start as an idea from the design team, then developers and product managers brainstorm together to make it a reality in just a few weeks, ready to surprise users.

6f Schedule Constraints

Content

Snapchat has to work within tight deadlines to push out updates and features quickly. This means Snapchat needs to be agile, releasing updates in stages and gathering feedback as they go.

Motivation

Frequent updates are what keep Snapchat fresh. To keep people engaged and attract new users, Snapchat needs to adapt to feedback and launch new features consistently.

Considerations

Snapchat's team uses a flexible approach, often releasing features in phases to get feedback and make tweaks along the way. Prioritizing essential features ensures that updates are meaningful and timely.

Example

If Snapchat wanted to introduce a new voice messaging feature, they might first release a simple version, gather user reactions, and then fine-tune it based on real feedback. This keeps the feature relevant and engaging without missing a beat.

6g Budget Constraints

Content

Budget Allocation: The cost of developing a social media app like Snapchat varies based on its complexity. For example:

- **Simple App:** \$25,000 – \$35,000
- **Medium App:** \$35,000 – \$45,000
- **Complex App:** Starts from \$50,000

Motivation

- **Budget Limits and Requirements:** The development budget directly influences the features and scope of the app. If the budget is limited, it may constrain the number of features that can be included. For instance:
 - A budget of \$25,000 – \$35,000 may only cover basic features like photo sharing and image filters.
 - A higher budget of \$35,000 – \$45,000 allows for more advanced features such as augmented reality filters and live video streaming.
 - For a budget starting from \$50,000, you can incorporate highly advanced features like virtual gifting and integrating AI technologies.
- **Determining True Interest:** If the client's budget is significantly below the cost estimates for their desired feature set, it may indicate a lack of commitment or value placed on the project. It's crucial to assess whether the client is willing to adjust their expectations or increase the budget to meet their requirements.

Considerations

- **Realistic Budget Assessment:** It is essential to evaluate whether the budget aligns with the desired features and quality of the app. Key considerations include:

- **Scope of Work:** A clear understanding of what can be realistically achieved within the budget. If the budget is insufficient for the desired features, prioritizing essential features or scaling down the project may be necessary.
- **Client Commitment:** Assess whether the client is truly invested in the project. If the budget constraints suggest that the client is not fully committed or does not value the project highly, it may be worth discussing the feasibility of continuing with the current budget.

7 Naming Conventions and Definitions

7a Definitions of Key Terms

It's critical to clarify all terminology, acronyms, and abbreviations utilized in Snapchat's project. To ensure that everyone involved understands them and to prevent any unintentional connotations, these should be carefully picked. The most crucial terminology thought to be mentioned here, and the paper should contain a comprehensive glossary for further reference.

- **Glossary:** All important names, acronyms, abbreviations, and terminology used in the project are defined in this section.
- **Clarity:** To guarantee that everyone is in agreement on the definition of each term, it should be properly defined with input from pertinent stakeholders.
- **Prevent Ambiguity:** To avoid miscommunication, omit acronyms whenever feasible. To make the document easier to read, use complete terms instead.
- **Acronyms:** Provided that the glossary provides a clear definition, they are permitted.

By using this method, the project's terminology is guaranteed to be clear and consistent, which lowers the possibility of misunderstandings or confusion.

Motivation

Names have power. They communicate precise meanings that can avoid misunderstandings and save time when they are well defined. Potential confusion can be prevented early in the project by properly selecting and specifying names. Throughout the project, the glossary that was produced during the requirements phase will be an invaluable tool that keeps everyone in sync.

Examples

- **Snap:** A Snapchat-sent image or video that vanishes once it is seen.
- **Story:** A set of Snaps that friends can view for a full day.

In order to enhance clarity in communication and prevent misunderstandings, these phrases have been defined expressly within the Snapchat environment.

Considerations

When defining terms, it's critical to refer to current sources and data dictionaries. Renaming well-known items is not a good idea unless they are really ambiguous and lead to misunderstanding. Emphasize from the outset how crucial it is to stay away from homonyms and synonyms because they can add complexity and expense to the project. Maintaining consistency in terminology is essential to keeping everyone in the loop.

7b UML and Other Notation Used in This Document

Content

Any symbols, punctuation, superscripts, subscripts, and other notations used in the text are explained in this section. The established standards—if any—along with any

deviations from them will be stated here. This guarantees that the precise notation used in the documentation is understood by all.

Motivation

When particular symbols—such as solid or hollow arrows—have significant connotations, it's critical that everyone recognizes these differences. Throughout the project, accurate communication is ensured and misunderstandings are avoided with proper notation.

Considerations

It may be more appropriate to describe a notation directly at that location rather than in a generic notation section if it is only used in that one particular location, as on a single diagram or in that particular section. This method maintains the explanation accessible and pertinent.

Example

The OMG UML Version 2.0 standard, as described by Fowler, is mostly followed in this paper. Any deviations from this norm are noted in the relevant places.

7c Data Dictionary for Any Included Models

Content

Definitions for all data flows and storage utilized in models are given in this part, with an emphasis on describing the characteristics of the data displayed in context models. Any technical details regarding the interfaces shown in these models are likewise included

.

Motivation

The context diagram provides an accurate definition of the scope of the work being studied or the scope of the product to be built. This definition can be completely accurate only if the information flows bordering the scope have their attributes defined.

Examples

Road de-icing schedule = issue number + {road section identifier + treatment start time + critical start time + truck identifier} + depot identifier

As you progress through the requirements specification, define each of the elementary terms in detail.

Considerations

The dictionary provides a link between the requirements analysts and the implementers. The implementers add implementation details to the terms in the dictionary, defining how the data will be implemented. Also, implementers add terms that are present because of the chosen technology and that are independent of the business requirements.

8 Relevant Facts and Assumptions

8a Facts

Content

- **User Behavior:** Snapchat users frequently interact with a large number of Snaps and Stories on a daily basis, enhancing their material with a variety of features such as filters and lenses.

- **Platform Features:** Snapchat includes augmented reality effects, movies, and picture integration. Moreover, it has tools like Stories, Discover, and Spotlight that influence how users communicate and share material.
- **Business Practices:** Snapchat emphasizes particular features or collaborations and monetizes the app through sponsored content and advertising.

Motivation

- **User Engagement:** Designing features that improve engagement and usability is made easier by the knowledge that users frequently submit multiple Snaps and see multiple Stories every day.
- **Feature Development:** Knowing that Snapchat provides a large selection of lenses and filters highlights the necessity for frequent updates and innovative features to maintain the app's attractiveness.
- **Platform Integration:** Understanding the ways in which Snapchat's multimedia features are employed contributes to performance and compatibility across various hardware and operating systems.

Examples

- **User Statistics:** Every day, 4 billion Snaps are created and viewed by Snapchat users on average.
- **Features:** Snapchat users can add more than 200 different filters and lenses to their images and videos.
- **App Size:** Depending on the device and upgrades, the Snapchat app's normal size is approximately 100 MB.

8b Assumptions

Content

- **Environment of Operations:**
 - **Technological Compatibility:** It is anticipated that Snapchat will function well on a range of iOS and Android smartphone models.
 - **Network Conditions:** Snapchat ought to function seamlessly and perform effectively across a variety of networks, including 4G, 5G, and WiFi.
- **Expectations for Features:**
 - **User Preferences:** New features, such as augmented reality lenses and filters, should be sought out and used often by users.
 - **Interactivity:** Expect seamless updates and new features as users interact with Snaps, Stories, and Discover.
- **Achievement:**
 - **Speed and responsiveness:** Even during periods of high user activity, Snapchat should load swiftly and function without a hitch.
 - **Optimization:** The application should function reliably on a range of devices and in a variety of network scenarios.
 - **User Base:** Engagement Patterns: It is expected that users will continue to use Stories and Snaps on a regular basis.
 - **Demographics:** It is anticipated that a younger user base will dominate, which will have an impact on content strategy and feature design.

Motivation

- **Clarify Expectations:** Make sure that everyone involved is aware of the goals and constraints that Snapchat is expected to meet. This matches everyone's

understanding of the app's capabilities and limitations and helps control what can be delivered in a practical manner.

- **Educate:** Ensure that all those working on Snapchat's development are aware of the presumptions that inform choices. This knowledge guarantees that assumptions are taken into account in the design and functionality of the app and aids in the early resolution of any possible problems.

Examples

- **Technological Environment:** Presumes that Snapchat will function flawlessly with a range of devices and screen sizes on the most recent iterations of iOS and Android.
- **Feature Availability:** This section makes the assumption that at the time of the next significant update, additional features like interactive elements or augmented reality (AR) lenses will be available and included into the app.
- **Dependencies:** Makes the assumption that in order to support Snapchat's functionality, third-party services, such as cloud storage or analytics APIs, will continue to operate consistently and dependably.
- **User Expectations:** This section makes the assumption that users will want regular updates and new features. As a result, development and release schedules for the app will be guided by these expectations.

Considerations

- **Software Tools:** Determine which development tools, SDKs, and APIs will be available for integrating new features into Snapchat. This includes evaluating if the latest tools will support desired functionalities and enhancements.

- **New Products:** Assess if upcoming software updates or new products (e.g., changes in mobile OS versions) could impact Snapchat's performance or compatibility. This helps in planning for any necessary adjustments or updates.
- **Current Products:** Consider if existing tools or services (e.g., APIs for analytics or cloud services) will be used in new ways, or if there are changes that could affect how Snapchat integrates with other applications and services.
- **Business Changes:** Stay informed about potential changes in the business environment, such as new advertising regulations or privacy laws, which could influence Snapchat's features, data handling practices, or overall operations.

II Requirements

9 Product Use Cases

Use case diagrams for Snapchat will help define the boundaries of the system and identify the specific functionalities and interactions between users and the system. The use cases will illustrate what the system does and how it interacts with its actors (users, admins, and external services).

9a Use Case Diagrams

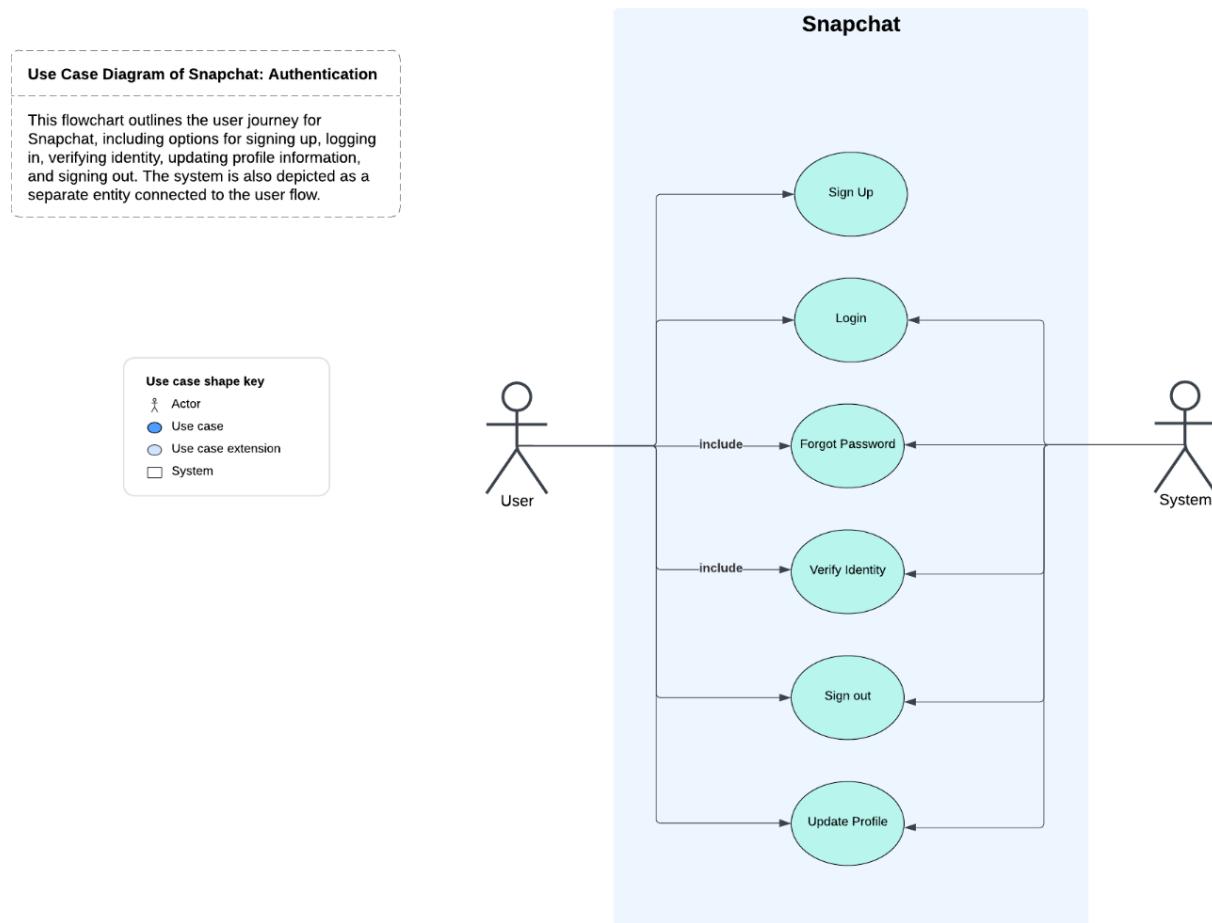


Figure 2: Use Case Diagram of Module 1: Authentication

(Website: <https://lucidchart.com>)

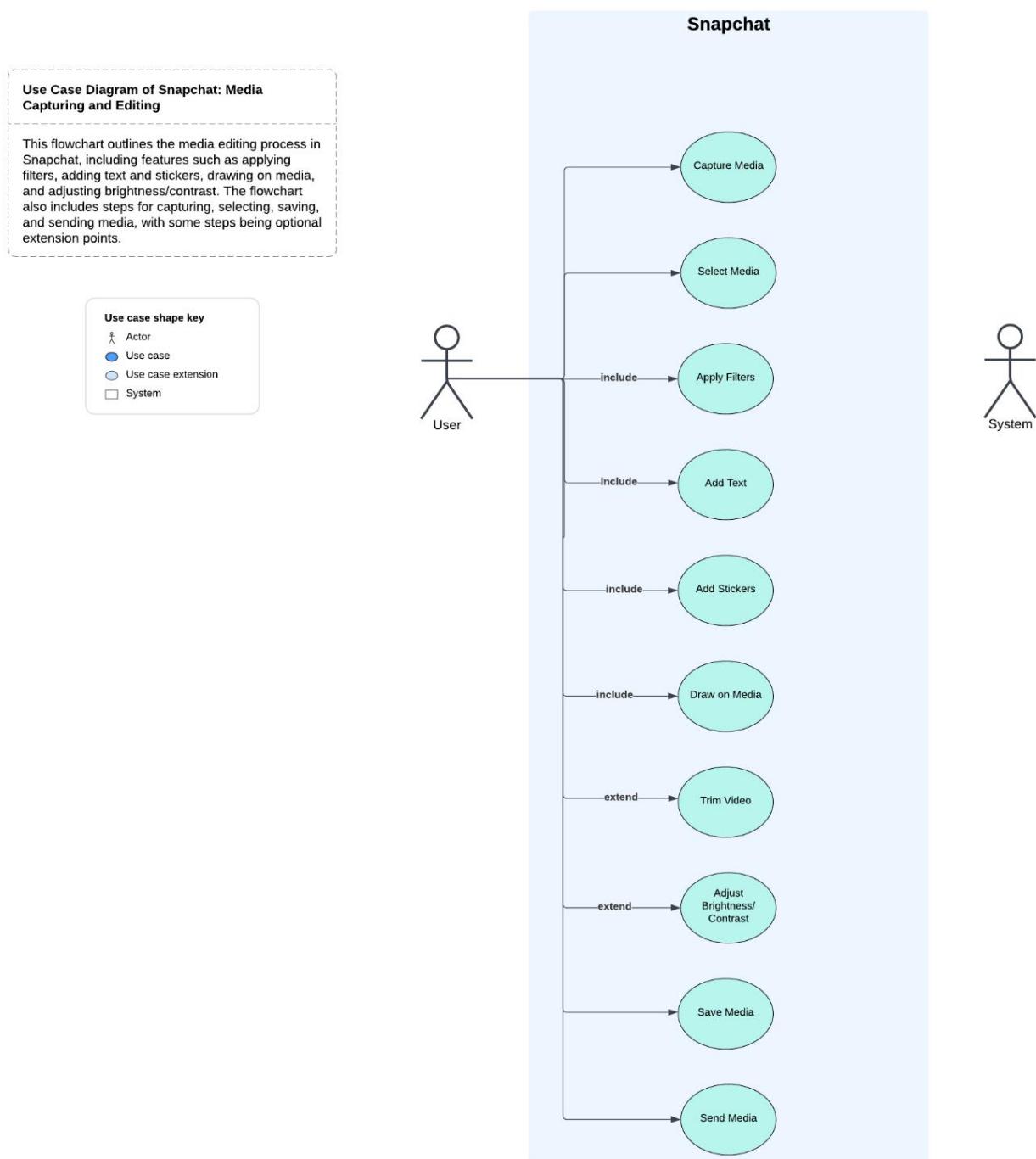


Figure 3: Use Case Diagram of Module 2: Media Capturing and Editing

(Website: <https://lucidchart.com>)

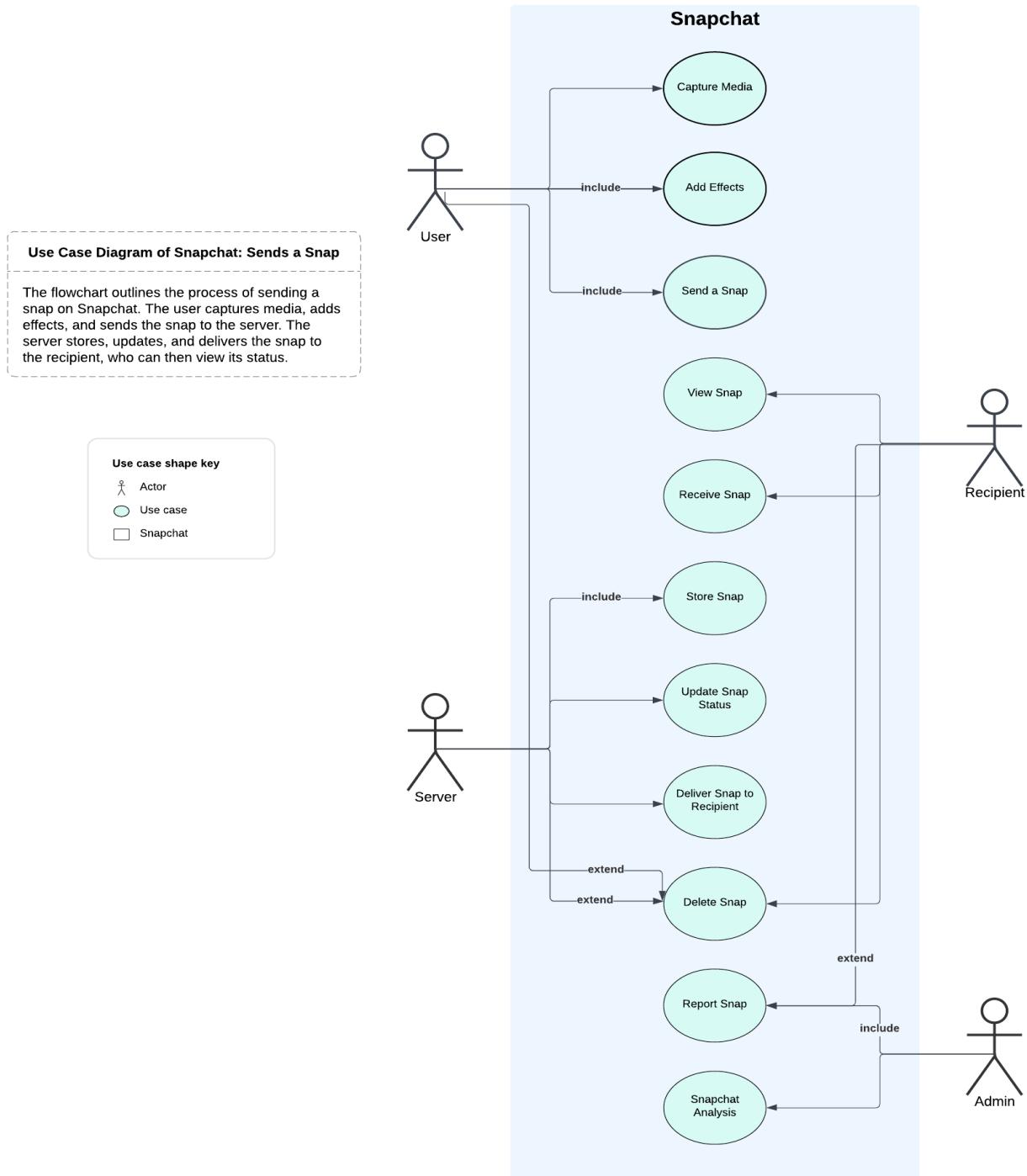


Figure 4: Use Case Diagram of Module 3: Sends a Snap

(Website: <https://lucidchart.com>)

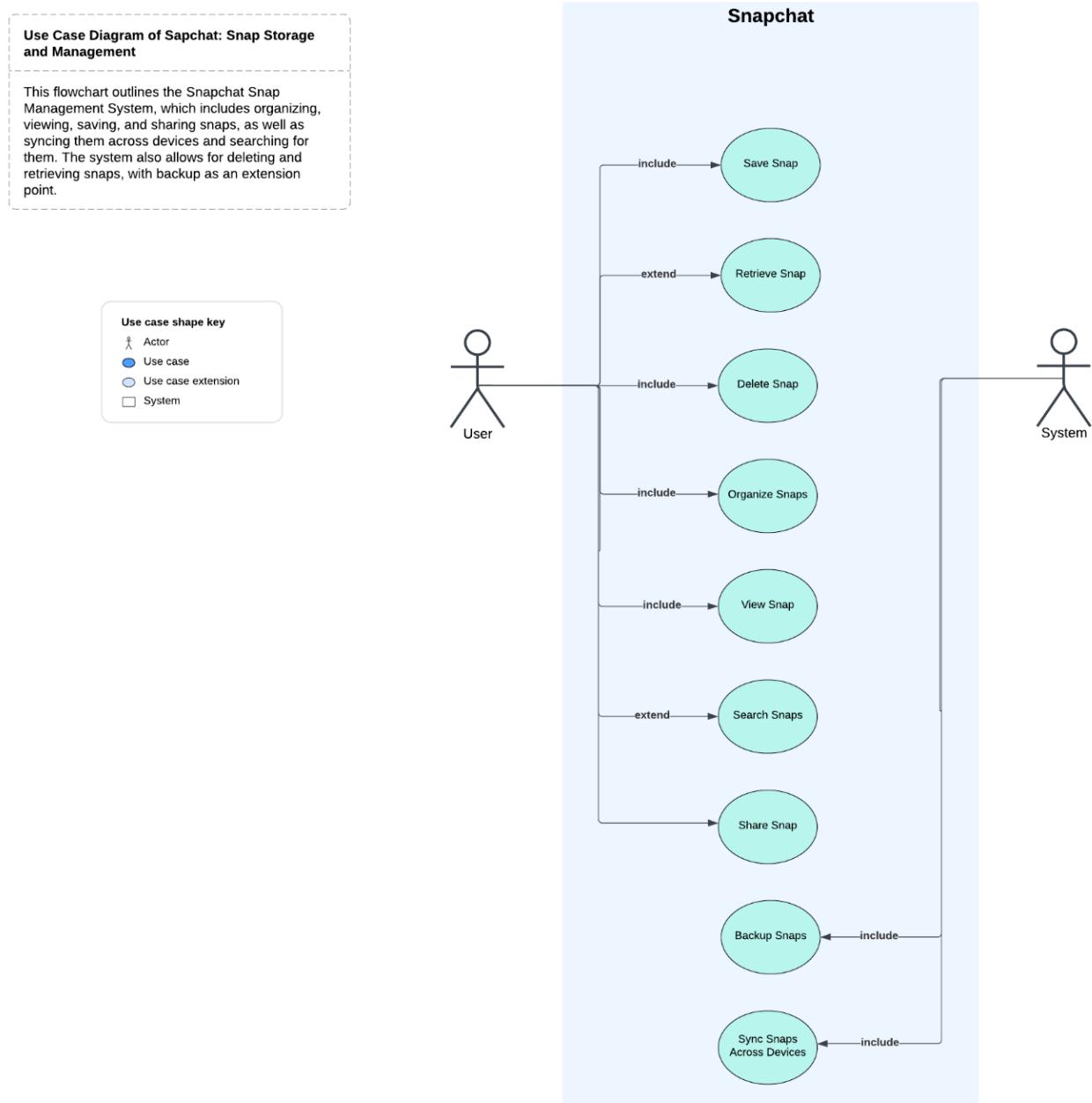


Figure 5: Use Case Diagram of Module 4: Snap Storage and Management

(Website: <https://lucidchart.com>)

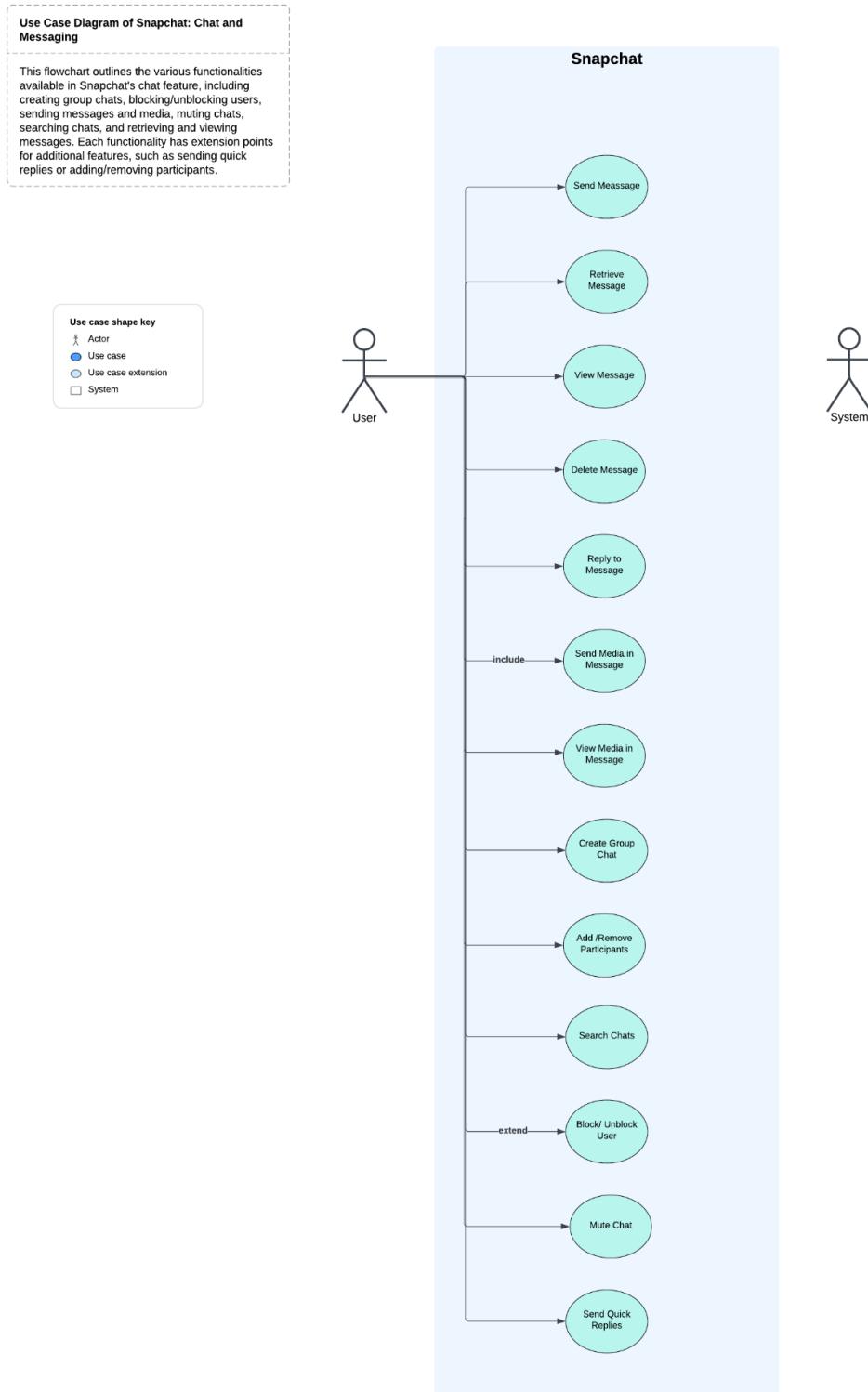


Figure 6: Use Case Diagram of Module 5: Chat and Messaging

(Website: <https://lucidchart.com>)

9b Product Use Case List

- 1. Sign Up/Log In:** Users create a new account or log into an existing one.
- 2. Send Photo/Video:** Users capture and send photos or videos to their friends.
- 3. View Stories:** Users view stories posted by friends and other users.
- 4. Apply Filters:** Users apply filters and effects to their photos and videos.
- 5. Send Messages:** Users send text or multimedia messages to contacts.
- 6. Make Video/Audio Calls:** Users initiate and receive video or audio calls.
- 7. View Chat History:** Users review past conversations and messages.
- 8. Manage Contacts:** Users add or remove friends and manage their contact list.
- 9. Update Profile:** Users update their personal information and profile settings.
- 10. Report Content:** Users report inappropriate content or behavior.
- 11. Admin Manage Users:** Admins manage user accounts and handle content reports.

9c Individual Product Use Cases

These use cases define specific interactions between users and the Snapchat system, outlining the system's responses to user actions and specifying the conditions and exceptions for each action. It shows how a particular use case works for an individual.

1. Use Case: Sign Up/Log In

Purpose: Allows users to create a new account or log into an existing account.

Primary Actor: User

Preconditions:

- The user has access to the Snapchat app.
- The user has an active internet connection.

Main Flow of Events:

1. User opens the Snapchat app.

2. User selects "Sign Up" or "Log In."

3. If "Sign Up":

- User enters personal details (name, email, password).
- User agrees to the terms of service.
- User taps "Create Account."
- System validates the information and creates a new account.
- System sends a confirmation email to the user.

4. If "Log In":

- User enters email and password.
- User taps "Log In."
- System validates the credentials and grants access.

Postconditions:

- User is successfully logged into the app or account is created and verified.

Exceptions:

- If credentials are incorrect, the system displays an error message.
- If the email is already in use, the system prompts the user to log in instead.

2. Use Case: Send Photo/Video

Purpose: Allows users to capture and send photos or videos to friends.

Primary Actor: User

Preconditions:

- The user is logged into their Snapchat account.
- The user has access to the camera and gallery.

Main Flow of Events:

1. User opens the Snapchat app.
 2. User selects the camera icon to capture a photo or video.
 3. User takes a photo or records a video.
 4. User applies filters or edits the media (optional).
 5. User selects the contacts or friends to send the media to.
 6. User taps "Send."
 7. System uploads the media and sends it to the selected recipients.
-

Postconditions:

- The media is sent to the selected contacts and appears in their chat feed.

Exceptions:

- If the upload fails, the system displays an error message.
- If the user cancels the action, the media is not sent.

3. Use Case: View Stories

Purpose: Allows users to view stories posted by their friends and other users.

Primary Actor: User

Preconditions:

- The user is logged into their Snapchat account.
- The user has contacts who have posted stories.

Main Flow of Events:

1. User opens the Snapchat app.
2. User navigates to the "Stories" section.
3. User selects a story to view.
4. System displays the selected story in a full-screen view.
5. User can swipe through multiple stories or tap to move to the next story.

Postconditions:

- User successfully views the selected stories.

Exceptions:

- If the story is no longer available, the system displays a message indicating that the story has expired or been removed.

4. Use Case: Apply Filters

Purpose: Allows users to apply various filters and effects to their photos or videos.

Primary Actor: User

Preconditions:

- The user is logged into their Snapchat account.
- The user has captured a photo or video.

Main Flow of Events:

1. User captures a photo or video.
2. User selects the "Filters" option.
3. System displays available filters and effects.
4. User selects a filter or effect.
5. User applies the filter to the media.
6. User can preview the media with the filter applied.
7. User saves or shares the media.

Postconditions:

- The media is updated with the selected filter and can be saved or shared.

Exceptions:

- If the filter application fails, the system displays an error message.
- If the user cancels the action, the media remains unchanged.

5. Use Case: Report Content

Purpose: Allows users to report inappropriate content or behavior.

Primary Actor: User

Preconditions:

- The user is logged into their Snapchat account.
- The user has encountered content they want to report.

Main Flow of Events:

1. User views the content they wish to report.
2. User taps on the content options menu.
3. User selects "Report."
4. System prompts the user to select a reason for reporting (e.g., harassment, inappropriate content).
5. User submits the report.
6. System logs the report and notifies the admin for review.

Postconditions:

- The report is logged and sent to the admin for review.

Exceptions:

- If the report submission fails, the system displays an error message.

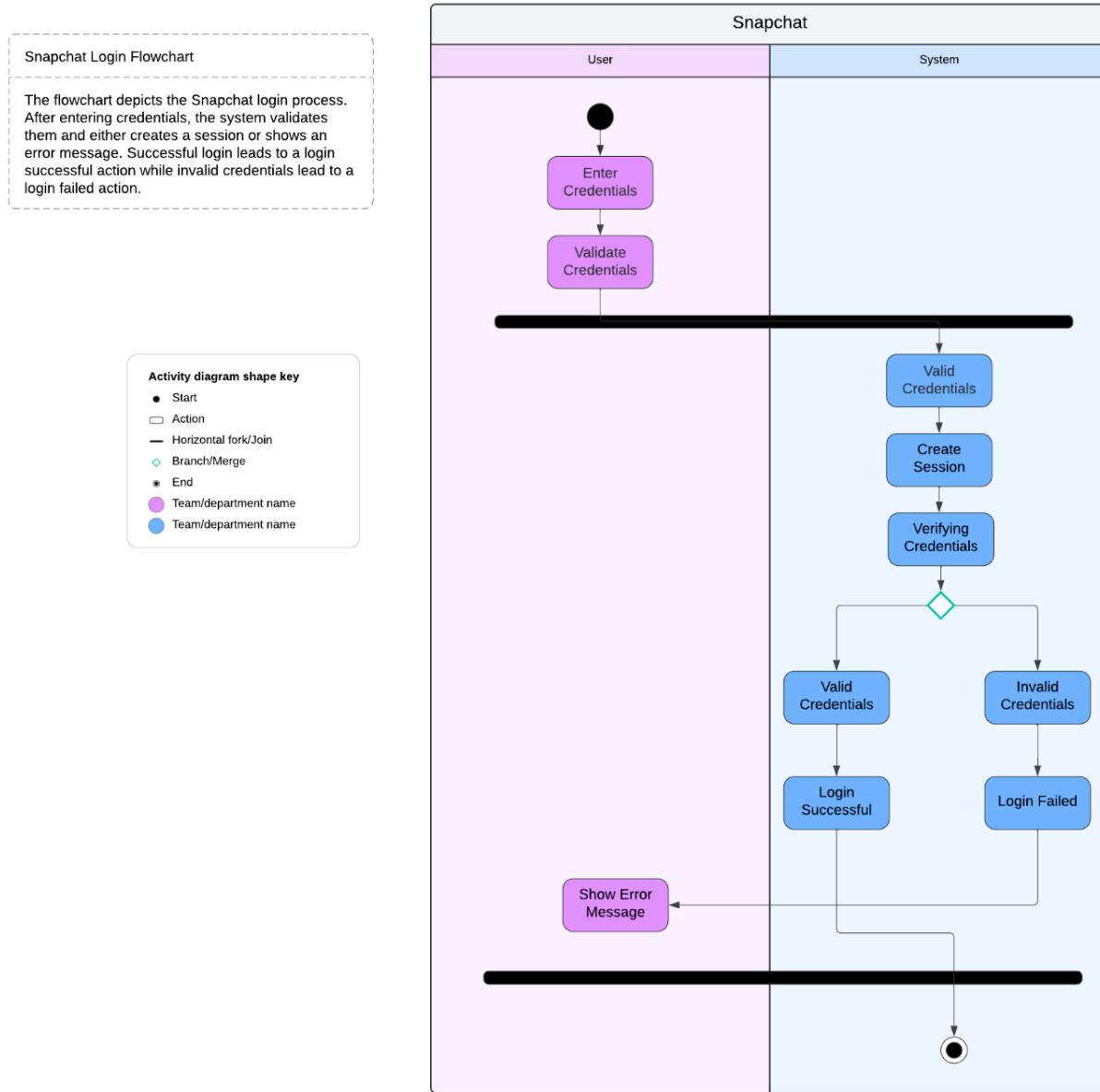


Figure 7: Activity Diagram of Module 1: Authentication

(Website: <https://lucidchart.com>)

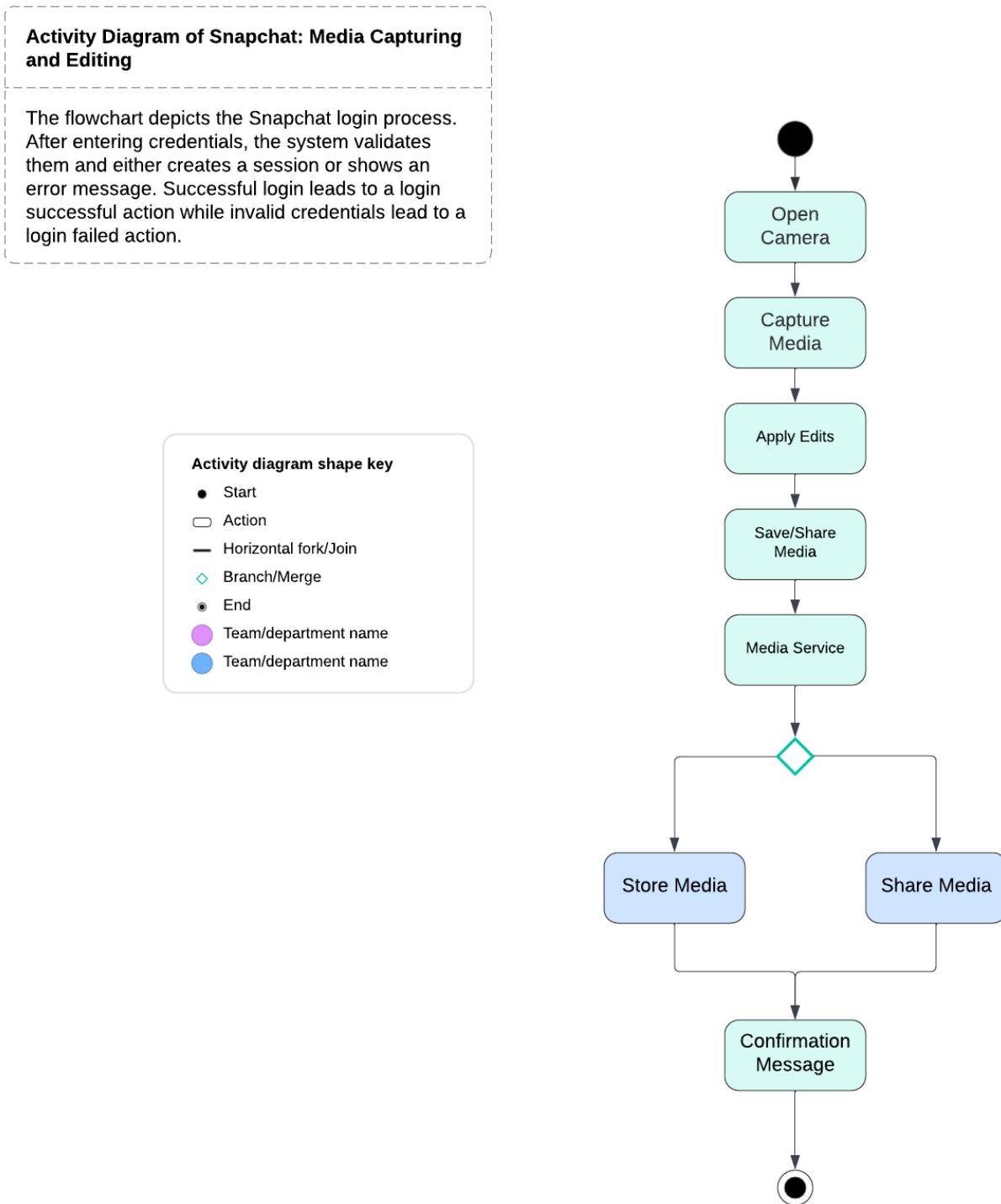


Figure 8: Activity Diagram of Module 2: Media Capturing and Editing

(Website: <https://lucidchart.com>)

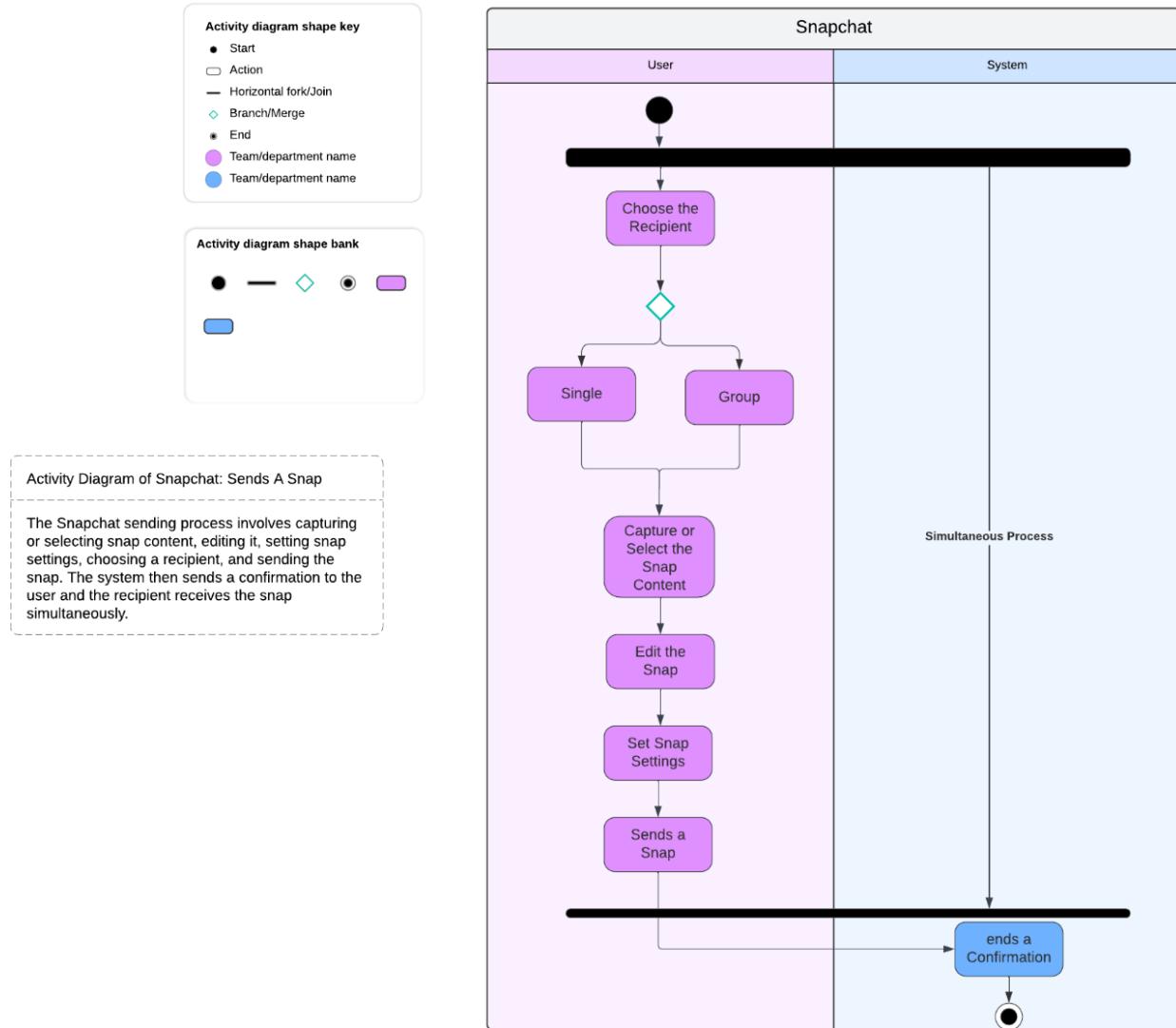


Figure 9: Activity Diagram of Module 3: Sends a Snap

(Website: <https://lucidchart.com>)

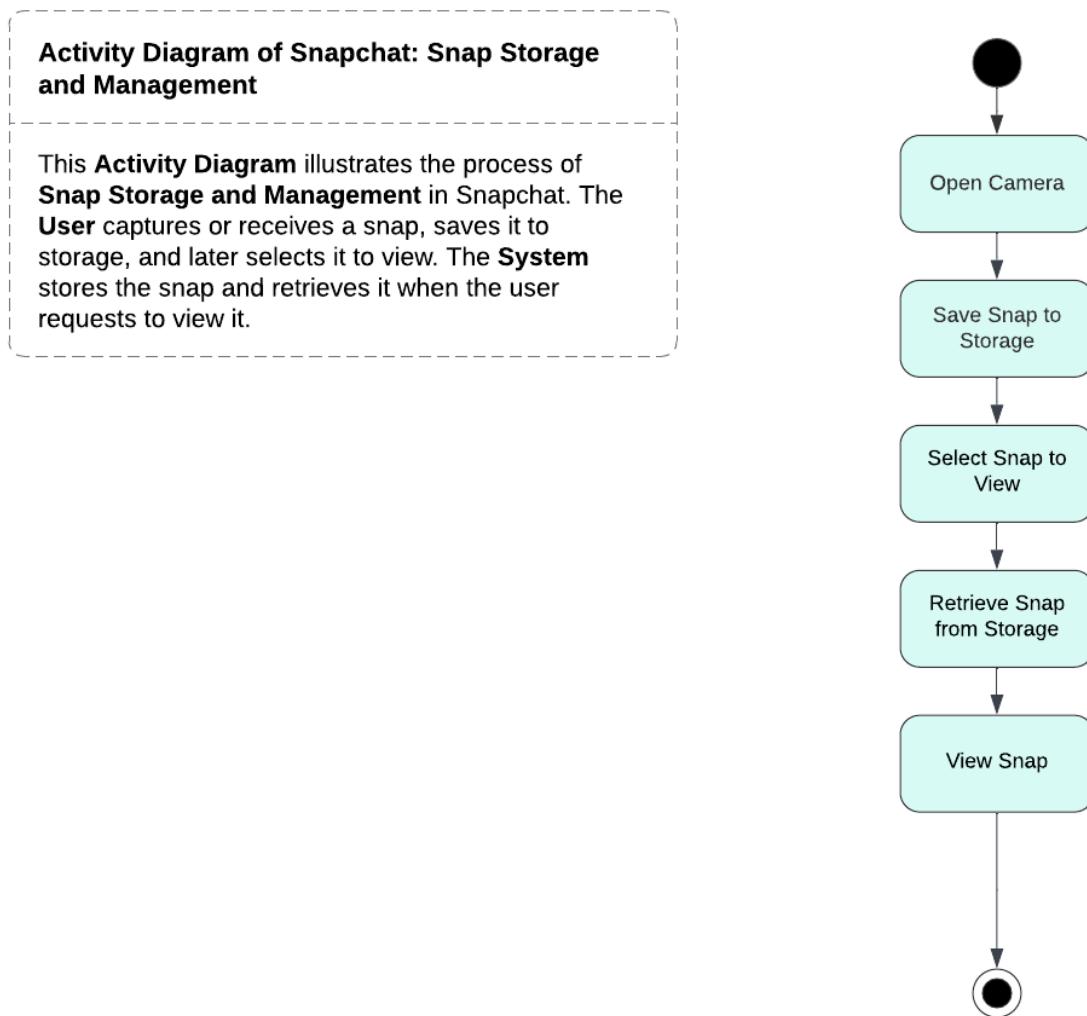


Figure 10: Activity Diagram of Module 4: Snap Storage and Management

(Website: <https://lucidchart.com>)

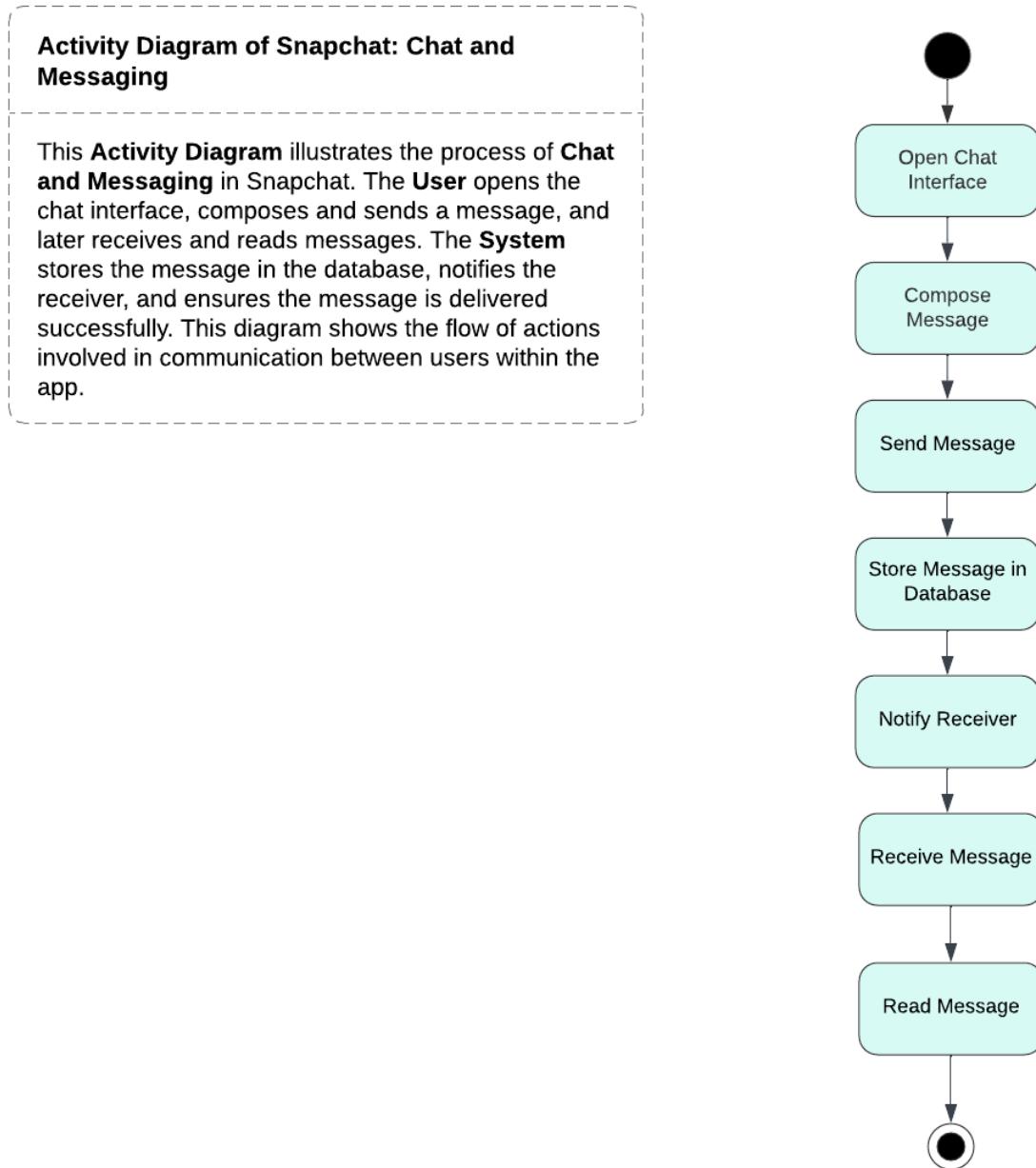


Figure 11: Activity Diagram of Module 5: Chat and Messaging

(Website: <https://lucidchart.com>)

10 Functional Requirements

Content

A specification for each functional requirement. As with all types of requirements, use the requirements shell. A full explanation is included in this template's introductory material.

Motivation

The functional requirements for Snapchat are designed to create a rich and enjoyable experience for users. They make sure that people can securely access their accounts and personalize their profiles, while also making it easy to share photos and videos. By offering creative filters and effects, Snapchat lets users add their personal touch to their content. The ability to view friends' stories and chat in real time keeps the social aspect of the app vibrant and engaging. Safety is also a priority, with features that allow users to report inappropriate content and tools for administrators to manage the platform effectively. Together, these features help make Snapchat a fun, creative, and safe place for social interaction.

Examples

Requirement: 01

Requirement Type: Event/Use Case : 1, 2, 3

Description: The system shall allow users to sign up and log in to their accounts.

Rationale: To provide access to personalized features and user-specific data.

Originator: Product Team

Fit Criterion: The user must be able to successfully create a new account or log in with valid credentials. Login attempts with incorrect credentials should prompt an error message.

Customer Satisfaction: 5

Customer Dissatisfaction: 2

Priority: High

Requirement: 02

Requirement Type: Event/Use Case : 4, 5

Description: The system shall allow users to capture and send photos and videos.

Rationale: To enable users to share visual content with their friends and followers.

Originator: Product Team

Fit Criterion: Users must be able to capture media using the app's camera and successfully send it to selected contacts. Any failures in sending or capturing media should be reported with appropriate error messages.

Customer Satisfaction: 4

Customer Dissatisfaction: 3

Priority: High

Requirement: 03

Requirement Type: Event/Use Case : 6, 7

Description: The system shall support the application of filters and effects to photos and videos.

Rationale: To enhance user engagement by allowing customization of shared media.

Originator: Product Team

Fit Criterion: The user should be able to apply selected filters and effects and preview them before saving or sending. If the application of filters fails, the system should provide feedback.

Customer Satisfaction: 4

Customer Dissatisfaction: 2

Priority: Medium

Requirement: 04

Requirement Type: Event/Use Case : 8

Description: The system shall allow users to view stories posted by friends and other users.

Rationale: To provide a way for users to view and interact with content shared by their network.

Originator: Product Team

Fit Criterion: Users must be able to navigate through and view available stories. Stories that are no longer available should be removed or marked as expired.

Customer Satisfaction: 5

Customer Dissatisfaction: 1

Priority: High

Requirement: 05

Requirement Type: Event/Use Case : 9, 10

Description: The system shall provide users with real-time messaging capabilities, including the ability to send and receive text and multimedia messages.

Rationale: To facilitate instant communication between users.

Originator: Product Team

Fit Criterion: Messages should be delivered and received in real time. Any delays or failures should be reported to the user.

Customer Satisfaction: 5

Customer Dissatisfaction: 3

Priority: High

Requirement: 06**Requirement Type:** Event/Use Case : 11**Description:** The system shall allow users to report inappropriate content or behavior.**Rationale:** To maintain a safe and respectful environment on the platform.**Originator:** Product Team**Fit Criterion:** Users should be able to submit reports with reasons for the report. Reports should be logged and flagged for review by admins.**Customer Satisfaction:** 4**Customer Dissatisfaction:** 2**Priority:** Medium**Requirement:** 07**Requirement Type:** Event/Use Case : 7**Description:** The system shall provide administrative functionalities for managing user accounts and content.**Rationale:** To allow admins to oversee and manage user activity, content, and platform integrity.**Originator:** Admin Team**Fit Criterion:** Admins must have access to manage user accounts, review content reports, and perform necessary moderation actions. Any issues in accessing or performing administrative tasks should be reported with feedback.**Customer Satisfaction:** 3

Customer Dissatisfaction: 4

Priority: High

11 Data Requirements

Content

For Snapchat, the essential subject matter and business objects can be represented through a simplified domain model or class diagram. The primary entities include:

1. User

- Attributes: UserID, Username, Email, PasswordHash, ProfilePicture, DateOfBirth, etc.
- Relationships:
 - **Friends:** Each user can have multiple friends.
 - **Stories:** Each user can create multiple stories.
 - **Messages:** Each user can send and receive multiple messages.

2. Story

- Attributes: StoryID, UserID, MediaContent, Timestamp, Visibility (public/private), ExpirationTime`
- Relationships:
 - **User:** Each story is associated with one user.
 - **Comments:** Stories can have multiple comments.
 - **Views:** Stories can be viewed by multiple users.

3. Message

- Attributes: MessageID, SenderID, ReceiverID, MessageContent, MediaContent, Timestamp, MessageType (text/image/video)
- Relationships:
 - **Sender:** Each message is sent by one user.

- **Receiver:** Each message is received by one user.

4. Filter

- Attributes: FilterID, FilterName, FilterType, EffectParameters
- Relationships:
 - **Stories/Media:** Filters are applied to media content in stories or messages.

5. Report

- Attributes: ReportID, ReporterID, ReportedContentID, ReportReason, Timestamp
- Relationships:
 - **Reporter:** Each report is made by one user.
 - **Reported Content:** Each report is associated with specific content.

6. Admin

- Attributes: AdminID, Username, Role, Permissions
- Relationships:
 - **Moderation:** Admins have the ability to review and act on reports and manage user accounts.

Motivation

Defining these data requirements clarifies the core components and relationships within Snapchat, ensuring that all necessary aspects of the platform are considered. By establishing a clear model of users, stories, messages, filters, reports, and administrative functions, this framework helps identify any additional requirements or dependencies that may need to be addressed. This comprehensive understanding of the data structure supports effective system

design, development, and integration, ensuring that all critical functionalities are properly supported.

12 Performance Requirements

12a Speed and Latency Requirements

Content

Response Times: Defines a time limit for Snapchat's response to user interactions. As an illustration:

Sending/Receiving of Snaps: The duration of time required for a Snap to be transmitted by a single user, received by the server, and shown to the intended recipient.

Story Loading: The amount of time it takes for a user-tapped story to begin playing.
Operating Speed: Establishes how quickly Snapchat should do different tasks.

AR Effects & Filters Loading: The speed at which augmented reality effects and filters load and apply to Snaps. The ease with which users may navigate between the app's various parts, such as going from Snaps to Stories or Discover, is known as navigation.

Lateness Tolerance: outlines acceptable wait times for data processing:
Snap Processing: The longest time that can be allowed between a snap being taken and its processing and display.

Content Updates: The Discover section's content has been updated and refreshed thanks to the delay.

Motivation

Speed and latency requirements are critical for Snapchat because:

- **Real-Time Interaction:** Because Snapchat is a real-time platform, users anticipate receiving responses right away. When a Snap is sent, for instance, it ought to arrive and be visible to the recipient right away. If the app doesn't live up to these expectations, users may become less engaged and have a disappointing experience.
- **User Experience:** Maintaining users' engagement and satisfaction requires quick response times and seamless operation. Reduced app usage and user satisfaction can result from slow performance or high latency, which can have a negative effect on how users interact with features like filters, Stories, and Discover.
- **High Volume Usage:** Every day, Snapchat manages a large number of interactions and uploads of content. To properly manage this load without experiencing performance deterioration, efficient performance is required. Delays in applying filters or loading Stories, for example, may affect how easy it is to use and how comfortable the app is for users as a whole.

Examples

- **Reaction Time**
 - **User activities:** To guarantee a seamless user experience, Snapchat's response time to user activities, such as sending or receiving a Snap, cannot be longer than two seconds.
 - **Story Playback:** When a user taps on the app, a story will begin to play within two seconds of the app loading.

- **Experience of the User**
 - **Seamless Interaction:** The application should operate at a speed that doesn't interfere with the user's train of thought, ensuring a seamless and captivating experience devoid of appreciable lag.
- **Data Update:**
 - **Content Updates:** In order to keep users up to date with the newest stuff, Snapchat will refresh the Discover section with new content or status updates every five minutes.

Fit Criterion

- **Response Time:** In 90% of situations, Snapchat will react to user events (such as sending a Snap or loading a Story) in less than a second. To preserve an excellent user experience, no response time should be longer than 2.5 seconds.

Considerations

- **Real-Time Interactions:** Quick response times are crucial for real-time features that require speed.
- **Snap Sending/Receiving:** In order to guarantee prompt user communication, the program must send Snaps within two seconds.
- **Story Playback:** To keep viewers interested and avoid annoyance, stories should begin to play in two seconds or less.
- **Feature Loading:** Features that improve user experience also depend on performance.
- **AR Effects:** To keep users interested and satisfied, filters and AR lenses should load and apply in no more than two seconds.
- **Navigation:** To prevent interfering with the user experience, transitions between other app areas (such as from Snaps to Stories) should be seamless and swift.

- **Content changes:** For features that receive less changes frequently, speed is less important.

Find Content Although updates should happen every five minutes, the precise timing is less critical than interactions that happen in real time, but it's still crucial to maintain content original.

Customization

- **High-Speed Requirements:** To increase user engagement, make sure that feature loading and real-time interactions adhere to strict speed requirements.
- **Moderate-Speed Requirements:** Give content updates top priority, but permit a little bit longer reaction times than those of real-time features.

12b Precision or Accuracy Requirements

Content

- **Data Accuracy:** Indicates the level of accuracy needed for data that Snapchat processes or displays. As an illustration:
- **Image Quality:** High fidelity rendering is necessary to preserve original quality in photographs and movies while minimizing compression artifacts.
- **Location Data:** To guarantee accurate geotagging of Snaps and Stories, Snapchat should use location services with an accuracy of ± 10 meters.

Motivation

User Expectations: To make sure that consumers know exactly what to anticipate in terms of the precision and correctness of the data and content on Snapchat. Reliable

information and excellent media support a better user experience and increase platform confidence.

Examples

- **Image Resolution:** To ensure that material appears crisp and clear, photos and videos taken with Snapchat and shared should retain the resolution accuracy suitable for the device's screen.
- **Location Accuracy:** To guarantee optimal performance and user satisfaction, Snapchat must use location-based features (such as geofilters) with location data that is accurate to within ± 10 meters.

Considerations

- **Units and Definitions:** Verify that the app's definitions and units correspond to the precise criteria. Make sure, for example, that location units (such as meters for precise position) and image resolution are used consistently.
- **Synchronization:** Take into account whether features such as Stories and Snap metadata require the app to operate in a certain time zone (such as UTC) or synchronize time stamps with a time server.
- **Dollar amounts with decimal places:** Make sure that all monetary amounts are accurate to two decimal places or consistent with the standard of the relevant currency if Snapchat integrates any kind of in-app purchases or financial transactions.

12c Capacity Requirements

Content

- **User Volume:** Indicates how many users Snapchat can accommodate at once without seeing a drop in speed.
- **Users at the Same Time:** Snapchat should be able to support up to 10 million active users at once, particularly during busy periods like holidays or popular occasions.
- **Data Storage:** Describes how much data Snapchat needs to handle and store.
- **Media Storage:** In order to ensure that its storage systems can withstand constant uploads and retrievals, Snapchat should be able to store a substantial amount of media content, including billions of Snaps, Stories, and videos.

Motivation

Processing Capacity: To guarantee that Snapchat can manage significant user activity and substantial data quantities, particularly during moments of peak usage. This supports preserving user pleasure and app performance.

Examples

- **User Load:** When demand is at its highest (such as during big events or the debut of new features), Snapchat should be able to handle up to 10 million concurrent users. This figure may be lower during off-peak hours, but seamless functioning must still be guaranteed.
- **Data handling:** Snapchat must effectively manage and store massive volumes of media content without compromising performance during periods of high usage, such as during significant events or viral trends.

Fit Criterion

User Load: The application must continue to operate at a reasonable pace.

13 Dependability Requirements

13a Reliability Requirements

Content

- **Failure Rate:** Indicates the frequency of Snapchat failures while maintaining user satisfaction and operational standards.
- **Failure Frequency:** In order to guarantee users' continuous availability and performance, Snapchat should not encounter more than one major failure every day.
- **Data integrity:** Makes sure that, even in the case of a malfunction, user data is secure and unaltered.
- **Fail-Safe Requirement:** No user data (such as chat history, stories, or snaps) shall be lost or corrupted in the event of a system failure. Mechanisms for data recovery and integrity maintenance should be included in the system.

Motivation

- **Consistency and Trust:** Preserving user pleasure and trust on Snapchat requires consistent operation. To avoid frequent disruptions that can impair user experience and engagement, high reliability is required.
- **Expectation management:** Giving customers and users a clear understanding of the frequency and potential consequences of failures can help them have reasonable expectations. It also directs Snapchat's strategy for development and upkeep.

Examples

- **Failure Frequency:** In order to maintain high availability, Snapchat should be built to withstand any malfunctions or downtime. However, it should not malfunction more than once a day.
- **Data Integrity:** Snapchat needs to make sure that all user data is saved and not destroyed or lost, even in the event of a failure. To do this, reliable backup and recovery solutions must be put in place.\

Considerations

Failure vs. Availability:

- **High Availability:** Because downtime undermines user trust, Snapchat needs to focus on being always accessible rather than just cutting down on errors.
- **Failure Recovery:** To ensure continuous service, there should be a priority on speedy recovery.

Dependability vs. Cost:

- **Cost Management:** While investing in dependability is essential for consumer pleasure, it must be weighed against other corporate requirements.

13b Availability Requirements

Content

- **Availability:** Snapchat should aim for 99.9% uptime and remain accessible around-the-clock to minimize user disturbances.
- **Factors Affecting Downtime:** The frequency of system breakdowns, the pace at which they recover, and the amount of time required for regular maintenance all affect availability.

Motivation

User Expectations: Because Snapchat is a real-time social media platform, users expect the app to be available whenever they want to use it. Therefore, high availability is critical.

Examples

- Snapchat ought to be up and running 99.9% of the time.
- To sustain this high availability, any failure in the system should be swiftly recovered from.

Considerations

- **Cost vs. Benefit:** Consider how important it is to minimize user downtime versus the costs associated with attaining near-constant availability.
- **Combined Requirements:** Because of their close relationship, reliability and availability can occasionally be addressed jointly.

The sections on reliability and availability can sometimes be combined.

13c Robustness or Fault-Tolerance Requirements

Content

Robustness specifies the ability of the product to continue to function under abnormal circumstances.

Motivation

To ensure that the product is able to provide some or all of its services after or during some abnormal happening in its environment.

Examples

The product shall continue to operate in local mode whenever it loses its link to the central server.

The product shall provide 10 minutes of emergency operation should it become disconnected from the electricity source.

Considerations

Abnormal happenings can almost be considered normal. Today's products are so large and complex that there is a good chance that at any given time, one component will not be functioning correctly. Robustness requirements are intended to prevent total failure of the product.

You could also consider disaster recovery in this section. This plan describes the ability of the product to reestablish acceptable performance after faults or abnormal happenings.

13d Safety-Critical Requirements

Content

Robustness: In the event of anomalous conditions, like server outages or network problems, Snapchat must be able to continue operating. For instance, users should still be able to interact with local information or cached functionality if the app loses access to the central server.

Motivation

Service Continuity: Preserving user trust and avoiding complete service failure require Snapchat to be able to continue offering essential services during or after disruptions.

Examples

- If Snapchat's connection to the central server is lost, it should continue to function in limited mode.
- If the app encounters connectivity problems, it should continue to work for a brief while.

Fit Criterion

- **Compliance:** Snapchat is required to adhere to certain performance requirements under unusual circumstances, such as continuing to offer a restricted range of features in the event of network or server outages.
- **Testing and Certification:** To make sure the app satisfies these robustness requirements, it should be certified by certified testing engineers. One way to evaluate Snapchat might be to see if it can continue to function for a predetermined amount of time after losing server connectivity.

Considerations

- **Handling Abnormalities:** Because Snapchat's system is so intricate, it's possible that any one of its many parts could malfunction at any time. Requirements for robustness are designed to reduce the effect of these failures on overall functionality.

- **Disaster Recovery:** Make a plan for getting back up and running after major errors or interruptions.

14 Maintainability and Supportability Requirements

14a Maintenance Requirements

Content

Change Implementation Time: Indicates how long it will take to update or modify Snapchat, including adding new features and repairing bugs.

Motivation

Clarity on Maintenance Needs: Establishes expectations for change management by ensuring that all parties involved are aware of the time and resources required for continuous maintenance.

Examples

- **New Features:** Within a month of the requirements being finalized, new features or major changes should be produced and made available.
- **Bug Fixes:** Within a week of discovery, critical bugs must be located and fixed.

Considerations

- **Maintainability:** Determine whether end users require some degree of self-service maintenance or whether Snapchat should be maintainable by people who were not engaged in its initial development. Design and development processes may be impacted by this.
- **Training and Documentation:** Make sure thorough training materials and documentation are available to support maintenance tasks, particularly for non-original developers.

- **Testability:** Establish testability standards to guarantee that modifications may be extensively examined and approved prior to implementation.

14b Supportability Requirements

Content

Support Level: Indicates the kind and extent of support that Snapchat needs, as well as whether it should be integrated into the app or offered through a help desk.

Motivation

Ensuring that the support requirements are well-defined and encompass all essential elements to properly maintain the application and aid users is known as adequate specification.

Considerations

- **Support Forms:** Describe the types of assistance that Snapchat will provide. Will in-app support features, a help desk, or both be used for assistance.
- **Self-Support:** Evaluate whether Snapchat should be made to be as self-sufficient as possible to reduce the need for outside assistance or printed guides.
- **Expected Support Amounts:** Based on user expectations and possible problems, estimate the projected support needs and make sure there are enough support mechanisms in place to handle them.

14c Adaptability Requirements

Content

Platform Support: Indicates which environments or platforms Snapchat needs to be ported or adjusted for. Information on the devices, operating systems, and markets that the app should support or be able to adapt to in the future is included in this.

Motivation

User and Market Expectations: Guarantees that Snapchat can satisfy users' and clients' expectations in terms of platform compatibility and possible future settings, enabling the growth of its user base and market share.

Examples

The most recent iterations of the iOS and Android operating systems must be compatible with Snapchat. Future upgrades ought to guarantee that Snapchat runs without a hitch on cutting-edge hardware and operating systems. It should be possible for Snapchat to localize and adapt to new markets; for example, it may include features that are appropriate for the Japanese market.

Fit Criterion

- **System Software:** Specify the versions and operating systems that Snapchat needs to run on (e.g., iOS 16 and Android 14).
- **Future Environments:** List any platforms or environments that you expect Snapchat will operate in the future.
- **Transition schedule:** Describe how long it will take Snapchat to adjust to new surroundings or platforms, making sure that updates and support are provided on schedule.

Considerations

- **Marketing Assumptions:** Get in touch with the marketing division to find out whether there are any unspoken presumptions regarding the mobility and

adaptability of Snapchat. This involves being aware of the future market growth and the particular needs for certain geographical areas or gadget kinds.

- **Prospect-Making:** In order to ensure that Snapchat stays relevant and useful across a variety of platforms and locations, plan for future flexibility by taking technical improvements and changing user needs into consideration.

14d Scalability or Extensibility Requirements

Content

Capacity growth refers to the anticipated rise in the number of users, volume of data, or transaction load that Snapchat will need to manage as it expands. This guarantees that future growth of the program won't result in performance decrease.

Motivation

Future-proofing: Ensures that Snapchat is built to expand and adapt to changing needs in order to meet user expectations and commercial expansion.

Examples

- **User Base Growth:** Within the first three years, Snapchat should be able to handle 100 million active users, and the system should be scalable to 300 million users after that.
- **Data Volume:** As user content increases, Snapchat should be able to handle a rise in data volume, storing and processing up to 10 terabytes of user-generated content, with the option to grow to 50 terabytes.
- **Transaction Load:** Within two years, the infrastructure should be scalable to handle five million interactions per hour, and the app should be able to process up to one million interactions (messages, snaps, etc.) each hour.

Fit Criterion

- **User Base Handling:** Indicate the initial and projected user capacity that the system must accommodate, together with the associated infrastructure requirements and anticipated growth rates.
- **Data Handling:** Describe the present and future needs for data storage, including the capacity to accommodate growing data quantities without compromising system performance.
- **Transaction Processing:** Describe the present processing capabilities and scalability targets, as well as the required transaction throughput.

Considerations

- **Infrastructure:** Make plans for scalable databases, load balancing systems, cloud services, and other infrastructure that can adapt dynamically to growing loads.
- **Performance Testing:** To confirm that Snapchat can manage anticipated growth and spot possible bottlenecks, do performance and stress testing.
- **Future Improvements:** Build the system with extensibility in mind so that it can accommodate new features or upgrades in the future without needing to be completely redesigned.

14e Longevity Requirements

Content

Expected Lifetime: Specifies how long Snapchat is expected to continue operating, taking into account things like upkeep, updates, and technology improvements.

Motivation

Return on Investment: Guarantees that Snapchat is designed with a thorough grasp of its anticipated long-term maintenance and operating expenses, in line with the organization's financial and strategic objectives.

Examples

- **Operational Longevity:** In order to guarantee continuous functionality and user interest, Snapchat should be built to last at least five years, with upgrades and maintenance offered.
- **Maintenance Budget:** The application must be supported within the allotted budget for maintenance, which covers bug fixes, feature updates, and making sure the program is compatible with new hardware and operating systems for the duration of its existence.

Fit Criterion

- **Budget for Maintenance:** Establish the highest amount that may be allocated for upkeep and assistance, and make sure that Snapchat stays within this limit while fulfilling its operational and performance standards for the duration of the app's anticipated lifespan.
- **Feature Updates:** Indicate how often and what kinds of updates—such as significant version upgrades and small fixes—are required to maintain Snapchat's functionality and relevance over time.
- **Technology Adaptation:** To guarantee Snapchat stays operational and competitive, incorporate standards for adjusting to technology changes, such as new operating systems or gadgets.

Considerations

Technological Evolution: As technology develops and consumer expectations shift, prepare for frequent upgrades and possibly even a redesign.

15 Security Requirements

15a Access Requirements

Content

- **Authorized Access:** Specifies who has access to what information and services on Snapchat. For example, regular users can only view their own personal stuff, whereas administrators can control user accounts and backend data.
- **Conditions of Access:** Describes the requirements that must be met in order to provide access, such as the need for multi-factor authentication or login credentials in order to access sensitive functionality.
- **Accessibility:** Indicates which areas of Snapchat are available to specific user roles. For instance, while developers or administrators have access to system settings and analytics, regular users can only view their own messages and tales.

Motivation

Confidentiality: Limits access to Snapchat's features and sensitive data to those who are permitted, according to their positions and responsibilities. This preserves user confidence and protects private data.

Examples

Only the Snap history is accessible to verified users. Administrators are the only ones with access to system-level settings and statistics.

Fit Criterion

- **User Roles:** Define distinct user roles (such as admin and regular user) and assign access levels accordingly.
- **Authorization:** Make sure that, according to preset user roles and permissions, access to particular system functions or data is properly regulated.

Considerations

- **Sensitive Information:** To avoid unauthorized access and security breaches, identify and safeguard important information such as user profiles, private communications, and account details.
- **Security Risks:** By putting strong security controls and policies in place, you may mitigate risks like unauthorized internal access or external attacks.
- **Specialist Consultation:** Considering how complicated security standards might be, speaking with a security specialist will help guarantee that Snapchat has adequate and efficient security measures in place.

15b Integrity Requirements

Content

Data integrity keeps false or corrupt data out of the system and guarantees that Snapchat keeps accurate and consistent data.

Protection Against Abuse: Preserves the app's functionality and data integrity by protecting it against deliberate misuse, such as alteration or malicious activity.

Motivation

Data Integrity: To guarantee that, even in the face of outside threats or inadvertent usage by authorized users, Snapchat's data is accurate and trustworthy.

Examples

The app will have safeguards in place to stop inaccurate or faulty data entry.

Put safeguards in place to prevent deliberate abuse or manipulation.

Considerations

Data Dependency: Because Snapchat relies heavily on its data, precautions must be taken to avoid data loss or corruption, which might negatively affect user confidence and the company's ability to operate.

15c Privacy Requirements

Content

Data integrity keeps false or corrupt data out of the system and guarantees that Snapchat keeps accurate and consistent data.

Protection Against Abuse: Preserves the app's functionality and data integrity by protecting it against deliberate misuse, such as alteration or malicious activity.

Motivation

Data Integrity: To guarantee that, even in the face of outside threats or inadvertent usage by authorized users, Snapchat's data is accurate and trustworthy.

Examples

The app will have safeguards in place to stop inaccurate or faulty data entry.

Put safeguards in place to prevent deliberate abuse or manipulation.

Considerations

- **Legal Advice:** Seek advice from legal professionals to make sure privacy laws are followed and to establish proper privacy procedures.

- **User Consent:** Make sure users are aware of and able to agree to procedures for the collection and storage of data. Users ought to have the ability to view and update their personal information as needed.
- **Data Security:** To avoid unwanted access and security breaches, maintain strict security standards for sensitive data, such as payment information.

15d Audit Requirements

Content

Record Retention: outlines Snapchat's plan for keeping records around in order to facilitate efficient audit reviews. This include keeping track of system access, data modifications, and user activity logs.

Motivation

Compliance: To make sure Snapchat complies with audit requirements and is able to be examined for internal policy and pertinent regulatory compliance.

Consideration

- **Legal Approval:** Verify the specifications for record keeping and audit capabilities by speaking with auditors or legal professionals.
- **User Accountability:** To prevent denial of usage or transactions, include methods within the app to track and save information on user behaviors. This will support accountability and transparency.

15e Immunity Requirements

Content

Malware Protection: outlines the precautions Snapchat has to take against unwanted malware, such as Trojan horses, worms, and viruses.

Motivation

Enhanced Security: To safeguard user data and preserve the integrity of the app while keeping Snapchat safe from harmful assaults and unwanted intervention.

Considerations

Growing Threats: In order to guard against fresh and new security threats, Snapchat must constantly update its defenses as malware threats change.

16 Usability and Humanity Requirements

Snapchat should be all about being easy and fun for everyone to use. The app needs to be simple to navigate so people can quickly find and enjoy features like messaging and filters without any hassle. It should let users personalize their experience, like customizing their profiles or picking favorite filters, to make it feel more personal and engaging. It's also important that Snapchat is accessible to everyone, including people with disabilities, by having options like voiceover support and bigger text. The app should work smoothly and be fast, providing a consistent experience on any device, and give clear feedback if something goes wrong, so users always feel at ease and enjoy their time on the app.

16a Ease of Use Requirements

Content

This section outlines how easy it should be for users to operate Snapchat. The app's usability is based on the abilities of the expected users and the complexity of its features.

The usability requirements for Snapchat should include the following properties:

- **Efficiency of Use:** Users should be able to navigate the app quickly and perform tasks like sending snaps or using filters accurately and with minimal effort.
- **Ease of Remembering:** Casual users should find it easy to remember how to use Snapchat after taking a break from it. The app's design should be intuitive, with familiar icons and consistent actions that make it easy to recall how to use key features.
- **Error Rates:** Snapchat should be designed to minimize errors. The app should help users avoid making mistakes, such as sending a snap to the wrong person or accidentally deleting a story.
- **Overall Satisfaction:** Users should find Snapchat enjoyable and engaging, encouraging them to use the app regularly. A positive user experience is crucial, especially in a competitive market for social media apps.
- **Feedback:** The app should provide clear and immediate feedback to users' actions. This could include confirmations when a snap is sent or notifications when a message is read, helping users feel confident that the app is functioning as expected.

Motivation

These requirements are intended to guide the design and development of Snapchat to ensure it meets the needs and expectations of its users, making the app userfriendly, efficient, and enjoyable.

Examples

- The app shall be easy for teens and young adults to use without prior training.
- The app shall help users avoid making accidental actions, such as sending snaps to the wrong person.
- The app shall encourage users to explore and engage with its features regularly.
- The app shall be usable by people with no technical background.

Fit Criterion

- Eighty percent of a test group of 15 to 25 year olds shall be able to successfully complete a list of common tasks (like sending a snap or adding a filter) within three minutes.
- After one month of use, the app shall have an average error rate of less than 2% for unintentional actions.
- An anonymous survey shall show that 75% of users are regularly using the app after a two-week familiarization period.

16b Personalization and Internationalization Requirements

Content

This section is all about making Snapchat feel like it's built just for you, no matter where you're from or what your preferences are. To do this, the app should offer:

- **Language and Spelling Options:** Snapchat should be available in multiple languages, so you can choose the one you're most comfortable with. Whether you speak English, Spanish, French, or any other language, you should be able to navigate the app easily in your preferred language.
- **Local Currency and Number Formats:** When you make a purchase or see a price, it should be in your local currency and format. This means you won't have to do mental math or get confused by unfamiliar symbols—everything will be in a format you know.
- **Customization Choices:** You should be able to set up Snapchat just the way you like it. This includes saving your favorite filters, deciding how you want to get notifications, and organizing your chats and stories in a way that makes sense to you. This helps make the app feel like your own unique space.

Motivation

The goal is to make Snapchat welcoming and easy to use for everyone, no matter where you are in the world or what language you speak. By letting you personalize your experience and adapting to different cultural preferences, Snapchat aims to make everyone feel included and valued.

Examples

- You should be able to set your preferred language so that all menus, settings, and messages appear in that language.
- When buying filters or other inapp items, prices should be shown in your local currency to avoid confusion.
- The app should remember your choices, like your preferred filters and settings, so you don't have to adjust them every time.

Considerations

When designing Snapchat, it's important to remember that users come from all over the world with different languages and cultural backgrounds. By offering customization options and supporting various languages and formats, Snapchat can make each user feel more at home and comfortable. It's all about giving everyone a chance to enjoy the app in their own way, making the experience fun and accessible for all.

16c Learning Requirements for Snapchat

Content

For Snapchat, learning requirements focus on how easily users can pick up and start using the app. Given its broad user base, from techsavvy teens to casual users unfamiliar with social media apps, Snapchat should be intuitive and straightforward.

The learning curve should be minimal to none, ensuring that new users can quickly understand and enjoy the app without needing detailed instructions or training.

Motivation

The goal is to minimize the time it takes for users to become comfortable with Snapchat. This means that as soon as someone downloads the app, they should be able to start using its main features—like sending snaps, viewing stories, and exploring filters—almost immediately. By making the app easy to learn, Snapchat can improve user satisfaction and engagement, ensuring that people keep coming back to use it.

Examples

- The app should be easy enough for a new user to learn how to send a snap within a few minutes of downloading it.
- A user who is not familiar with social media should be able to view and interact with stories after a short initial exploration of the app.
- All key features, such as adding friends or using filters, should be discoverable and usable without needing any tutorial or external guidance.

Fit Criterion

- New users should be able to successfully send their first snap within 5 minutes of opening the app for the first time.
- At least 90% of users in a usability test should be able to find and use the filters feature within 3 minutes.
- A casual user should be able to add a friend and view their story within 10 minutes of starting to use Snapchat.

Considerations

It's important to consider all the different types of users when designing Snapchat. This includes not only young, tech savvy users but also those who might not be as familiar with using apps. The design should cater to a broad audience, making sure that everyone finds the app easy to learn and enjoyable to use from the moment they start.

16d Understandability and Politeness Requirements

This section is concerned with discovering requirements related to concepts and metaphors that are familiar to the intended end users.

Content

This specifies the requirement for the product to be understood by its users. While “usability” refers to ease of use, efficiency, and similar characteristics, “understandability” determines whether the users instinctively know what the product will do for them and how it fits into their view of the world. You can think of understandability as the product being polite to its users and not expecting them to know or learn things that have nothing to do with their business problem.

Motivation

To avoid forcing users to learn terms and concepts that are part of the product’s internal construction and are not relevant to the users’ world. To make the product more comprehensible and thus more likely to be adopted by its intended users.

Examples

The product shall use symbols and words that are naturally understandable by the user community. The product shall hide the details of its construction from the user.

Considerations

Refer to section 3, Users of the Product, and consider the world from the point of view of each of the different types of users.

16e Accessibility Requirements

Content

The requirements for how easy it should be for people with common disabilities to access the product. These disabilities might be related to physical disability or visual, hearing, cognitive, or other abilities.

Motivation

In many countries it is required that some products be made available to the disabled. In any event, it is self-defeating to exclude this sizable community of potential customers.

Examples

The product shall be usable by partially sighted users.

The product shall conform to the Americans with Disabilities Act.

Considerations

Some users have disabilities other than the commonly described ones. In addition, some partial disabilities are fairly common. A simple, and not very consequential, example is that approximately 20 percent of males are red-green colorblind.

16f User Documentation Requirements

Content

List of the user documentation to be supplied as part of the product.

Motivation

To set expectations for the documentation and to identify who will be responsible for creating it.

Examples

- Technical specifications to accompany the product.
- User manuals.
- Service manuals (if not covered by the technical specification).
- Emergency procedure manuals (e.g., the card found in airplanes).
- Installation manuals.

Considerations

Which documents do you need to deliver, and to whom? Bear in mind that the answer to this questions depends on your organizational procedures and roles.

For each document, consider these issues:

- The purpose of the document
- The people who will use the document
- Maintenance of the document

16g Training Requirements

Content

A description of the training needed by users of the product.

Motivation

To set expectations for the training. To identify who is responsible for creating and providing that training.

17 Look and Feel Requirements

17a Appearance Requirements

Content

Branding and Aesthetics: outlines Snapchat's specifications for visual design, such as conformity to company branding, color palettes, and general design components.

Motivation

Consistency: Makes ensuring Snapchat's design adheres to brand guidelines and appeals to its intended demographic, resulting in a unified and captivating user experience.

Examples

Snapchat's demographic, which is predominantly younger, should find it visually appealing.

The app needs to adhere to the corporate branding criteria set forth by Snapchat.

Fit Criterion

- **User Engagement:** Four minutes after downloading the app for the first time, a representative sample of teenagers should be using it.
- **Branding Compliance:** The appropriate department must confirm that the design complies with Snapchat's branding guidelines.

Considerations

Design Alignment: It is essential to have a clear idea of appearance needs early on in the design process to ensure that prototypes live up to expectations.

17b Style Requirements

Content

- **Mood and Style:** Describes the general vibe and aesthetic of Snapchat, encompassing the interactive and visual components that shape the way users perceive the app.
- **Involvement Level:** Describes the kind and quantity of user involvement that the app is meant to encourage. When it comes to physical products, the packaging

specification should outline the dimensions, design, and coherence with other products.

Motivation

Market Expectations: Make sure Snapchat's aesthetic meets user expectations and adds to its allure. Style typically determines a product's success even when functional needs are satisfied.

Fit Criterion

Trust Perception: After using Snapchat for the first time, at least 70% of a representative sample of potential users should believe the app to be trustworthy.

Considerations

Client Vision: Style specifications should be specific enough to direct designers in attaining the intended look and feel, reflecting the client's vision for the product's appearance.

18 Operational and Environmental Requirements

18a Expected Physical Environment

Content

- **Usage Conditions:** Describes the actual physical environments in which Snapchat will be utilized, including variations in background noise and illumination.
- **Device Compatibility:** Makes certain that Snapchat works with a range of gadget sizes, including smartphones small enough to carry in a purse or pocket.

Motivation

Environmental Suitability: Makes that Snapchat works well in a variety of settings, including indoor and outdoor spaces, different lighting situations, and noisy backgrounds.

Examples

Snapchat ought to function well in both bright and low light. Regardless of the noise levels in the area, the app should work properly.

Considerations

Work Environment: Assess whether Snapchat's design has to be adjusted for particular situations, such different lighting, different noise levels, or physical locations where users may engage with the app.

18b Requirements for Interfacing with Adjacent Systems

Content

Interface Compatibility: Specifies what needs to happen in order for Snapchat to function flawlessly with other programs, hardware, and services.

Motivation

Early Identification: Rework and implementation problems are reduced when interface requirements are identified in advance.

Examples

Snapchat needs to work with the most recent iterations of Android and iOS. For content sharing, the app ought to be integrated with the main social media networks. Snapchat must establish connections with outside analytics providers in order to monitor user activity.

Fit Criterion

Data Content Indicate the kinds of data that are shared, including user interactions and media files.

Physical Material Content Specify picture formats and JSON for APIs, among other formats and standards.

- **Medium:** Explain data transfer techniques, such as REST APIs and SDKs. Specify the frequency of data exchanges, such as real-time or recurring updates.
- **Volume:** Provide data volume indicators, like the daily throughput and average file size.

18c Productization Requirements

Content

Distribution and Installation: Describes how to distribute and install Snapchat, including how to set it up, comply with app store policies, and package it.

Motivation

Setting clear expectations for time, cost, and resource allocation guarantees that the distribution and installation processes are precisely defined.

Examples

- **Distribution:** Google Play and the Apple App Store, two of the biggest app retailers, should offer Snapchat for download.
- **Installation:** Snapchat should be easy to set up so that new users may use it right away without needing any further guidance or troubleshooting.
- **Size Optimization:** To prevent problems during submission, the program should be streamlined to meet the size restrictions set by app stores.

Considerations

- **Security:** Put safeguards in place to prevent unwanted access to the app, such as secure authentication procedures and in-app purchase verification.
- **Marketing Insights:** Work together with the marketing division to ensure that consumer expectations are met by agreeing on expectations about installation timeframes and costs.
- **Commercial Needs:** Make sure the product satisfies consumer demands and market standards by attending to any particular requirements pertaining to its commercial distribution and usability.

18d Release Requirements

Content

Release Cycle: Specifies the number of times and format of new Snapchat versions (e.g., major updates, minor patches).

Motivation

Transparency: Ensures that expectations are met by keeping stakeholders aware about the nature of updates and the delivery timeline.

Examples

- **Major Updates:** Every six months, Snapchat will release new versions that include important feature additions or modifications.
- **Critical maintenance patches:** Released when necessary to fix pressing problems with the least possible interference with already-existing functionality.
- **Backward Compatibility:** Every release makes sure that the features still work with earlier iterations.

Fit Criterion

Provide a description of the maintenance, including the resources allotted for each release type and the kinds of maintenance (such as performance enhancements and bug repairs).

Considerations

- **Contractual Obligations:** Examine current maintenance contracts or agreements to make sure the new release schedule does not interfere with these commitments.
- **User Expectations:** Make sure the release cycle satisfies users' needs for continuous app security and performance, as well as for timely updates.

19 Cultural and Political Requirements

19a Cultural Requirements

- **Respect for Diversity:** The app must be sensitive to different cultures and avoid content that could offend any religious or ethnic groups.
- **Language Options:** It should support various languages and regional dialects to make the app accessible to users around the world.
- **Cultural Norms:** The design and content must align with local customs, including using appropriate symbols and colors.
- **Local Holidays:** It should recognize and optionally adjust for local public holidays to enhance relevance for users.

19b Political Requirements

- **Regulatory Compliance:** The app must follow local laws and digital policies in each country where it operates.
- **Data Storage Rules:** It should adhere to regulations about where data is stored and processed, like keeping data within specific regions.
- **Content Restrictions:** The app needs to comply with local political requirements, such as moderating content according to regional rules.

20 Legal Requirements

20a Compliance Requirements

- **Data Protection:** The app must comply with privacy laws like GDPR and CCPA to protect user data.
- **Privacy Features:** It should offer strong privacy controls, including data encryption and user consent options.
- **Intellectual Property:** The app must respect copyrights and trademarks, ensuring no infringement on others' intellectual property.

20b Standards Requirements

- **Development Best Practices:** It should follow industry standards for secure and efficient software development.
- **Accessibility:** The app must meet accessibility standards, such as WCAG, to ensure it's usable for everyone, including those with disabilities.
- **Quality Assurance:** Rigorous testing is required to meet high standards of reliability and performance.

III Design

21 System Design

21a Design goals

Content

Snapchat is a social app that's all about sharing quick, real-time moments with friends. It's designed to feel playful and natural, encouraging people to capture their day as it happens. Photos, videos, and even texts shared through Snapchat disappear after being viewed, making interactions feel a little more personal and less permanent than on other platforms. It's packed with fun features like AR filters, Bitmoji, and short "Stories" that users can add to throughout the day to keep friends in the loop.

Motivation

The heart of Snapchat's mission is to help people connect in real-time, sharing small, everyday moments without the worry of leaving a permanent record. It lets users show their unfiltered selves, express creativity through filters, and discover new things through content like news and entertainment. Snapchat is also all about keeping things fresh, giving users tools to bring their photos and videos to life with effects, doodles, and text. It's less about carefully curated posts and more about having fun and staying in touch.

Considerations

- **Privacy and Ephemerality:** Snapchat's biggest draw is that snaps disappear after being seen, which makes sharing feel more like real life. Users feel more comfortable being themselves because their posts won't live online forever.

- **Fun and Engaging Features:** Snapchat is always adding new filters, AR lenses, and interactive features to keep the experience exciting. This helps users keep in touch in a more creative way, using visuals rather than just words.
- **Ease of Use:** Snapchat's interface is designed to be simple and fast. With just a swipe or tap, users can start recording or snapping, which feels effortless and casual.
- **Security and Safety:** Because Snapchat deals with lots of temporary content, it puts extra effort into protecting users' data and privacy, especially given how quickly things are shared and disappear.
- **Continuous Innovation:** Snapchat is always rolling out new updates, whether it's adding music to Snaps, voice filters, or new lenses. The app's ever-changing features give people a reason to keep checking back in to see what's new.

Example

Imagine you're out with friends and you take a funny photo. Instead of posting it somewhere permanent, you use Snapchat to add a goofy filter and send it directly to a friend. They laugh, reply with their own snap, and within seconds, both of your photos are gone. The whole interaction feels light, funny, and in-the-moment. The "Send a Snap" feature makes it easy to share these small, everyday moments in a way that feels casual and connected, with no pressure to make it perfect.

22 Current Software Architecture

Snapchat is built with a focus on speed, simplicity, and privacy, making it easy for users to capture and share real-time moments. It uses a combination of backend servers, a frontend client, and a robust cloud infrastructure to ensure snaps are delivered quickly and then disappear as expected. Snapchat's architecture supports unique features like Stories, Lenses, and Discover while ensuring data security and privacy. By leveraging cloud storage for temporary data and a distributed network, Snapchat efficiently manages millions of snaps, chats, and content views every second.

Content

Snapchat's primary content includes Snaps (photos and videos), Stories, Bitmoji, AR lenses, and Discover, where users can view news and entertainment updates. Content in Snapchat is designed to be short, engaging, and temporary, encouraging users to check back often to see what's new. Each snap, whether it's a picture of a friend's coffee or a quick update about someone's day, reflects Snapchat's focus on spontaneity and simplicity.

Motivation

Snapchat's design and features are motivated by a desire to make social interactions feel fun, authentic, and private. The app encourages people to share moments freely, without the pressure of creating perfect, lasting posts. Snapchat's commitment to ephemeral messaging means users can express themselves in real-time, capturing their day as it unfolds. By making interactions disappear after they're viewed, Snapchat creates a safe space for sharing that mimics face-to-face interactions, keeping it lighthearted and casual.

Considerations

- **Speed and Performance:** Snapchat is designed for fast, real-time communication. This means it prioritizes lightweight content delivery and rapid processing so users don't experience delays or lag.
- **Privacy and Ephemeral:** Privacy is a central focus; snaps disappear after viewing, and this fleeting nature makes it easier for people to share freely.
- **Content Innovation:** Snapchat continually updates its features to keep users engaged, like introducing new AR lenses, Bitmoji features, and interactive games. The variety keeps the experience fresh and encourages users to keep coming back.
- **User-Friendly Interface:** The app's swipe-based navigation is intuitive and easy to learn, helping users quickly access the camera, stories, and chat features.

- **Security and Data Protection:** Since Snapchat handles personal and often sensitive media, it uses encryption and strict data protection protocols to secure user data and ensure safe, temporary content sharing.

Example

Let's say you want to send a quick update to your friends about your day. You open Snapchat, swipe to the camera, and snap a photo of your coffee. In seconds, you add a fun filter, type "Morning vibes" on it, and send it to your friends. They see it, react with their own snaps or messages, and after a moment, it's gone. This entire process is powered by Snapchat's architecture, designed to make sharing feel instant, private, and stress-free. The interaction captures Snapchat's mission to make communication simple, creative, and, most importantly, in-the-moment.

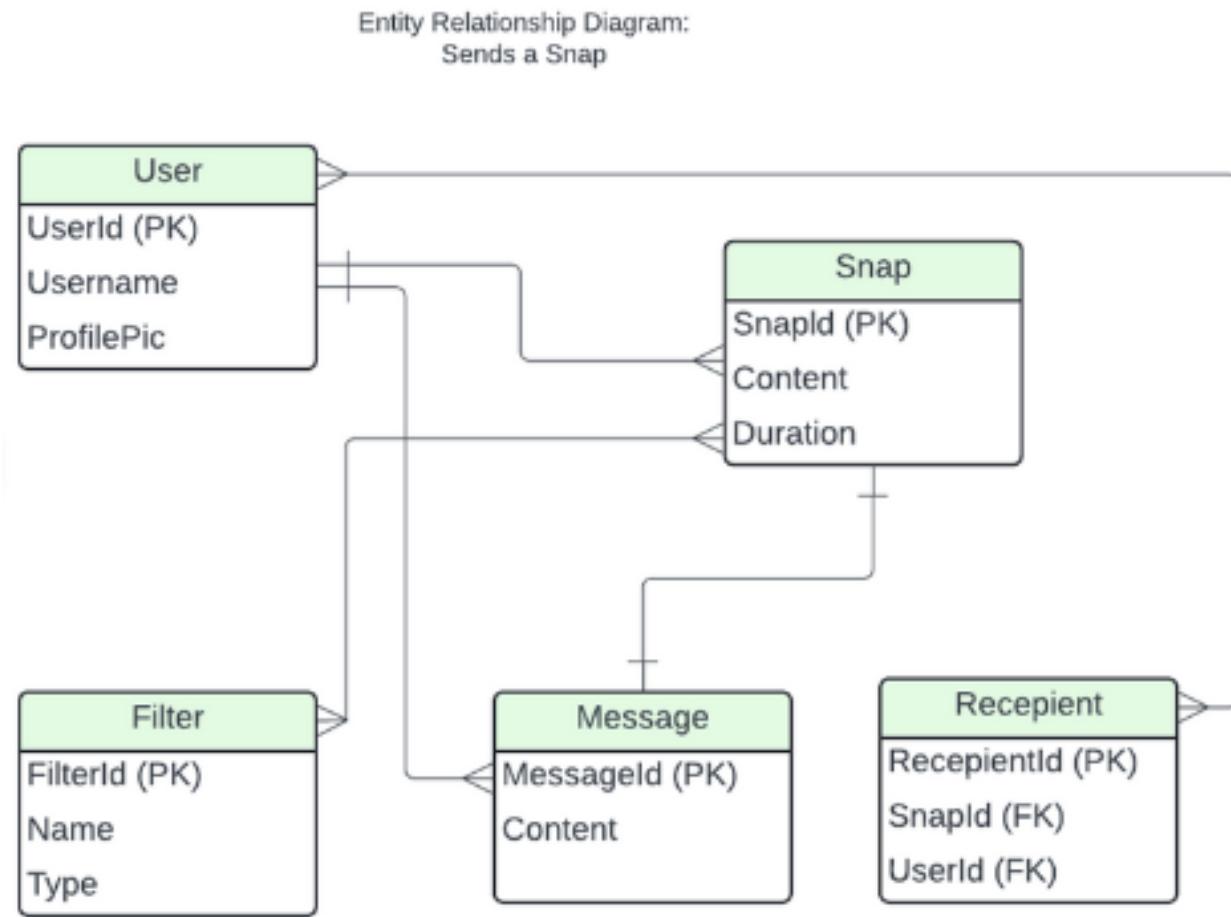


Figure 12: Entity Relationship(ER) Diagram of Primary Functionality of Snapchat

(Website: <https://lucidchart.com>)

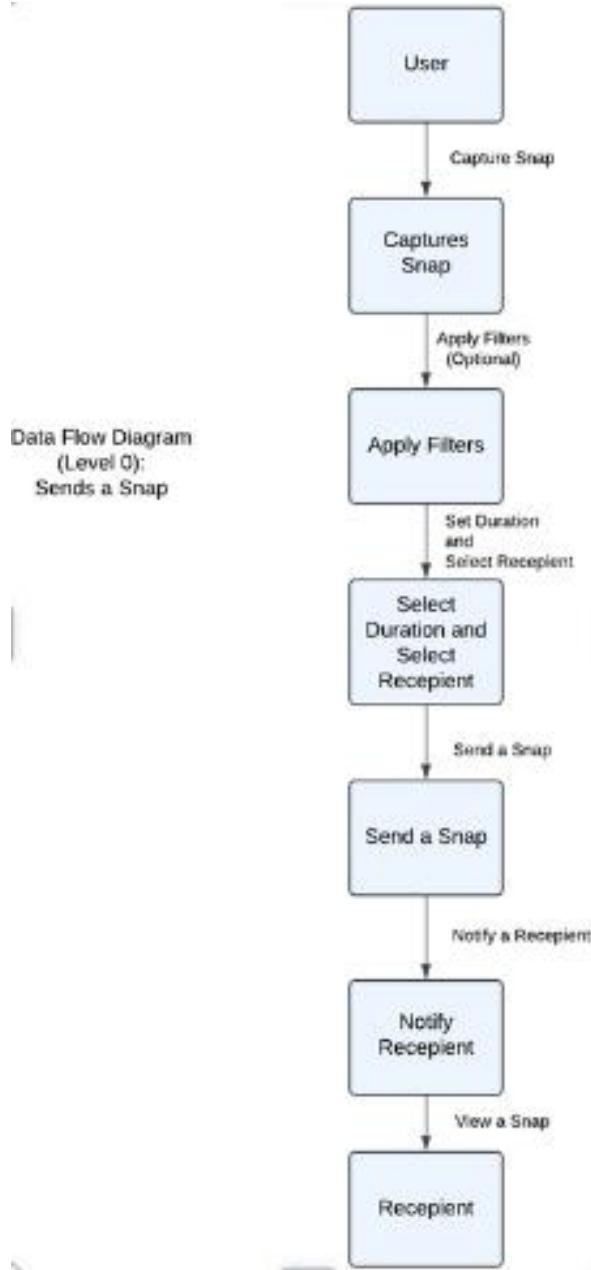


Figure 13: Data Flow Diagram(DFD) (Level 0) of Primary Functionality of Snapchat

(Website: <https://lucidchart.com>)

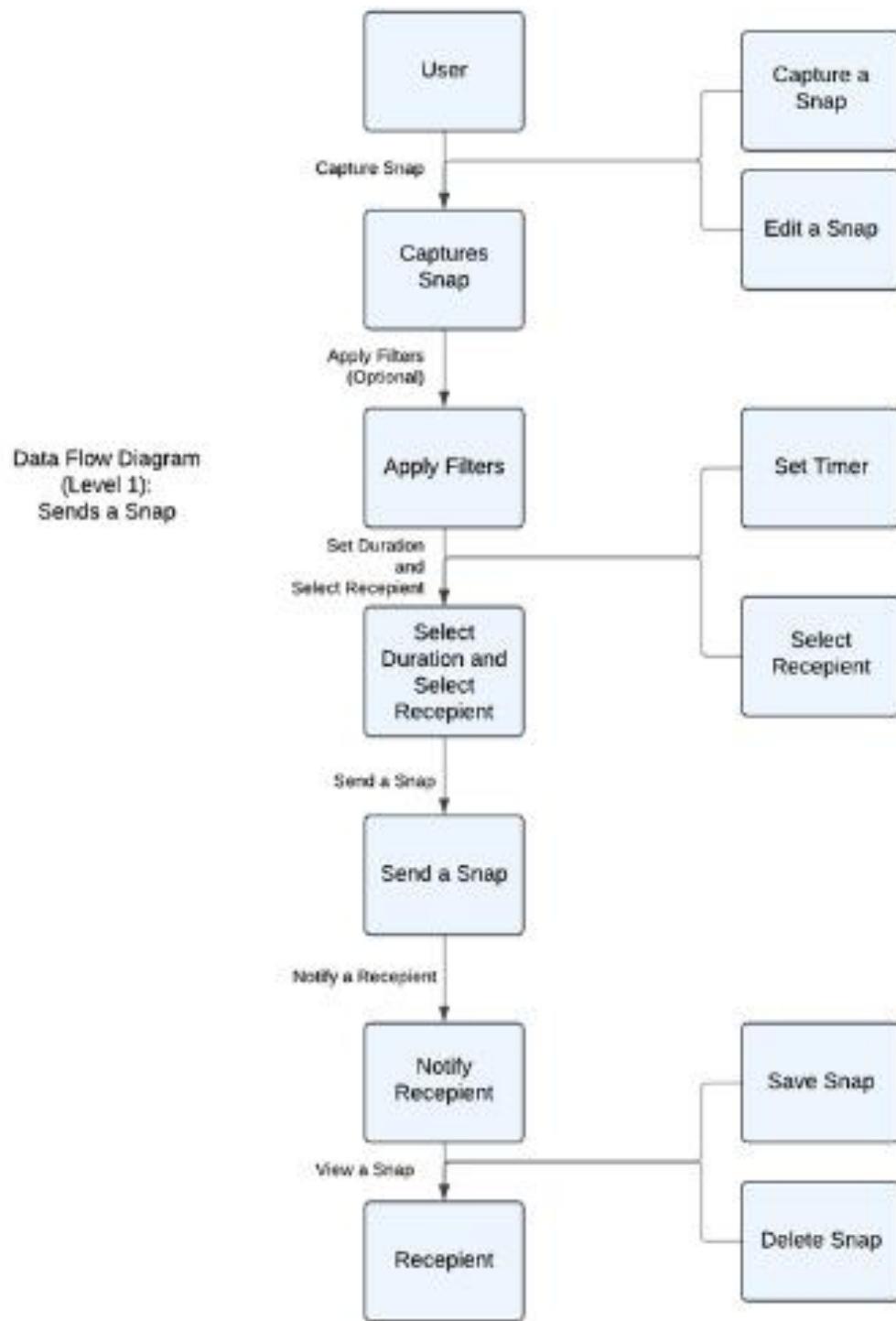


Figure 14: Data Flow Diagram(DFD) Diagram (Level 1) of Primary Functionality of Snapchat

(Website: <https://lucidchart.com>)

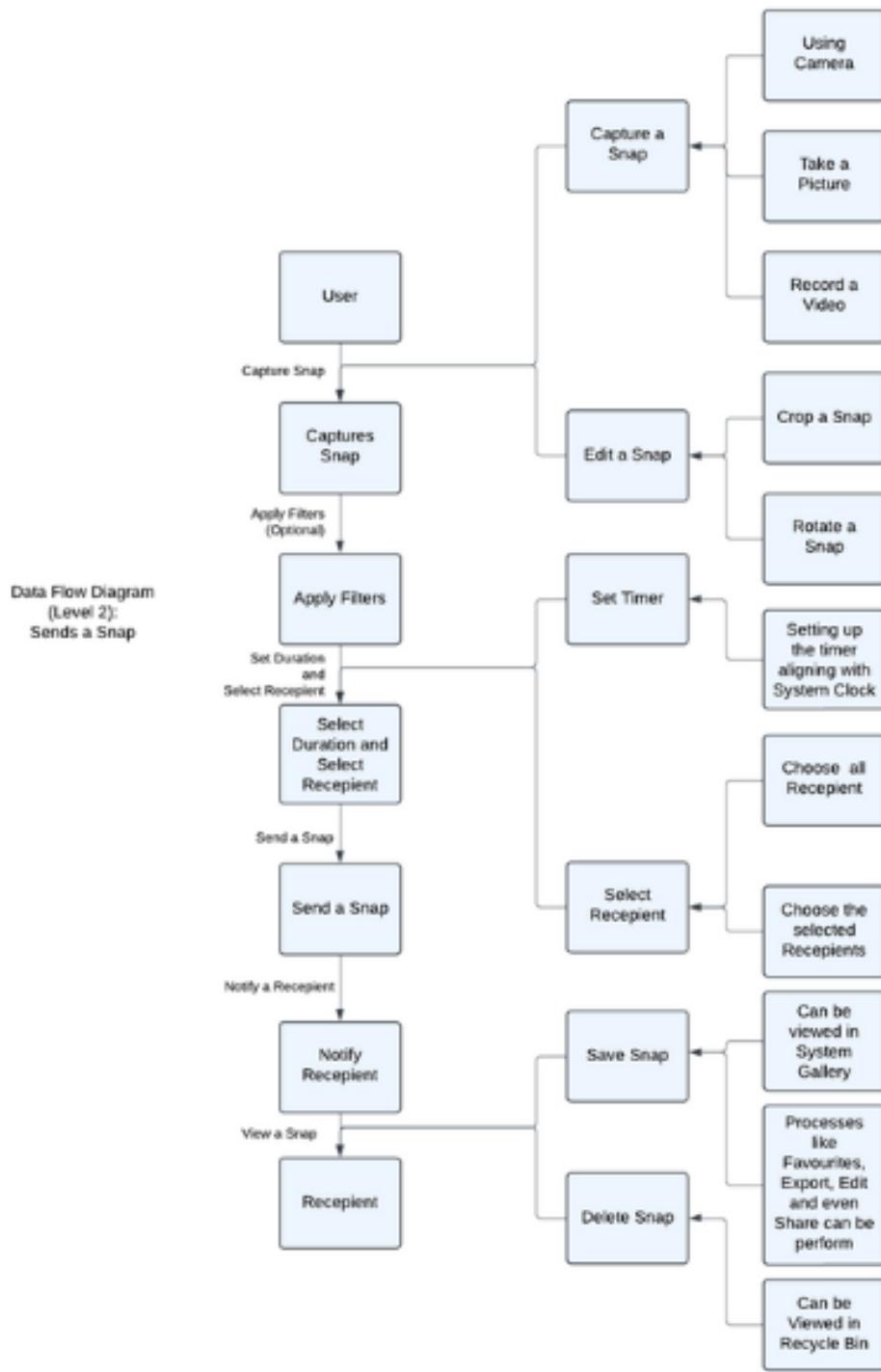


Figure 15: Data Flow Diagram(DFD) Diagram (Level 2) of Primary Functionality of Snapchat

(Website: <https://lucidchart.com>)

23 Proposed Software Architecture

Snapchat's new proposed design is all about making the app faster, smarter, and more personal for each user. By breaking down features into smaller, separate components (called microservices), Snapchat could easily update or scale parts of the app without affecting the whole system. This setup also allows for smarter recommendations, improved security, and a smoother experience during busy times. With this structure, Snapchat would rely on flexible cloud storage and cutting-edge tools to handle the massive number of snaps and stories users create every day.

23a Overview

In this new setup, Snapchat's main features—like Snaps, Stories, Lenses, and Discover—would operate independently from each other. This means Snapchat can make updates to one feature without interrupting the rest of the app, making changes easier and faster. New AI-driven features would bring personalized filters, content recommendations, and interactive lenses based on what each user enjoys. Snapchat's reliance on cloud storage would ensure there's always enough space for the app to run smoothly, even when millions of people are snapping at the same time.

Content

Snapchat's core content—Snaps, Stories, Discover, and AR lenses—would remain the same but would feel more personalized. The new architecture would allow Snapchat to deliver custom recommendations for filters, lenses, and Discover content that feel tailored to each user. By being able to work on each feature separately, Snapchat could improve these experiences faster and with more variety.

Motivation

The motivation for this new setup is to keep Snapchat fresh, fast, and personalized for every

user. By switching to microservices, Snapchat can scale each feature independently, making sure nothing slows down during peak times. The modular design also allows Snapchat to test and release new features quickly. AI personalization means Snapchat can deliver the right content to the right people, creating a more engaging experience and keeping users coming back.

Considerations

- **Scalability and Flexibility:** Each feature can scale on its own, so if Stories gets a lot of traffic, it can expand without impacting other parts of the app.
- **Personalized Experience:** AI in the new setup would allow Snapchat to suggest filters, stickers, and content based on each user's preferences, making the experience feel more relevant.
- **Enhanced Privacy:** Security improvements would protect snaps and personal data, giving users confidence in sharing their real moments.
- **Simplicity in Use:** This new architecture would be implemented carefully to keep the app's easy-to-use interface unchanged, ensuring no disruptions for users.
- **Efficient Resource Management:** Cloud storage and container management would balance high performance with resource costs, keeping the app affordable while running smoothly.

Example

Imagine Snapchat wants to launch a new lens that changes based on the time of day. With the new microservices setup, Snapchat could introduce this feature within the Lenses service without affecting the rest of the app. So, when a user opens Snapchat in the evening, they might see a sunset-themed lens without any delay or disruption in the Stories or chat features. This modular design means Snapchat can keep innovating and updating specific parts of the app quickly and efficiently, while users enjoy a personalized, smooth experience.

23b Class Diagrams

Snapchat's class diagrams would show how different parts of the app connect and interact. Each feature, like Snaps, Stories, Lenses, and Chats, would have its own class with specific attributes and functions. For example, a "Snap" class might have attributes like "duration," "sender," and "recipient," while functions could include "sendSnap" and "deleteAfterView." By organizing each feature with its own set of attributes and actions, Snapchat can keep the code clean and manageable, making it easy to update or add new features in the future.

Content

Each feature in Snapchat—such as Snaps, Stories, Lenses, and Discover—would have its own set of classes, each responsible for its unique data and actions. These classes make it easy to define what each feature can do and how they interact with each other. For instance, the "User" class could hold information like the username, Bitmoji, and friend list, while the "Lens" class could include details about the lens effect and how it applies to snaps in real-time.

Motivation

The goal of these class diagrams is to make Snapchat's system more organized and easy to work with. By clearly defining each feature in its own class, developers can manage changes or fix bugs without affecting the rest of the app. This setup helps Snapchat improve and add new functions faster while ensuring that each feature is separated and easier to maintain.

Considerations

- **Feature Organization:** Each class represents a specific feature, keeping everything organized and separated for better code management.

- **Easy Updates:** With each feature in its own class, Snapchat can make changes to one feature without causing issues for others.
- **Scalability:** New features can be added as new classes without disrupting existing ones, allowing Snapchat to grow smoothly.
- **Consistency:** Class diagrams help developers maintain consistency, ensuring that all parts of Snapchat interact as intended.
- **Clarity:** By mapping out how each class connects, developers can quickly understand and work on different parts of the app.

Example

Consider a “Snap” class that includes attributes like “sender,” “recipient,” “duration,” and “timeStamp.” Functions within this class, such as “sendSnap” and “viewSnap,” define the actions a snap can perform. The class diagram might also show connections to other classes, like “User” (for sender and recipient) and “Lens” (for any filters applied). With this organized structure, adding a new feature, like a time-limited sticker, becomes straightforward—developers would just add it to the “Snap” class. This approach keeps the system organized, efficient, and easy to update.

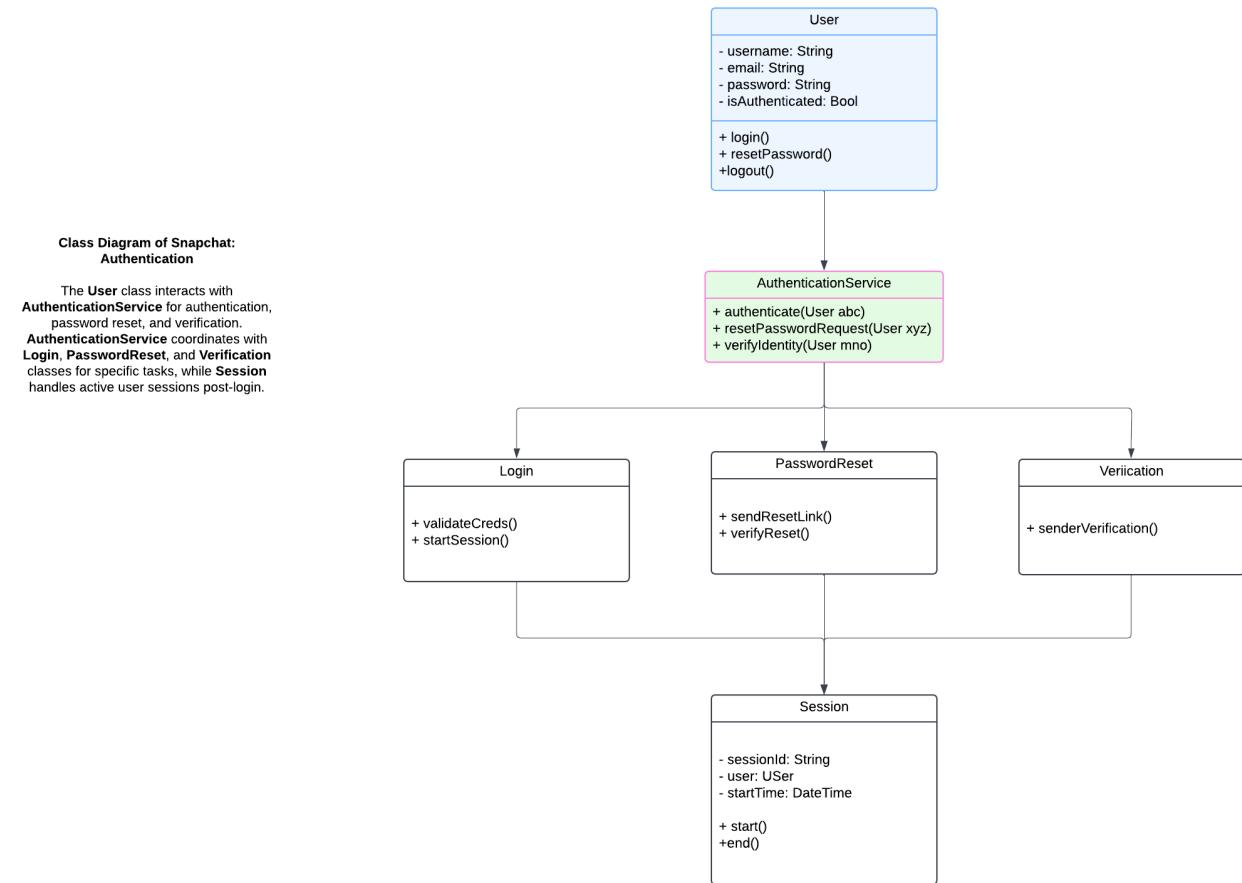


Figure 16: Class Diagram of Module 1: Authentication

(Website: <https://lucidchart.com>)

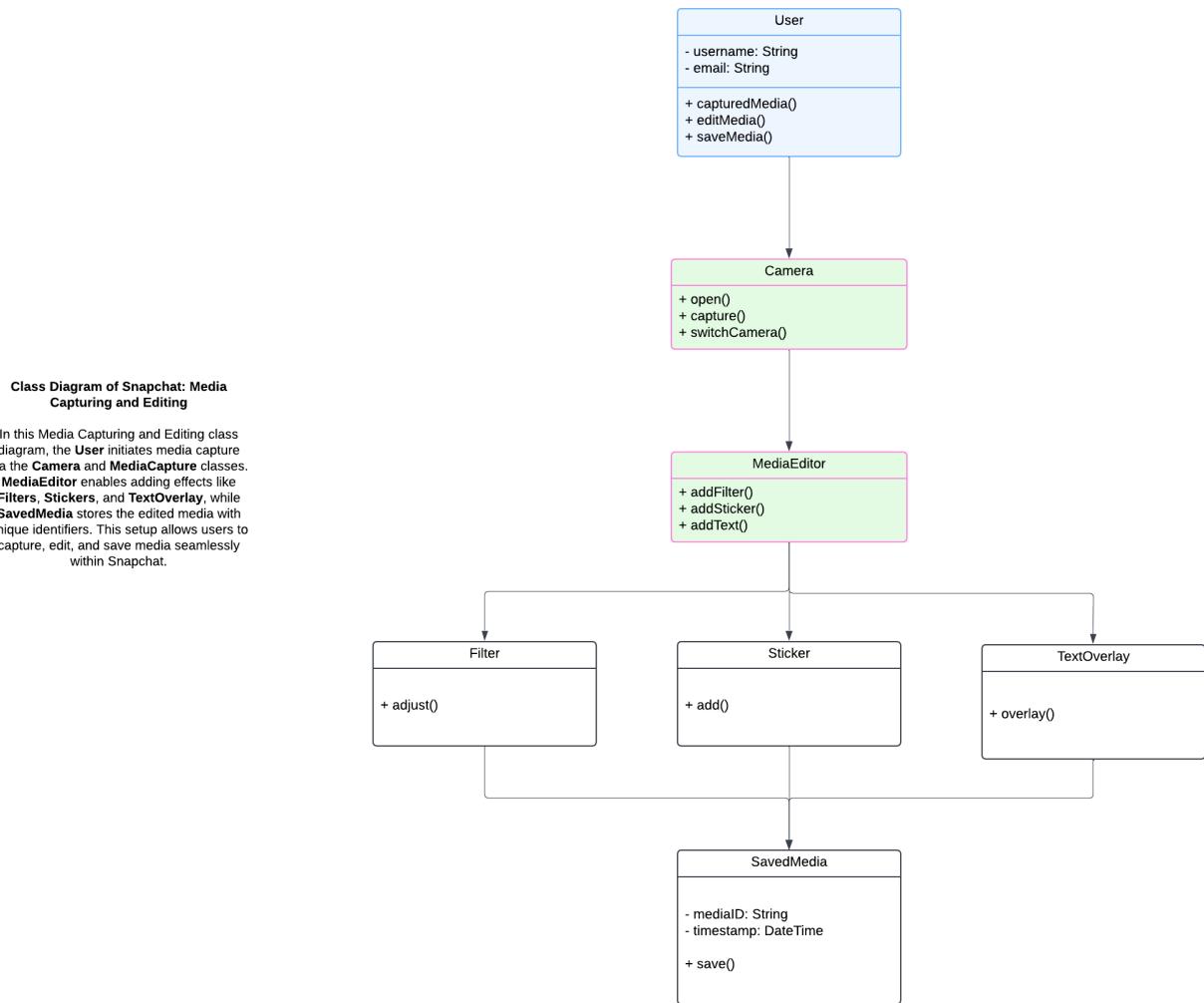


Figure 17: Class Diagram of Module 2: Media Capturing and Editing

(Website: <https://lucidchart.com>)

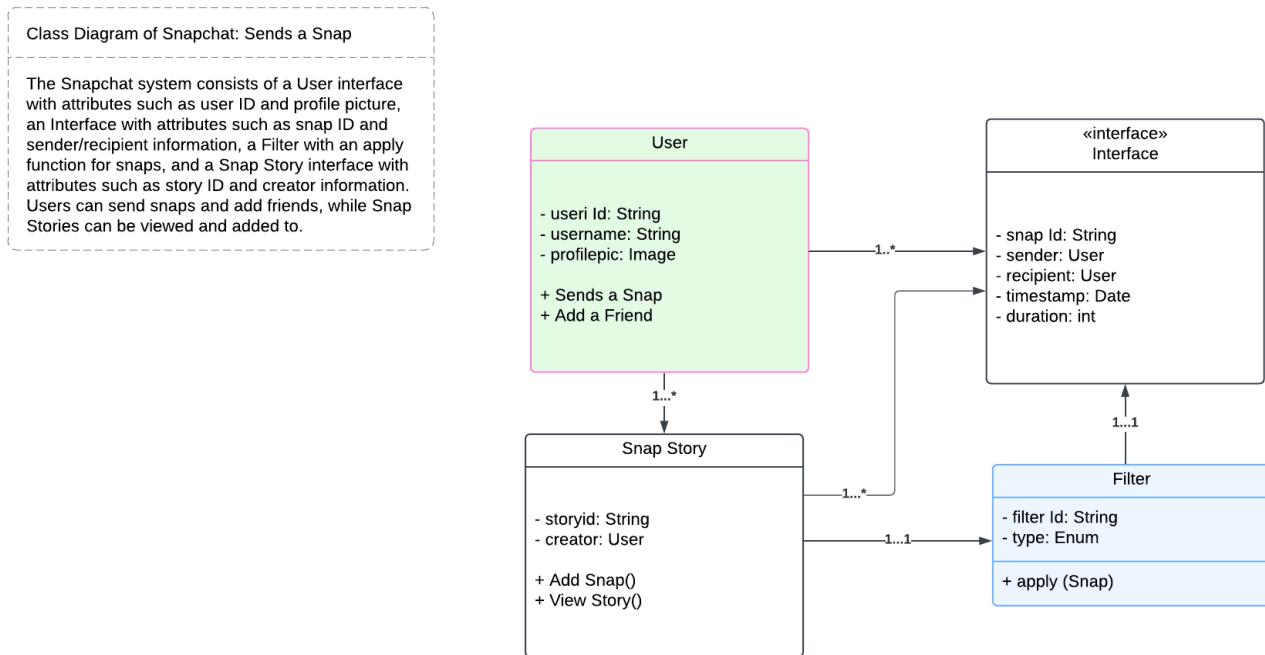


Figure 18: Class Diagram of Module 3: Sends a Snap

(Website: <https://lucidchart.com>)

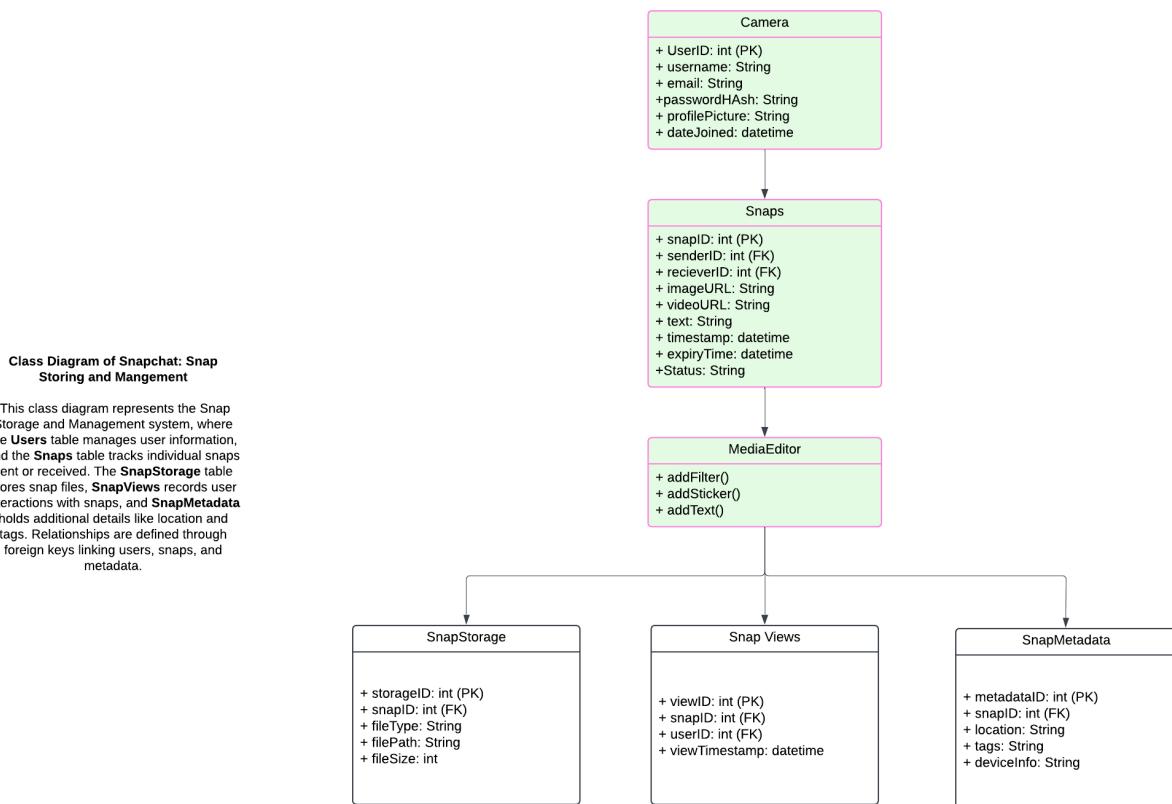


Figure 19: Class Diagram of Module 4: Snap Storing and Management

(Website: <https://lucidchart.com>)

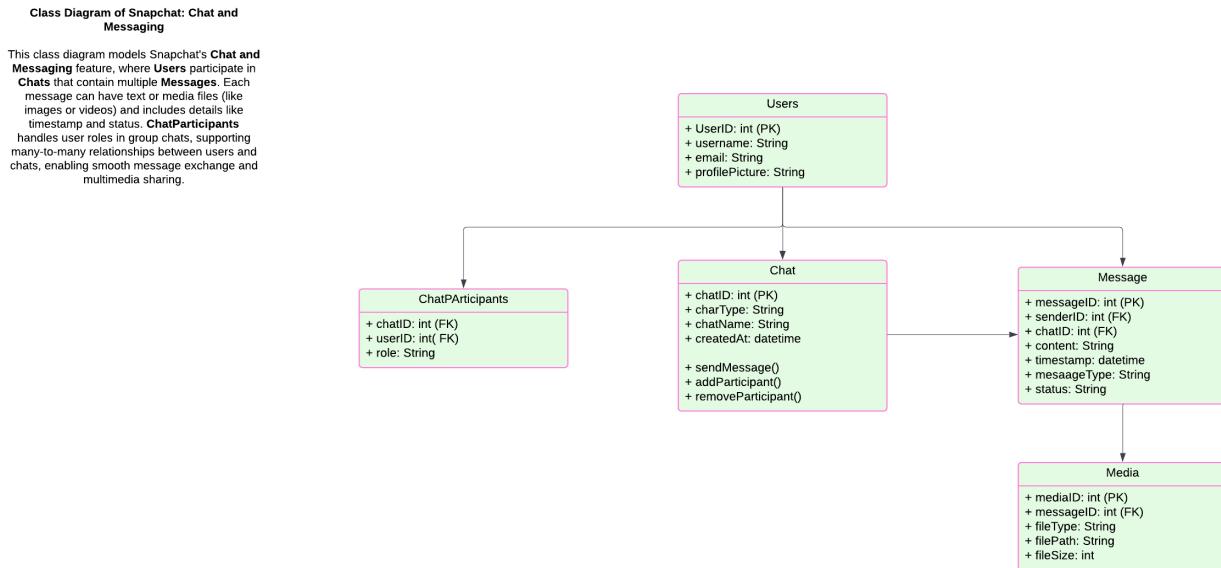


Figure 20: Class Diagram of Module 5: Chat and Messaging

(Website: <https://lucidchart.com>)

23c Dynamic Model

The dynamic model of Snapchat shows how different parts of the app interact over time, especially when a user performs actions like sending a snap, viewing a story, or applying a lens. It maps out the sequence of interactions between components, helping developers understand how data flows and features work together. This model highlights key steps and

processes, showing, for instance, how a snap travels from one user's camera to a friend's inbox and then disappears after viewing.

Content

In Snapchat's dynamic model, each action (like sending a snap or viewing a story) involves multiple steps. These include capturing the snap, applying filters, sending it to the recipient, notifying the recipient, and deleting it after it's viewed. The model would show the entire journey a snap takes, along with the interactions between services and features, making it clear how different parts of Snapchat work together in real-time.

Motivation

The main goal of Snapchat's dynamic model is to capture and illustrate the real-time flow of interactions. This helps developers see how each feature operates and how different components depend on each other. By mapping out these interactions, Snapchat can quickly identify any weak points, improve performance, and keep the user experience smooth and responsive.

Considerations

- **Real-Time Flow:** Snapchat's dynamic model focuses on real-time interactions, ensuring that snaps, messages, and filters load and display instantly.
- **Efficient Data Transfer:** The model highlights the movement of data through Snapchat's services, helping optimize network performance to reduce delays.
- **Smooth User Experience:** Each interaction is mapped to ensure that snaps, stories, and other features run seamlessly, even during high activity periods.

- **Security in Transitions:** Since snaps disappear after viewing, the model includes checkpoints for secure data deletion.
- **Interdependency Awareness:** The dynamic model reveals how different components rely on each other, helping developers spot dependencies that could affect overall performance.

Example

When a user takes a snap and sends it, the dynamic model would outline each step involved: the snap is captured, a filter is applied, it's uploaded to Snapchat's servers, a notification is sent to the recipient, and once the recipient views it, the snap is deleted from the server. This flow shows how the app coordinates between the "Camera," "Snap," "Notification," and "Delete" services, ensuring each component works together smoothly. By visualizing this process, developers can see exactly where delays or issues might occur and make improvements accordingly.

23d Subsystem Decomposition

Snapchat's design divides the app into smaller sections, or "subsystems," each handling a specific part of what the app does. For example, there's a subsystem for messaging, one for the camera and lenses, one for stories, and so on. This setup allows each part to run smoothly and independently, so updates or fixes can be made to one area without disturbing the others. This way, Snapchat can manage each feature more easily, improve specific parts as needed, and keep everything working seamlessly.

Content

- **Messaging:** Takes care of everything related to sending and receiving snaps, chatting, and notifications.
- **Camera and Lenses:** Covers all the camera functions, from taking photos to adding filters and fun AR effects.
- **Stories and Discover:** Manages all user stories, popular content, and personalized recommendations.
- **User Profiles:** Holds details like usernames, Bitmojis, friend lists, and personal settings. Each subsystem focuses on its specific job, working alongside the others to create the full Snapchat experience.

Motivation

Dividing Snapchat into these smaller subsystems makes everything easier to manage and improve. Instead of working with one huge, interconnected system, developers can focus on making changes or adding features to just one part, like Stories or the Camera. This setup also allows each feature to grow and scale individually, so Snapchat can handle more users and more snaps without slowing down.

Considerations

- **Independent Operations:** Each subsystem works mostly on its own, so updates or new features in one part don't mess up the rest of the app.
- **Scalability:** Subsystems can be expanded individually based on user demand, so Snapchat can handle increased activity in any area, like Messaging, as needed.

- **Better Performance:** By focusing on each subsystem, Snapchat can keep everything running quickly and efficiently.
- **Easy Maintenance:** Problems in one part of the app can be fixed without impacting the rest, making it easier to keep things running smoothly.
- **Smart Resource Allocation:** Each subsystem can get the attention it needs, improving the app's overall responsiveness.

Example

Take the **Camera and Lenses** subsystem. This section of the app includes everything from opening the camera to adding filters and lenses. When you take a snap, the Camera and Lenses subsystem handles it all, from capturing the image to letting you add fun AR effects. This subsystem is designed to be separate from others, like Messaging or Stories, so Snapchat can quickly update or add new filters without affecting the rest of the app. This approach allows Snapchat to keep each feature updated, responsive, and easy to manage without causing any interruptions for users.

23e Hardware / software mapping

In Snapchat's hardware and software setup, specific parts of the app's system are mapped to work with particular hardware resources. This layout helps Snapchat make the most out of the devices and servers it runs on, making sure that features like the camera, lenses, and chat load quickly and work reliably. Snapchat's software is divided to handle different tasks based on the device (like your smartphone) or the servers in the cloud that support high-demand functions, making the user experience as smooth and fast as possible.

Content

Snapchat's setup ensures that each feature is optimized for the hardware it's running on:

- **On-device processing:** Tasks like taking photos, applying filters, and displaying snaps happen directly on your phone, reducing the time it takes for images and effects to load.
- **Cloud processing:** Snapchat relies on cloud servers to handle things like Discover content, Stories recommendations, and real-time messaging, as these tasks require more computing power and storage than a single device can manage.
- **Networking:** Snapchat's system balances between Wi-Fi, data, and device connectivity to make sure snaps and stories upload quickly, even when the network connection isn't perfect.

Motivation

The goal of hardware/software mapping is to make Snapchat as responsive and efficient as possible. By carefully deciding what happens on the user's device and what goes to the cloud, Snapchat can deliver a fast experience without putting too much strain on either the phone or the servers. This setup also helps Snapchat balance resource usage, reduce battery drain, and manage high traffic times smoothly.

Considerations

- **Device Limitations:** Snapchat optimizes functions that can run on different devices, making sure they perform well even on older phones.
- **Network Reliability:** By balancing between on-device and cloud processing, Snapchat maintains responsiveness even with slow network connections.

- **Battery and Data Use:** Running intensive tasks on the cloud, like data-heavy Discover content, keeps Snapchat from draining too much battery or data.
- **Scalability:** Cloud resources allow Snapchat to handle millions of users at once without slowing down, while device processing makes it faster for individual users.
- **Security:** Snapchat carefully controls how data flows between devices and the cloud, making sure personal information is protected and only stored temporarily.

Example

When you open Snapchat and start using the camera, most of that processing—capturing, applying lenses, and enhancing images—happens directly on your phone. But when you swipe over to Discover, the app pulls content from cloud servers, where data-intensive recommendations and high-quality images are processed and loaded. This setup ensures your phone doesn't have to handle everything, keeping the app fast and responsive, while Snapchat's servers manage the larger, resource-heavy tasks.

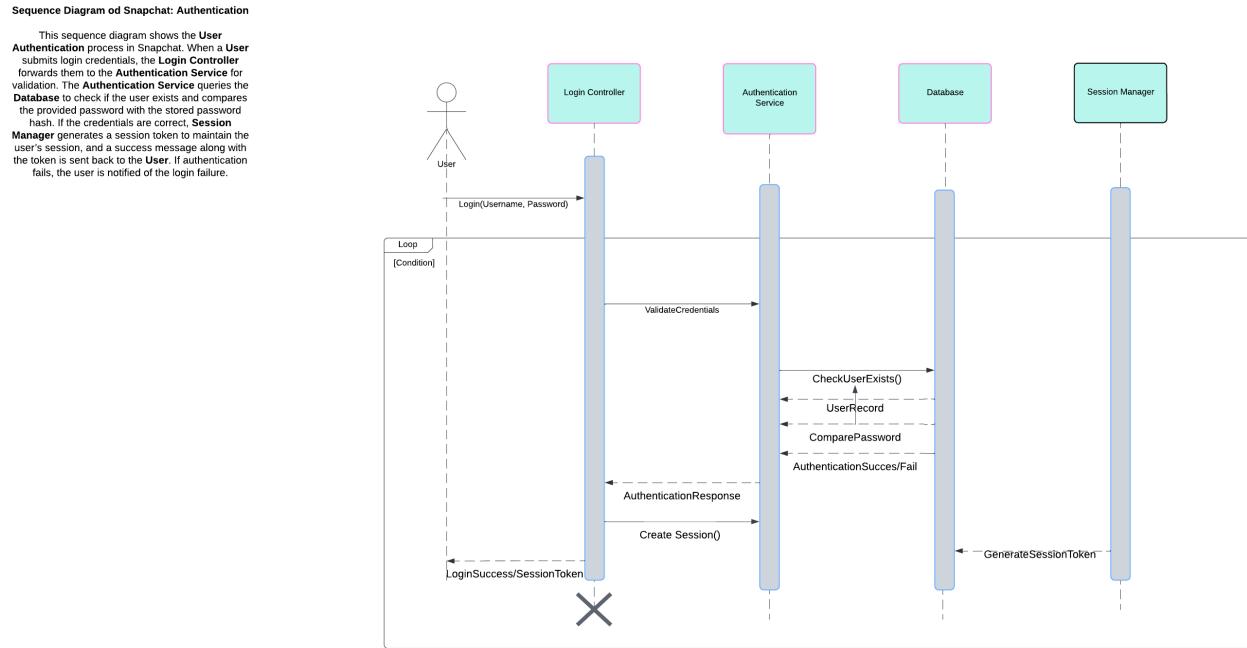


Figure 21: Sequence Diagram of Module 1: Authentication

(Website: <https://lucidchart.com>)

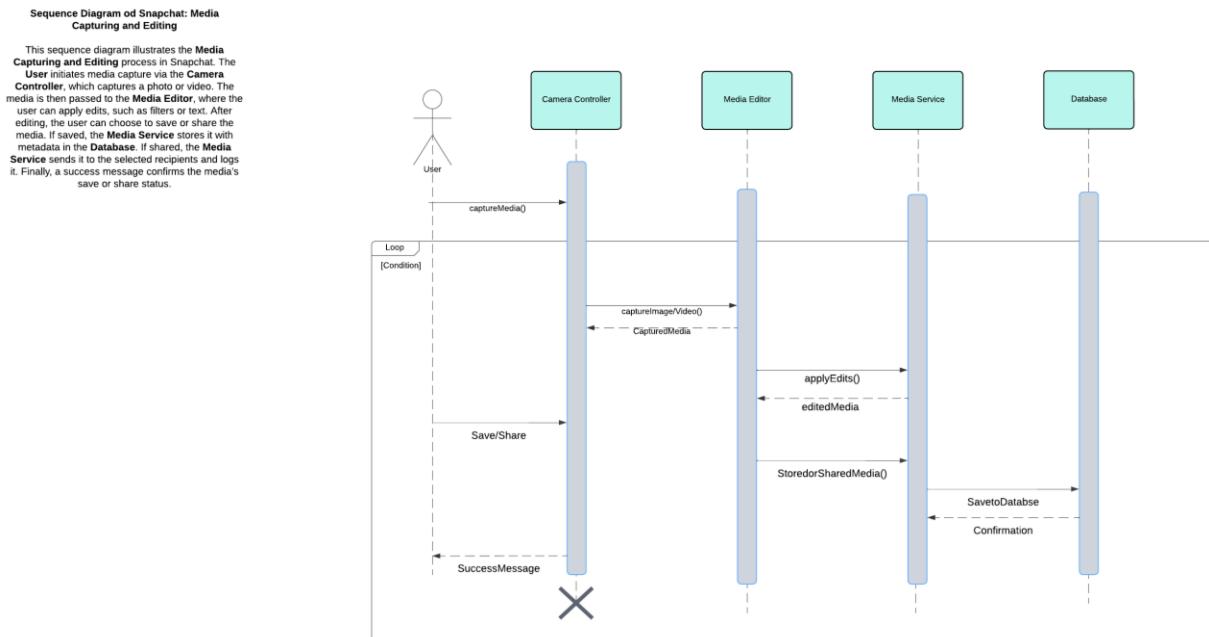


Figure 22: Class Diagram of Module 2: Media Capturing and Editing

(Website: <https://lucidchart.com>)

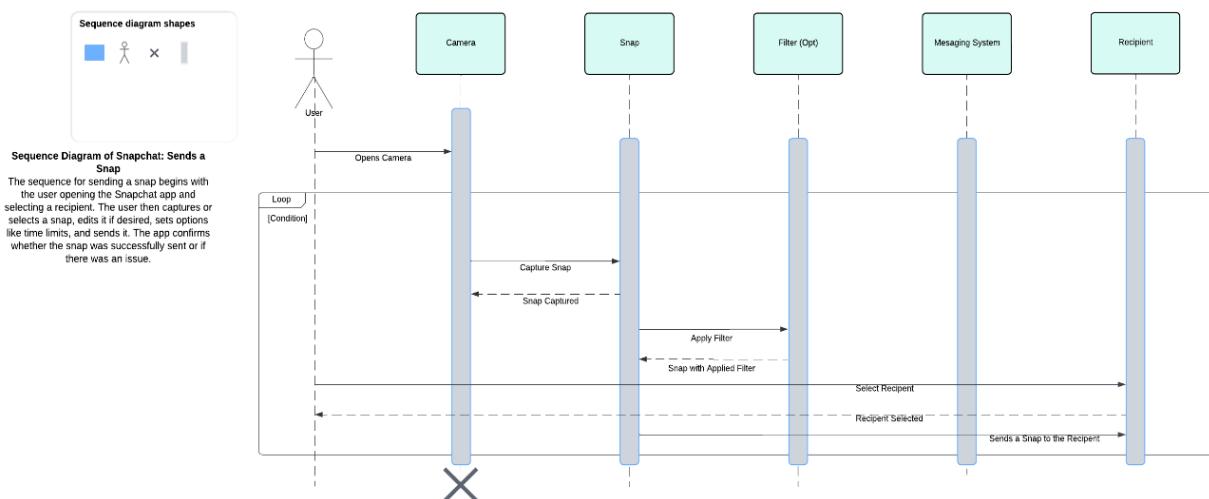


Figure 23: Sequence Diagram of Module 3: Sends a Snap

(Website: <https://lucidchart.com>)

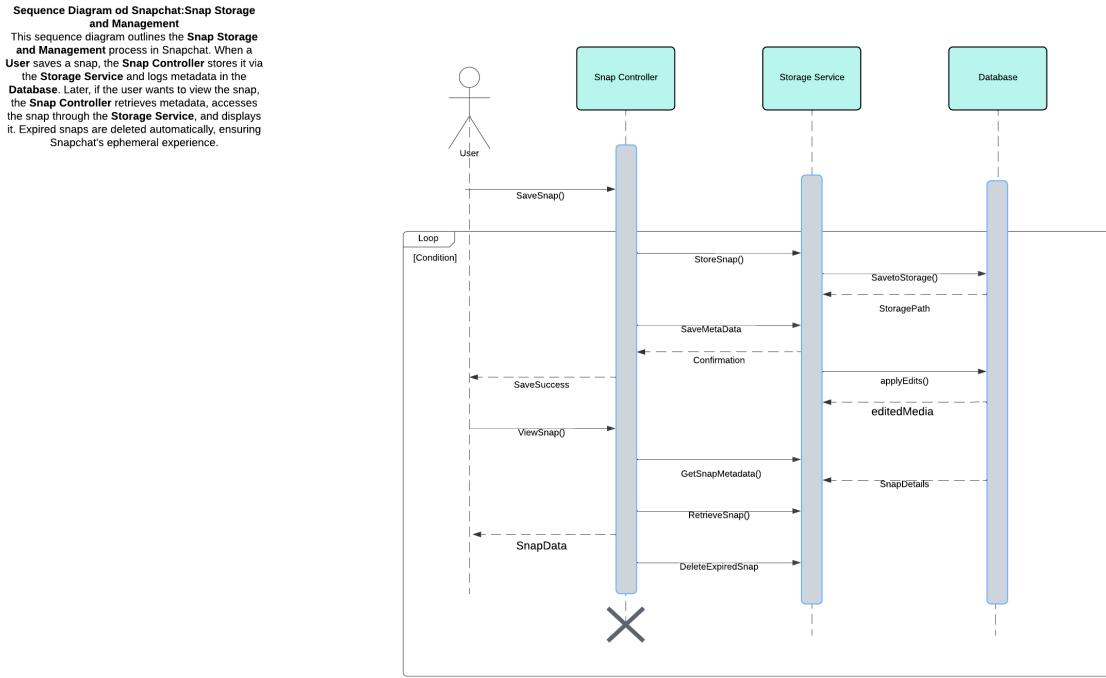


Figure 24: Sequence Diagram of Module 4: Snap Storage and Management

(Website: <https://lucidchart.com>)

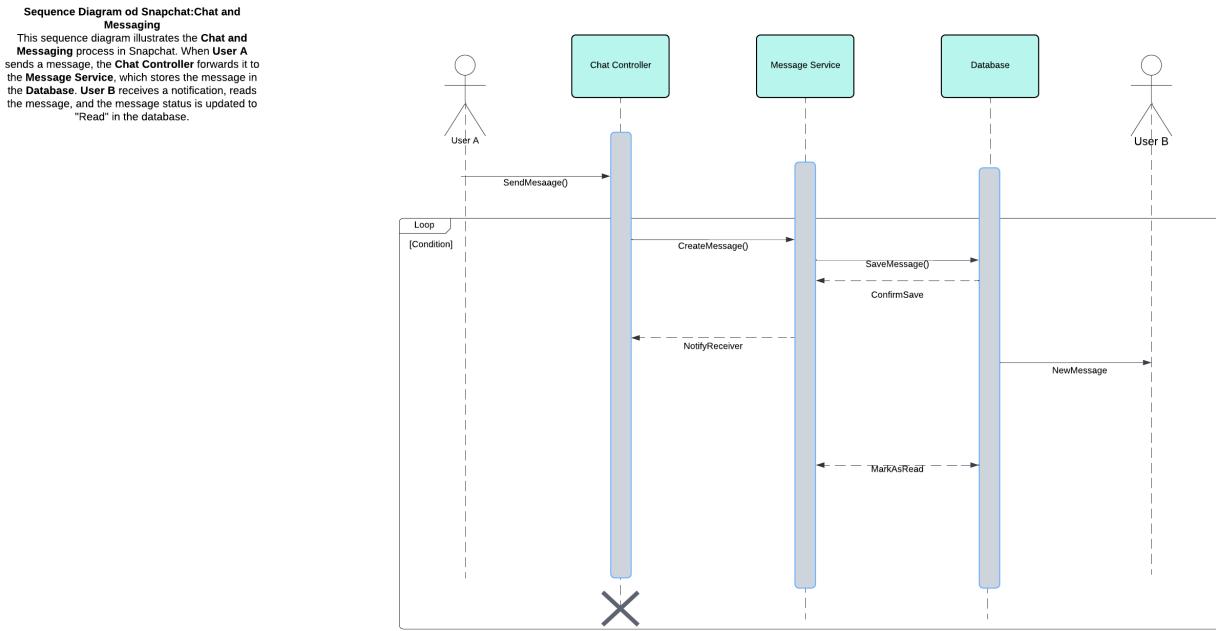


Figure 25: Sequence Diagram of Module 5: Chat and Messaging

(Website: <https://lucidchart.com>)

23f Data Dictionary

It's like a guide that shows exactly what data Snapchat uses, what it represents, and how it all connects. For example, when you send a Snap, the dictionary would define things like who's sending it, who's receiving it, how long the Snap will last, and what filters or lenses you've used. This makes it easier for anyone working on Snapchat to understand the data and ensures everything in the app works correctly without any confusion.

Content

Here's a breakdown of the kinds of data you'd find in Snapchat's data dictionary:

- **User Data:** This includes usernames, your friend list, Bitmoji details, and your personal account settings.
- **Snap Data:** Information like who sent the Snap, who will receive it, the type of media (photo or video), how long it stays visible, and if any filters or lenses are applied.
- **Story Data:** Details about stories you create, including when they were posted and who's seen them.
- **Chat Data:** The content of your messages, when they were sent, and whether the person you messaged has seen them.
- **Discover Data:** Information about Discover content, like video titles, creators, and how many people liked or shared it.

Each bit of data is clearly defined so developers know exactly what to do with it.

Motivation

The reason Snapchat has a data dictionary is to keep everything organized and clear. When

everyone knows exactly what data means and how it's used, it makes Snapchat more reliable and easier to work with. This avoids mistakes and confusion, especially when the app is being updated or new features are added. It's about making sure that data is used correctly and consistently, so Snapchat can keep running smoothly.

Considerations

- **Consistency:** By defining things clearly, like what a “Snap” is, the app always behaves in a consistent way. Everyone knows what’s meant when they talk about “Snap data” or “chat data.”
- **Data Integrity:** The dictionary helps make sure that all the data in the app is accurate, preventing errors in how things like messages or Snaps are displayed.
- **Simplifying Updates:** When Snapchat wants to add a new feature, the data dictionary helps developers understand how new data needs to fit into the system. It’s like following a recipe—everything gets added in the right way.
- **Security:** The data dictionary also points out which information is sensitive (like your private messages), so developers can make sure it’s protected properly.
- **Efficiency:** Knowing exactly what data is needed means Snapchat can avoid storing or transferring unnecessary data, which helps the app run faster and use less battery.

Example

Let’s say the dictionary defines the “Snap” data like this:

- **Sender (User ID):** This tells us who sent the Snap, so we know who created it.

- **Recipient** (User ID): This shows who the Snap is going to.
- **Duration** (integer): This is how long the Snap lasts before disappearing.
- **Filter** (string): If there's a filter or lens, this tells us what kind it is (like a puppy face or sepia tone).
- **Timestamp** (datetime): The exact time the Snap was taken, so we know when it was shared.

With these definitions, everyone working on Snapchat knows exactly what “Snap” means and how to handle all the data involved. This makes development easier, updates simpler, and the app more reliable overall.

23g Persistent Data management

Persistent data management in Snapchat is all about keeping important information safe and accessible, even when you close the app. This includes things like your friends list, chat history, saved Snaps, and story content. Unlike temporary data that disappears after a short period, persistent data is stored in a way that ensures it's available the next time you open the app, so you never lose the important things. The app uses databases and cloud storage to keep track of this data.

Content

In Snapchat, persistent data might include:

- **User Accounts:** Information like your username, password, profile picture, and friend connections.
- **Chat History:** Stored messages and media from conversations you've had with friends.

- **Saved Snaps:** Any Snaps you decide to save to your memories or other parts of the app.
- **Stories:** The content you post to your Snapchat story and who has viewed it.
- **App Settings:** Preferences like notification settings, privacy options, and display choices.

This data is stored securely in a way that ensures it's there every time you log into the app, even if you close it or switch devices.

Motivation

The main goal of persistent data management is to ensure that users' information and content are saved securely and reliably. Since Snapchat involves ongoing communication and media sharing, it's essential to keep track of all this data over time. Whether it's remembering who your friends are, keeping your Snap history, or ensuring your settings remain the same, persistent data management ensures your experience stays consistent no matter when or where you use Snapchat.

Considerations

- **Data Security:** Since persistent data often includes sensitive personal information, like messages or saved Snaps, it's crucial that Snapchat keeps this data safe with encryption and secure access protocols.

- **Storage Efficiency:** Snapchat needs to store a large amount of data efficiently to prevent app performance issues. This means balancing how much data is stored with how quickly it can be accessed.
- **Syncing Across Devices:** Many Snapchat users have the app on multiple devices. Persistent data needs to sync across them seamlessly, so if you switch from your phone to your tablet, you can access the same content without issues.
- **Data Integrity:** The app must ensure that no data is lost or corrupted over time, especially when handling large media files like photos and videos.
- **User Control:** Users should be able to manage their persistent data, such as deleting old conversations or clearing saved Snaps. Giving users control over their data is essential for maintaining privacy.

Example

When you send a Snap to a friend, the data (like the Snap itself and the recipients) is stored temporarily during the sending process. However, once that Snap is saved to your Memories or your chat history, it becomes part of your persistent data. This ensures that if you want to view that Snap again later, or even after reinstalling the app, it will still be available for you. The app keeps this data in cloud storage, so it's accessible across your devices, ensuring you have access to your content anytime you open Snapchat.

23h Access control and security

Think of Snapchat's access control and security as the guard at the entrance to a party, making sure only invited guests get in and that no one sneaks in to steal the spotlight. Snapchat takes your privacy seriously, and its security features ensure that your account and personal content, like your Snaps and chats, are protected from unwanted access.

Content

In simpler terms, here's what Snapchat does to keep you safe:

- **Authentication:** When you log into Snapchat, the app checks if you're you. You enter your username and password, but if you've set up extra security, like two-factor authentication, Snapchat will send you a code to make sure it's really you trying to log in.
- **Encryption:** Imagine you send a private Snap. It gets encrypted, meaning it's turned into a secret code that can't be read by anyone else. Even if someone intercepts it, they won't be able to see what you've sent unless they have the secret code.
- **Authorization:** Snapchat controls what you can access. For instance, you can see your own private messages, but not your friend's private stuff. It's about making sure the right people see the right things.
- **Session Management:** If you leave your Snapchat open on a device, it doesn't stay open forever. After a while, Snapchat will log you out automatically to stop anyone from using it if you forget to close it.
- **Data Privacy:** Snapchat is careful about your personal data. It follows strict rules about how your data is collected, stored, and shared. For example, Snapchat doesn't sell your data to third parties, and it makes sure it's protected by encryption.

Motivation

The main reason Snapchat focuses so much on access control and security is simple: it wants to keep your private moments private. You share photos, videos, and messages that you don't want just anyone to see. Snapchat needs to keep these safe from hackers, unwanted eyes, or anything that could compromise your privacy. Plus, by keeping things secure, Snapchat

builds trust with its users, ensuring you feel comfortable using the app without worrying about your personal data being exposed.

Considerations

- **Privacy Comes First:** The goal is to keep your personal data safe. Snapchat does its best to ensure that only you and the people you choose can see your Snaps, chats, or stories.
- **Balancing Security and Ease:** Too many security steps can be a pain. Snapchat tries to find the right balance—ensuring your data is protected without making you jump through too many hoops.
- **Encryption:** Encryption is Snapchat's way of saying, "Even if someone tries to intercept your data, they won't be able to read it." Your messages and Snaps are encrypted, so they're kept private.
- **Responding to Threats:** Snapchat is constantly on the lookout for strange activity. If it notices something suspicious, like a bunch of failed login attempts, it will take action to protect your account.
- **Giving Users Control:** You're in charge of your security settings. You can turn on two-factor authentication, decide who sees your content, and manage app permissions to ensure everything is locked down just the way you want it.

Example

When you log in, Snapchat makes sure it's really you trying to get into your account. If you've set up two-factor authentication, you'll get a code sent to your phone to confirm it's you. When you send a Snap or message, Snapchat encrypts it so no one else can see it unless they have the decryption key. And if someone tries to hack into your account, Snapchat detects the suspicious activity and locks you out until it's sure everything's safe. These steps help you enjoy the app with peace of mind, knowing that your personal information stays secure.

23iGlobal software control

For Snapchat, global software control means ensuring that the app works seamlessly for users all over the world while managing things like updates, features, and performance across different regions. It ensures that the app is consistent, reliable, and works well no matter where you are.

Content

Here's how global software control works in Snapchat:

- **Updates:** Snapchat regularly rolls out updates to improve the app, fix bugs, or introduce new features. These updates are managed globally, meaning no matter where you are in the world, you get the latest version of the app, ensuring consistency.
- **Feature Management:** Different countries may have different features available depending on local regulations, cultural preferences, or the technical environment. Snapchat manages and controls which features are accessible in which region.
- **Global Performance Monitoring:** Snapchat keeps an eye on how well the app performs in different regions. This includes making sure the app runs smoothly in terms of speed, data usage, and server reliability, regardless of the user's location.
- **Internationalization:** Snapchat adjusts its language, content, and design to fit various regions. This includes supporting multiple languages and tailoring content to different cultures and local laws.

Motivation

The goal of global software control is to ensure that Snapchat provides a consistent and smooth experience for all users, no matter where they are. As Snapchat is used worldwide, it's essential that the app is tailored to meet the needs of users in different countries. Whether

it's ensuring the app works fast in countries with varying internet speeds or making sure that content complies with local laws, global software control is all about maintaining the quality and availability of the app on a global scale.

Considerations

- **Cultural Sensitivity:** Snapchat needs to be mindful of cultural differences and local laws when rolling out features or content. What works in one country may not work in another, so regional adaptations are necessary.
- **Consistency:** It's important that the core experience of Snapchat is the same no matter where you are. This means updates, features, and content should be delivered consistently around the world.
- **Regional Performance:** Snapchat needs to account for varying internet speeds and mobile networks around the world. Ensuring that the app performs well, even in areas with slower connections, is crucial.
- **Data Privacy and Compliance:** Different countries have different laws around data privacy, like GDPR in Europe. Snapchat must comply with these local regulations to ensure the app respects users' privacy everywhere.
- **Multilingual Support:** Since Snapchat is used globally, the app must support multiple languages and adapt to the needs of users from different linguistic backgrounds.

Example

Let's say Snapchat releases a new feature, like a special filter. The global software control team makes sure that it's available in every country, but some regions might get slightly

different versions due to local regulations. For example, a country might have a law that prevents certain types of content, so Snapchat would adjust the feature to ensure it complies. Meanwhile, users in countries with slower internet speeds might have access to a version of the app that uses less data. This global management ensures that no matter where you use Snapchat, you get a great, localized experience.

23jBoundary conditions

Boundary conditions are like the rules or limits that define how Snapchat behaves in different situations or environments. Just like a sports game has rules about what players can and can't do, boundary conditions set the limits for how the app operates—whether it's handling user input, managing network issues, or controlling how features work under certain circumstances.

Content

Here's how boundary conditions work in Snapchat:

- **Network Conditions:** Snapchat needs to handle situations where the internet connection is weak or unreliable. It might delay sending Snaps, lower image quality to save bandwidth, or store content temporarily until the connection improves.
- **User Device Limitations:** The app must account for different devices with varying hardware capabilities. Older smartphones may struggle to run the latest version of Snapchat, so the app may limit some advanced features on such devices to ensure a smooth experience.
- **Data Usage:** Users in some regions might have limited data plans. Snapchat provides options to limit data usage, like a ‘low data mode,’ to make sure users don’t exceed their data limits.

- **Content Restrictions:** Snapchat needs to respect local laws and cultural boundaries. For example, certain content types may be blocked or restricted based on country-specific regulations or community guidelines.
- **App Version Compatibility:** Not everyone uses the most up-to-date version of Snapchat. The app has to be designed to work with multiple versions, ensuring compatibility between older and newer updates.

Motivation

Boundary conditions ensure that Snapchat works well for everyone, no matter the circumstances. Whether you're using an old phone with limited processing power, a slow internet connection, or live in a country with strict content rules, boundary conditions allow Snapchat to adapt to those situations. It's all about giving users the best possible experience within the constraints of their environment, while also complying with regulations.

Considerations

- **Adaptability to Device and Network Constraints:** Snapchat needs to function smoothly on a variety of devices, from the latest smartphones to older models, and under different network conditions, from high-speed Wi-Fi to slow mobile data.
- **Regulatory Compliance:** Snapchat must respect country-specific regulations that might restrict or limit content in certain regions, ensuring the app stays compliant while still offering a valuable service.

- **User Control Over Features:** Users should be able to manage how Snapchat works on their devices, like choosing whether to limit data usage or disable certain features to enhance performance.
- **Backward Compatibility:** As Snapchat updates its app, it must continue to support older versions of the app to ensure that users who haven't updated their software can still use it.
- **Performance Under Different Conditions:** Snapchat needs to perform well even under boundary conditions like low internet speed or device performance limitations, which could affect features like video streaming or image quality.

Example

Let's say you're on a trip to a remote area where your internet connection is slow. Snapchat will automatically adjust, reducing the quality of Snaps or temporarily saving your content until you're connected to a better network. Or if you're using an older phone, certain features—like high-definition video or advanced filters—might be limited to make sure the app still runs smoothly. Meanwhile, if you're in a country with content restrictions, Snapchat might block certain types of content based on local laws. These boundary conditions ensure that Snapchat still functions well, no matter where you are or what device you're using.

24 Subsystem services

Subsystem services are like the backstage workers in a play—they're the different services within Snapchat that support the main functions of the app. These are the smaller, specialized tasks that work behind the scenes to make sure everything runs smoothly, from sending a Snap to managing your profile or ensuring you stay connected.

Content

Here's what subsystem services look like in Snapchat:

- **Authentication Service:** This subsystem ensures that only authorized users can log into Snapchat by verifying usernames, passwords, and handling two-factor authentication.
- **Messaging Service:** This service takes care of all the chatting and Snapping you do on the app. It's responsible for sending and receiving Snaps, text messages, and video calls.
- **Media Service:** This subsystem handles everything related to media—whether it's taking photos, applying filters, recording videos, or storing them safely in your Snapchat memories.
- **Notification Service:** This ensures you never miss anything important by sending you push notifications for messages, updates, or activities in your Snapchat feed.
- **Location Service:** This subsystem allows Snapchat to track your location and offer location-based features, like geofilters or Snap Map, showing where your friends are.

Motivation

The motivation behind subsystem services is to break down Snapchat's functionality into manageable, specialized tasks that can work independently but still communicate with each other. By organizing the app into smaller services, Snapchat can handle different tasks efficiently and ensure everything from sending Snaps to managing your profile works flawlessly. These services make Snapchat feel like one smooth experience, even though it's supported by multiple systems.

Considerations

- **Scalability:** As Snapchat grows in terms of users and data, the subsystem services need to be able to handle more requests without slowing down or breaking. For example, the messaging service needs to be able to send millions of Snaps per second without crashing.
- **Reliability:** Subsystems must be reliable and functional at all times. If the authentication service goes down, no one can log in, so Snapchat has to ensure that each service is always available.
- **Efficiency:** The subsystem services need to do their tasks efficiently, using minimal resources while still providing a fast and smooth experience. For instance, media services need to compress large video files to save space and bandwidth.
- **Security:** Since services handle sensitive data like login credentials, Snaps, and location, they must be secure. Encryption, access control, and monitoring are critical for preventing unauthorized access and data breaches.
- **Interoperability:** All the subsystems need to work well together, meaning the authentication service has to talk to the messaging service to ensure that only authenticated users can send Snaps. This seamless interaction between subsystems is crucial for the app's functionality.

Example

Imagine you're sending a Snap to a friend. The **Messaging Service** takes the Snap and sends it through the app. It communicates with the **Media Service** to apply any filters or edits you've chosen, and the **Location Service** checks if you're adding a geofilter or sharing your location. Once the Snap is ready, the **Notification Service** might send a ping to your friend's

device to let them know they have a new Snap. Meanwhile, the **Authentication Service** ensures that both you and your friend are logged in and authorized to use the app. All these subsystems work together seamlessly to get your Snap delivered.

25 User Interface

The user interface (UI) is the part of Snapchat you interact with—the design, buttons, icons, and layout that make the app easy and enjoyable to use. It's like the front door to the app; it welcomes you in and guides you through everything. Snapchat's UI is designed to be intuitive and visually appealing, so you don't have to think too much about how to use it.

Content

Here's what makes up Snapchat's user interface:

- **Home Screen:** The main screen where you can quickly see your friends' stories, take a Snap, or open chats. It's designed to be simple and minimalist, with easy access to all the essential features.
- **Navigation Bar:** A bar at the bottom or side of the screen that lets you switch between different sections of the app, like the camera, chat, or your profile. It's there to make moving around the app faster and more efficient.
- **Camera Interface:** Snapchat's main feature is the camera, so the camera interface is a big part of the UI. The buttons to take photos or videos, add filters, or change camera settings are all placed within easy reach, ensuring a smooth experience when snapping photos.
- **Chat Window:** A clean and simple design where you can see your messages, photos, and videos in conversations with friends. The UI here focuses on clear communication without distraction.

- **Stories Feed:** The UI of stories lets you scroll through what your friends are sharing and click on individual stories to view them. It's designed to be visually rich and easy to navigate.
- **Profile Page:** Where you can view and edit your account details, see your Snap score, add friends, and adjust settings. This page is user-friendly and designed to give you quick access to anything related to your account.

Motivation

The motivation behind Snapchat's user interface is to make the app fun, easy to navigate, and visually engaging. Snapchat aims to keep things simple while offering plenty of features. The UI must support the primary function of Snapchat—sharing and viewing content—while making the overall experience as enjoyable and intuitive as possible. The interface should reduce any friction so that users can focus on having fun and sharing moments with friends.

Considerations

- **Simplicity:** Snapchat's interface is designed to be easy to understand, even for new users. The buttons are large and clear, and features are logically organized so users can quickly find what they're looking for.
- **Accessibility:** The UI needs to be accessible to all users, including those with disabilities. This means having text labels, color contrasts, and support for screen readers for a more inclusive experience.

- **Consistency:** The UI elements need to look the same across all parts of the app so users aren't confused when they switch from one section to another. Consistency helps create a more seamless experience.
- **Responsiveness:** The UI should respond quickly to touch and gesture inputs. Whether you're swiping through stories or typing a message, the interface should feel fast and responsive.
- **Aesthetics:** Snapchat's UI is visually designed to be fun and engaging. The use of bold colors, playful animations, and custom icons helps give the app a personality that users enjoy interacting with.

Example

When you open Snapchat, you're immediately taken to the home screen, where you can see your friends' stories at the top and have easy access to the camera and chat options. This simple, organized layout means you don't have to search around for features. If you want to send a Snap, the camera button is always front and center, making it quick to capture a photo. If you want to see your messages, just tap the chat icon in the bottom navigation bar. The UI is designed so that each screen is self-explanatory, letting you spend more time sharing moments and less time figuring out how to use the app.

26 Object Design

26a Object Design tradeoffs

Object design tradeoffs refer to the decisions that Snapchat's development team must make when creating the underlying structure of the app's objects (the components, classes, and data that the app works with). These decisions impact performance, maintainability, flexibility, and user experience. In simple terms, it's about balancing different factors to get the best results for both the users and the development process.

Content

Here are some of the key object design tradeoffs in Snapchat:

- **Data Storage vs. Speed:** Snapchat needs to store a lot of data, from photos and videos to chat messages and user profiles. The tradeoff is deciding whether to optimize for faster access (like loading data quickly) or for storing more data in a smaller space, which might make the app slower.
- **Complexity vs. Maintainability:** As Snapchat's features grow, developers face tradeoffs between making the codebase more complex to support new features or keeping it simpler for easier maintenance. More complex systems might offer more functionality, but they can also be harder to maintain and update.
- **Customization vs. Consistency:** Snapchat allows users to personalize their experience with filters, lenses, and themes. The tradeoff here is balancing the desire for customization (giving users more control) with maintaining a consistent and smooth experience for all users.
- **Modularity vs. Performance:** Snapchat's object design must decide between modularizing features, which allows for easier updates and testing, or keeping the system more integrated to improve overall performance and reduce the load time of different actions in the app.
- **Security vs. Usability:** Ensuring that Snapchat is secure (e.g., protecting user data, preventing unauthorized access) can sometimes mean more complex design and slower performance. Developers have to find ways to ensure security without sacrificing a smooth, fast user experience.

Motivation

The motivation behind considering object design tradeoffs is to make sure Snapchat works as efficiently and effectively as possible for users while also being maintainable and flexible for developers. Each design decision can have consequences, so it's about finding the right balance. For example, Snapchat might prioritize performance to ensure that Snaps load quickly but have to carefully manage how much data is stored on the device or in the cloud to avoid slowing the app down.

Considerations

- **Performance:** How fast the app responds to user actions, like sending a Snap or opening a chat, is essential. The tradeoff may involve choosing a design that sacrifices storage space for faster load times.
- **Scalability:** Snapchat must design objects that can handle a growing number of users and data over time. A decision that works well for a small number of users may not scale well as the app grows.
- **Security:** Snapchat handles a lot of sensitive data, so object design must ensure secure storage and transmission. However, overly complex security systems might slow down performance or complicate the development process.
- **Flexibility:** Snapchat's object design needs to allow for easy feature updates and changes. Developers may need to decide whether to build features that are highly flexible or simpler and faster to implement but harder to modify later.
- **User Experience:** The ultimate goal is to create a seamless user experience. Every design choice—from how data is structured to how features are modularized—affects how easily users can interact with Snapchat.

Example

Let's say Snapchat is deciding how to handle user-generated content like photos and videos. The team could choose to store the content directly on the user's device (which makes the app load faster but might take up a lot of storage) or on remote servers (which saves space on the device but can slow down loading times depending on the network). They might also need to decide how to manage user profiles—whether to use a more complex, detailed model that allows for more customization or a simpler version that's quicker to process. Each of these decisions comes with tradeoffs in terms of speed, complexity, security, and user experience.

26b Interface Documentation guidelines

When Snapchat's developers want to build or update a feature, they need clear guidelines on how everything should connect and work together. This helps keep things smooth and prevents any misunderstandings that could lead to bugs or a poor experience for users.

Content

Here's what you'll find in Snapchat's interface documentation:

- **What Each Function Does:** Every method, like “sendSnap()” or “applyFilter()”, has a clear explanation of what it does. It's like a step-by-step guide for developers to understand what happens when someone presses a button to send a Snap or apply a filter.
- **How Data Moves Around:** When you send a Snap, how does it get from your phone to your friend's? The documentation breaks this down, showing how Snapchat's systems send, store, and display content, so developers know exactly how things are supposed to flow.

- **The Rules of Communication:** There are clear instructions on how different parts of Snapchat should “talk” to each other. For example, how should the login feature talk to the user database, or how should the camera work with the messaging system to send images?
- **Handling Problems:** Sometimes things don’t work perfectly. If a Snap doesn’t send, or if the app is freezing, the documentation explains what to do. Developers follow these guidelines to fix issues fast, so the app keeps running smoothly.
- **Version History:** Whenever Snapchat updates or changes something in its system, the documentation keeps track. This way, anyone working on the app knows what changed and when.
- **How Fast It Should Work:** The documentation also lays out how fast things should happen. For example, how long should it take for a Snap to load, or how should the app handle a surge of users when a major event is happening?

Motivation

The reason Snapchat keeps detailed interface documentation is to avoid chaos. Without it, each developer might build their own way of doing things, and that could break features or confuse users. With clear documentation, everyone follows the same rules, so everything works together seamlessly. It also helps new team members get up to speed quickly because they can just follow the instructions instead of trying to figure out how things are connected on their own.

Considerations

- **Clear and Simple:** The documentation has to be written clearly so that any developer, regardless of experience, can easily understand it and follow it. The simpler, the better!
- **Stay Consistent:** The way the documentation is laid out should be the same every time. That means using the same language, format, and structure throughout, so no one gets lost or confused.
- **Keep It Updated:** Snapchat is always evolving, and so are its features. Documentation needs to be easy to update, so when a change is made—like a new filter or a new way to send Snaps—it's immediately reflected in the guide.
- **Accessible to Everyone:** Whether it's the design team, the developers, or even someone new to the project, the documentation should be easy for anyone to access and use.
- **User-Centered:** The goal of the interface documentation isn't just to make the app work—it's to make sure the user experience is top-notch. Any changes or additions should improve the user experience, not get in the way.

Example

Let's say Snapchat's team is updating how the **Camera Interface** works. The documentation would explain exactly what happens when you take a photo, from the moment you press the shutter button to when the image is saved to your phone or sent to a friend. If something goes wrong, like if the camera can't access the storage, the guide would explain what the app should do next, like showing a message that says, "Couldn't save your photo, please try again." This keeps everything running smoothly and ensures that developers know exactly what's expected, even if something doesn't go as planned.

26c Packages

Imagine a package like a well-organized box where everything related to one specific task in Snapchat gets neatly packed together. It helps keep the app's code organized so developers can focus on one part of the app at a time without getting lost in all the other details. If Snapchat is a big house, packages are like the rooms, each with its own purpose—whether it's for taking photos, sending messages, or applying filters.

Content

- **Related Code Together:** Packages keep everything connected to a single task in one place. For example, there might be one package for all the code needed to send Snaps, and another one for everything related to the camera and taking pictures.
- **Independent but Connected:** Each package can work on its own, but some might need to talk to others. The camera package might need to connect with the file storage package to save a photo, for example. Developers know where to find everything they need because of this neat structure.
- **Reusable Parts:** If a piece of code is used in different places (like the image compression tool), it goes into a package so it can be easily accessed and reused wherever needed, saving time and effort.

Motivation

The big reason for using packages is to make Snapchat's code easier to work with. If everything was thrown together in one big pile, it would be confusing, especially when it's time to update or fix something. By putting related pieces of code into different packages, developers can easily focus on one area at a time, making it much quicker and simpler to work on the app. When Snapchat wants to add new features or fix bugs, it's all possible without disrupting other parts of the app.

Considerations

- **Stay Focused:** Each package should do one thing really well. If a package is trying to do too many things, it can get messy and harder to manage. For example, the messaging code should live in its own package, separate from the camera-related code.
- **Keep it Loose:** Packages shouldn't be tightly linked together. If one package changes, it shouldn't break the rest. This makes the app flexible and easy to expand without causing problems in other areas.
- **Room to Grow:** As Snapchat adds more features, the packages should be easy to expand. The way packages are organized today should leave room for new additions in the future without making everything feel cramped.
- **Easy to Test:** With everything grouped into neat packages, it's much easier to test each part individually. Developers can check one function or feature at a time, making it faster to catch bugs before they cause trouble.
- **Clear Instructions:** Every package needs a set of simple, clear instructions that explain what it does, how it connects to other parts of the app, and how developers should use it. This helps new team members quickly understand how to work with the code.

Example

For example, Snapchat could have a **Camera Package** that takes care of everything related to the camera. Inside this package, there could be different parts: one piece of code for actually taking the picture, another for applying filters, and a third for saving the photo. If Snapchat wants to add a new feature, like a cool new filter, developers can just update this

camera package without having to touch anything else in the app. It keeps things simple, organized, and easier to handle, even as Snapchat grows and adds more features.

26d Class Interfaces

Think of class interfaces like the rules of a game. They tell different parts of Snapchat how they should work together, but they don't get into the nitty-gritty of how things are done behind the scenes. Just like how a game has instructions on how to play, but doesn't explain the exact mechanics of every move, class interfaces tell the app what to expect, without worrying about the technical details. This keeps things running smoothly and allows different teams to work on their own parts without stepping on each other's toes.

Content

Here's what you'll find in class interfaces:

- **Method Signatures:** These are like the instructions for each action Snapchat can take. For instance, a method might be something like "sendSnap()", but it won't tell you exactly how the Snap gets sent—just that it happens.
- **Inputs and Outputs:** Each class needs something to work with, like the photo for a Snap or the name of the person you're sending it to. The interface also says what it'll return, like a success message or an error.
- **No Details on How It Works:** The interface doesn't say how things should be done—just that they need to be done. This allows different teams to build the same functionality in different ways, as long as they follow the same rules.
- **Consistency:** By following the interface, everyone knows what to expect from the class. This means you can rely on it to work the same way every time, no matter who's working on it.

Motivation

The reason class interfaces are so important is that they act as the foundation for Snapchat's code to talk to itself. Without them, developers would have to learn how every single piece of Snapchat's code works before they could use it. That's a lot of extra work! With clear interfaces, developers know exactly what to expect and can focus on building the app rather than figuring out how things connect.

Considerations

- **Keep It Simple:** If the interface is too complex, developers will get confused. It should be easy to follow, just like a simple set of instructions.
- **Flexibility Is Key:** Snapchat needs to be adaptable, so the interface should be flexible enough to handle different situations. For example, maybe you send a Snap with a message or without one—the interface should handle both.
- **Don't Tangle Things Up:** The idea behind interfaces is to keep things separate and independent. That way, if one part of Snapchat changes, it won't break the rest of the app.
- **Handle Problems:** Sometimes things go wrong—like when the network drops. A good interface makes sure Snapchat knows how to handle these problems, like giving an error message when something can't be sent.
- **Easy to Maintain:** As Snapchat grows, the interfaces should be able to grow with it. Changes should be easy to make, without disrupting everything else that's already working.

Example

Imagine Snapchat has a **SendSnap Interface**. This would be the part of the code that handles everything involved in sending a Snap. The interface might have a method like “send()”, which needs a photo and a recipient’s information to work. When developers want to send a Snap, they don’t need to know exactly how it gets sent—they just call the “send()” method, and it takes care of the rest. If the way a Snap is sent needs to be changed, developers can update the method behind the scenes without touching the interface, which means other parts of Snapchat won’t be affected. It’s like switching the way a game is played while still following the same basic rules.

IV Test Plans

27 Features to be tested / not to be tested

Features to be Tested in Snapchat:

- **User Authentication Authorization:**

Content

Create an account, log in, and reset your password. Create an account, log in, and reset your password.

Motivation

Assuring users ability to retrieve their profiles and log in securely

Considerations

Examine different circumstances such as fraudulent login credentials account lockouts etc.

Example

Testing the integration of Google and Facebook accounts for login.

- **Sending Snaps (Photos/Videos):**

Content

Emailing to friends, adding text and filters, and uploading images or videos.

Motivation

This is Snapchat's primary feature, and it needs to be tested for dependability and usability.

Considerations

Test various network circumstances, file kinds, and size restrictions.

Example

Sending a friend a video that has been filtered and making sure it shows up properly.

- **Stories:**

Content

Producing, consuming, and becoming acquainted with experiences.

Motivation

Implementing seamless content witnessing and appearing for users.

Considerations

Fluctuating access authorizations and confidential settings.

Example

Testing the accessibility of facts depending on the privacy settings (friends-only, public).

- **Snapchat Filters & Lenses:**

Content

Producing, consuming, and becoming acquainted with experiences.

Motivation

Implementing seamless content witnessing and appearing for users.

Considerations

Fluctuating access authorizations and confidential settings.

Example

Testing the accessibility of facts depending on the privacy settings (friends-only, public).

- **Chat Functionality:**

Content

Sending interactive media (pictures, movies, GIFs), texting, and using Snapchat.

Motivation

Ensuring that intellectual property dissemination and interaction are straightforward for users.

Considerations

Check for notifications, multimedia rendering, and text message delivery.

Example

Using an interactive conversation to compose and receive multidimensional media transmissions.

- **Notifications (Push and In-App)**

Content

Push informs for improvements to narratives, messages as well as or photographs that are new.

Motivation

Ensuring that clients receive notifications on time.

Considerations

Prioritizing and multiple notification announcements (on and disabled).

Example

Testing a push alert that emerges when a friend snaps a picture.

- **Privacy and Security Features:**

Content

Controlling account confidentiality preferences, encrypting records, and preventing and bypassing users.

Motivation

Ensuring independence and protection of personal information of users.

Considerations

Test numerous information utilization rights, encryption stages, ranging and setting for confidentiality.

Example

Establishing whether a blocked user is unable to transmit snaps.

- **Geolocation Features:**

Content

Services centered around location, which includes sharing locations and Snap Map.

Motivation

Examining geolocation information confidentiality and accuracy.

Considerations

Test under various network circumstances and geolocations.

Example

Using Snap Map, peers may share your whereabouts.

- **Snapchat Discover and Ads:**

Content

Exploring Discover content while watching commercials.

Motivation

Ensure that consumers can readily access material and that advertisements are correctly displayed.

Considerations

Verify that the content displays as expected and that the advertising download correctly.

Example

Participating in Discover's funded offerings and advertisements.

Features Not to be Tested in Snapchat:

- **Third-Party Integration:**

Content

Characteristics that work with external services (such as Spotify and the Bitmoji incorporation).

Motivation

Independent software engineers are in possession of accomplishing these integrations.

Considerations

Pay attention to Snapchat's primary features rather than those of other services.

Example

Bitmoji embedding into negotiations and stories.

- **User-generated Content Beyond Snapchat:**

Content

Data shared throughout many channels, including stories published on Snapchat and posted on various other social networking sites.

Motivation

Snapchat should make sure that the content functions only on their platform.

Considerations

Restrict testing to the behavior of content inside Snapchat's ecosystem.

Example

Exporting a Snapchat photo about Instagram (just test Snapchat's uploading aptitude, not Instagram's reception).

- **Server-Side Performance (Unless Critical to User Experience):**

Content

Processing of data, the back end of software server burden, etc.

Motivation

Although the server's efficiency is crucial, front-end testing regularly fails to evaluate it.

Considerations

Integration testing can be used to test for server-related problems that impact the user experience, such as crashes or sluggish performance.

Example

Although assessing server speed at the UI level is usually not required, Snapchat delays could indicate a server-related problem.

28 Pass/Fail Criteria

- **User Authentication & Authorization:**

Pass Criteria

The stakeholder creates an account, refreshes their login credentials, or logs in successfully and fault-free.

Fail Criteria

Account lockout fails, incorrect credentials don't provide the proper error messages, or login takes too long.

Motivation

Assuring users' safe access to their accounts is essential for maintaining their privacy.

Considerations

Verify adherence to security protocols and conduct tests on various devices and networks.

Example

A pass happens if a user is able to change their password and then log in. If an incorrect password does not result in an error message, it would be a failure.

- **Sending Snaps (Photos/Videos):**

Pass Criteria

Sending images or movies to pals is possible, and the recipients get the pictures without any problems.

Fail Criteria

The receiver is unable to access the material, the snap fails to send, or it loads erroneously.

Motivation

Snapchat's main purpose is to allow users to send snaps, hence a seamless experience is crucial.

Considerations

Experiment with various file sizes, formats, and network setups.

Example

If a video clip is transmitted and received without any compression problems, it passes.
If the video quality drops to an unsatisfactory level or doesn't send, fail.

- **Stories:**

Pass Criteria

With respect to the privacy settings they have selected, peers can see tales that users have posted, viewed, and deleted with ease.

Fail Criteria

Stories are fuzzy, don't load, or violate privacy settings.

Motivation

One important interaction element that increases user engagement is stories.

Considerations

Experiment with various file kinds and confidentiality settings (public, friends-only).

Example

Pass whether a buddy can access a story marked "Friends only." If the story is not visible to authorized users

- **Filters and Lenses:**

Pass Criteria

Lenses and filters work on a variety of device models, load rapidly, and apply appropriately.

Fail Criteria

Filters don't apply, load slowly, or don't line up with the user's face.

Motivation

Snapchat's filters and lenses are very well-liked features, and seamless operation improves the user experience.

Considerations

A few things to think about are testing on various devices, network speeds, and facial recognition capabilities.

Example

If a lens tracks the user's face precisely and applies smoothly, it passes. If a lens misaligns or freezes, fail.

- **Chat Functionality:**

Pass Criteria

Chat media and messages are displayed accurately and are sent immediately.

Fail Criteria

Chats that crash, graphics that don't render correctly, or messages that are delayed or never delivered.

Motivation

Users stay connected and involved on the platform thanks to dependable chat functionality.

Considerations

Take into account the handling of emojis and stickers, image and video rendering, and media delivery timings.

Example

If the recipient sees a photo delivered in chat right away, it passes. If the image is distorted or fails to load, fail.

- **Notifications (Push and In-App):**

Pass Criteria

Being notified of snaps, chats, and story updates can be delivered to users in an efficient and precise way.

Fail Criteria

Inaccurate, missing, or delayed notifications.

Motivation

Users stay interested and up to date with the app when they receive timely messages.

Considerations

Experiment with notification channels, device types, and settings.

Example

If a user gets notified right away when a friend publishes a new article, it passes. If you struggle to get the notification or it comes hours later, fail.

- **Privacy and Security Features:**

Pass Criteria

User data is protected by encryption technologies, and privacy choices (such as who can contact, access tales, or see location) are maintained.

Fail Criteria

Data is not encrypted, blocked users can contact, or privacy settings are disregarded.

Motivation

Prevents unwanted access, fosters trust, and safeguards user privacy.

Considerations

Test privacy settings for diverse user interactions on a range of devices.

Example

If a blocked user is unable to transmit snaps, pass. If the prohibited user can still communicate, fail.

- **Geolocation Features:**

Pass Criteria

Snap Map's location-sharing features function well, and user locations are displayed with accuracy.

Fail Criteria

Snap Map violates privacy settings or the location is incorrect.

Motivation

Precise location sharing improves privacy and user engagement.

Considerations

Experiment with various device types, network speeds, and location permissions.

Example

If a user's position on Snap Map is correct, then pass. If location privacy settings are not adhered to, fail.

- **Snapchat Discover and Ads:**

Pass Criteria

Ads appear as planned, and Discover's content loads without hiccups.

Fail Criteria

Content that doesn't load, is inaccurate, or displays advertisements incorrectly are examples of fail criteria.

Motivation

Encourages ad income and guarantees a satisfying user experience with external content.

Considerations

Examine ad performance in a range of network scenarios and test on many devices.

Example

If the advertising and Discover content load without any issues, then pass. Ads that disrupt the layout, are misaligned, or don't load should fail.

29 Approach

Content

Test cases should include both functional and non-functional elements in a range of scenarios to guarantee snapchats features operate dependably and protect user privacy cases should verify seamless delivery quality preservation and privacy compliance for sending snaps and viewing stories even in the event of a network outage or a change in settings while viewing tests of filters and lenses should confirm seamless operation and installation on all devices especially older models or those without location permissions even on devices with limited capacity or when using a lot of media cases for chat and video calls should guarantee message and call dependability with crystal-clear audio and video snap map tests ought to validate precise location updates and privacy safeguards for all kinds of networks tests of the discover advertising and content should show that they load quickly and that the ads are tailored to the users preferences and settings lastly tests of alerts and privacy settings should confirm that modifications take effect immediately and are synchronized across devices while also honoring settings such as ghost mode these test cases guarantee that consumers consistently experience snapchats performance usability and privacy standards

Motivation

Delivering a dependable, seamless experience in Snapchat's dynamic, multimedia-rich environment requires thorough testing. Because the app depends on real-time interactions, thorough testing protects user confidence by guaranteeing that privacy settings work as intended, Snaps arrive on time, and functions work flawlessly on a range of devices and network conditions. Because of this attention to detail, Snapchat's privacy pledge is upheld, multimedia quality in Snaps and Stories is guaranteed, and users can interact with ease and confidence on any device or in any place.

Considerations

It's critical to take into consideration variables like network strength, device variety, and stringent data protection regulations when testing Snapchat. Real-world circumstances like insufficient connection that may affect Snap delivery times, integrating them with older devices that might not be able to employ sophisticated filters and lenses, and adherence to data security regulations like GDPR to preserve user privacy should all be covered in tests. To provide a consistent, private, and effective user experience in all settings, studies ought to glance at battery life, data consumption, and performance under various network types (such as 4G, 5G, and Wi-Fi).

Example

A test case for the "Sending a Snap" functionality would entail capturing a picture or video Snap, adding text or stickers, applying filters, choosing a recipient, and confirming that the Snap is sent and received quickly without sacrificing audio or visual quality. Various network strengths (e.g., 4G, 5G, Wi-Fi, and weak signals), device kinds (e.g., iPhone 12, Samsung Galaxy S21, and earlier versions), and file sizes (e.g., high-resolution photographs or movies) would all be covered in this test. In order to guarantee that Snaps are sent

promptly and error-free, it would also evaluate behavior under varying network conditions, giving all users, regardless of device or connectivity, a seamless experience.

30 Suspension and resumption

Content

Make thorough test cases to investigate how Snapchat handles feature suspension when a user leaves the app and resumption upon their return. In order to determine whether Snapchat can smoothly preserve each feature's state, these test cases should encompass typical scenarios such as taking a Snap, watching a Story, making a video call, and perusing Discover material. For instance, the user may receive a call or switch to another app while filming a Snap; based on expected behavior, Snapchat should either let the recording continue where it left off or notify the user that they need to restart.

When viewing a Story in a test case, the Story should pause at the exact moment that you transition to a different app. The Story should continue from that precise frame when you return to Snapchat without refreshing or sacrificing quality. It should be confirmed that Snapchat seamlessly reconnects the connection upon returning and preserves audio and video quality without lag or desynchronization in situations where video calls are interrupted, such as exiting to check a notification or reply to a text. These scenarios should also verify that features appropriately adjust to variations in network strength or connectivity, that data—such as text, stickers, or filter effects applied to a Snap—remains intact, and that connections are quickly restored.

Motivation

Because users regularly multitask by switching between applications or taking calls while utilizing Snapchat's capabilities, it is imperative that the app manage suspension and resume. Maintaining a flawless user experience requires confirming that Snapchat preserves the precise state of interrupted actions—such as an ongoing video conversation, paused Story,

or Snap in progress—without requiring forced restarts or data loss. Users should be able to return to Snapchat without any delays, content reloading, or loss of modifications if they briefly exit the app to reply to a message or check a notification. Users' trust in Snapchat is strengthened by its dependability, particularly during real-time interactions like video conversations and Story sharing, where disruptions are frequent and can ruin the experience if not managed well.

Considerations

Tests must cover common situations such as getting a call while taking a snap or briefly navigating to another app to view a notice before going back to Snapchat; these tests ought to confirm that recording may either continue without any issues or notify the user if a restart is required. They should also test the stability of reconnection in services like video calls, making sure that there is no noticeable latency or reloading delays and that the audio and video quality is constant upon restarting. Additionally, tests should keep an eye out for surges in data usage and make sure that reconnecting to content-heavy features like stories is handled efficiently in order to guarantee that Snapchat provides a dependable experience across a range of user contexts. It is important to account for variations in device capabilities such as RAM and processing capacity, app versions, and shifting network conditions from Wi-Fi to cellular data or bad signal.

Example

Starting a video Snap recording and purposefully switching to another app—like the messaging app or a browser—mid-recording could be one way to test suspension and resume in Snapchat. The test should verify whether the video recording resumes precisely where it was paused or whether Snapchat clearly indicates that the recording has to be restarted because of the interruption. Features like video calls, where the user may quickly disconnect, should undergo similar testing. When the user returns, the call should automatically re-

connect without any audio-visual lag, guaranteeing constant quality. When watching a Story, navigating to another app and then returning to Snapchat should allow you to restart the Story from the precise moment it was interrupted, without requiring you to reload or lose any data. This guarantees that Snapchat can seamlessly manage real-time disruptions across various device and network circumstances, preserving the integrity of the user experience.

31 Testing materials (hardware / software requirements)

Content

An extensive assortment of devices spanning various models, OS versions, and dimensions should be part of the necessary hardware in order to perform thorough testing for Snapchat. In order to test compatibility and performance differences between platforms, this includes a variety of Android smartphones from various brands (such as the Samsung Galaxy, Google Pixel, and OnePlus) and several iPhone models (such as the iPhone 8 through the most recent iPhone models). To guarantee that Snapchat's multimedia features function properly on both high-end and low-end devices, the devices should reflect a range of processing capacities.

The most recent version of Snapchat as well as a few older versions are needed for software verification in order to ensure backward compatibility and consistent performance across upgrades. To guarantee compatibility with outdated operating systems, testing on many iterations of iOS and Android is crucial. While debugging tools like Android Studio and Xcode help with tracking app performance and logging errors, emulators for iOS and Android may help replicate a wider variety of device situations. In order to evaluate Snapchat's behavior under actual connectivity constraints, network simulators are also required to recreate various conditions, including low bandwidth, latency, and multiple network types (e.g., 3G, 4G, 5G, Wi-Fi).

Motivation

To make sure Snapchat provides a stable and consistent satisfying experience to its large user base, testing across a multitude of software and hardware configurations is critical. Snaps, Stories, and video calls are just a few of Snapchat's resource-intensive and processing-intensive multimedia capabilities. As a result, testing on a range of devices—from expensive models to older models—helps confirm that performance is consistent across a range of processing capacities. By enabling the behavior of the application to be assessed under many network circumstances and strengths, network simulators and debugging tools further improve testing. This maintains usability and efficiency across all user situations by guaranteeing that Snapchat's real-time functions, such as video calls and Snap sending, work properly even on slower or erratic connections.

Considerations

It's important to consider possible hardware constraints on low-end smartphones when testing Snapchat because these devices may not be able to handle resource-intensive features like filters, lenses, or video processing because of their reduced processing power and memory capacity. In order to make sure that these gadgets don't malfunction or impair the user experience, test cases should include situations in which they manage intricate multimedia elements. Additionally, Snapchat's basic capabilities, including taking and viewing Snaps or making video calls, can be greatly impacted by device-specific peculiarities like inconsistent display (e.g., color calibration), variable camera quality, and audio performance. To guarantee that Snapchat's real-time features—such as sending Snaps, viewing Stories, or conducting video calls—continue to perform without consuming large amounts of data or suffering from noticeable quality loss, testing must also account for real-world network fluctuations, such as sluggish connections or signal drops. This guarantees a smooth and dependable Snapchat experience for consumers, irrespective of device or network conditions.

Example

A variety of devices, including both more recent versions like the iPhone 14 and Samsung Galaxy S21 as well as more traditional models like the iPhone 8, should be included in the test setup in order to evaluate Snapchat's video call function. These devices should be connected to different network circumstances, such as Wi-Fi, 4G, and 3G. Regardless of device specs or network strength, the test should confirm that the audio and video quality stay clear and constant. Additionally, it should guarantee seamless video call resumes during disruptions like network outages or app switching. In order to verify backward compatibility and guarantee peak performance throughout Snapchat's heterogeneous user base, testing should also be done on several iterations of the app and OS versions (iOS and Android). This guarantees dependable video calls with negligible quality deterioration for all customers, irrespective of their device or network conditions.

32 Test cases

Content

Test cases should include both functional and non-functional aspects of Snapchat's operation to guarantee a comprehensive assessment verifying essential functions like sending snaps viewing stories conducting video calls and applying filters should be the main goal of functional test cases to guarantee resilience in difficult situations these test cases should outline the inputs such as images videos and filters and expected outputs such as successful snap delivery and media quality as well as edge cases such transmitting snaps with insufficient storage or unstable network conditions to guarantee consistent behavior across various speeds non-functional test cases should assess Snapchat's performance under various circumstances such as network types eg wi-fi 3g 4g battery usage over extended use app starting time to guarantee quick loading and device compatibility with different models and os versions such as ios and android should all be tested further the app's continued excellent performance and usability across a variety of devices and user contexts will be ensured by these testing.

Motivation

Test cases are designed to make sure Snapchat works flawlessly on a range of devices, networks, and operating systems. To ensure that users can reliably send Snaps, view Stories, and make video calls without any disruptions, extensive testing is necessary given Snapchat's emphasis on real-time interactions and multimedia features. Especially on low-end smartphones, testing should concentrate on making sure the app runs smoothly with few lags, quick startup times, and optimal battery utilization. By assessing these variables, testing contributes to the maintenance of Snapchat's dependability and user experience in various settings, offering a broad spectrum of users constant functioning and performance.

Considerations

The performance of multimedia features like Snaps, Stories, and video calls is directly impacted by hardware variability, therefore test cases should take this into consideration. Examples of this variability include variations in device processing power, camera quality, and display resolution. To make sure that real-time features function at their best even on slower connections, testing should also cover a range of network situations (such as 3G, 4G, and Wi-Fi). In order to guarantee that sensitive user data is handled effectively and that data usage stays efficient, especially in situations with limited bandwidth or heavy network demand, privacy settings and security procedures must also be thoroughly verified.

Example

Taking a picture, adding a filter, and then emailing it to a buddy would be an example of a Snap test case. It is anticipated that the Snap would be supplied successfully, without any delays or deterioration of quality. To make sure the Snap still sends without problems or data loss, edge scenarios like delivering huge image files or using shaky network connections

should be verified. To ensure that the Snap works consistently and as intended for all users, more testing should concentrate on device compatibility, confirming that it appears correctly across devices with different screen sizes, camera types, and operating systems.

Functional Test Cases

1. Send a Snap (Image)

- **Input:** Image, recipient selected.
- **Expected Result:** Snap is delivered successfully to the recipient.

2. Send a Snap (Video)

- **Input:** Video, recipient selected.
- **Expected Result:** Snap is delivered successfully to the recipient.

3. Add Filters to Snap

- **Input:** Image or video, filter applied.
- **Expected Result:** Snap is delivered with the selected filter applied.

4. Send a Snap with Text Overlay

- **Input:** Image or video with added text.
- **Expected Result:** Snap is delivered with text overlay visible.

5. Send a Snap with Insufficient Storage

- **Input:** Image or video with limited device storage.
- **Expected Result:** Error message; snap is not sent.

6. Send a Snap with Unstable Network

- **Input:** Image or video under fluctuating network conditions.
- **Expected Result:** Snap retries sending, or an error message is shown.

7. Send a Snap to Multiple Recipients

- **Input:** Image or video, multiple recipients selected.
- **Expected Result:** Snap is delivered to all selected recipients.

8. Send a Snap with Location Filter

- **Input:** Image or video with location filter applied.
- **Expected Result:** Snap is delivered with the location filter.

9. View Received Snap

- **Input:** Open received snap.
- **Expected Result:** Snap opens and displays correctly.

10. View Story

- **Input:** Select a friend's story.
- **Expected Result:** Story plays without lag or interruptions.

11. Send Video Call

- **Input:** Initiate a video call with a contact.
- **Expected Result:** Video call connects successfully with clear audio and video.

12. Accept Incoming Video Call

- **Input:** Accept an incoming video call request.
- **Expected Result:** Call connects and displays audio and video.

13. Apply Lens Filter During Snap

- **Input:** Image or video, apply lens filter.
- **Expected Result:** Snap is delivered with lens filter effect.

14. Mute Microphone During Video Call

- **Input:** Mute option selected during video call.
- **Expected Result:** Microphone is muted without disconnecting the call.

15. Use Sticker on Snap

- **Input:** Image or video, apply sticker.
- **Expected Result:** Snap is delivered with the sticker.

16. Re-send Failed Snap

- **Input:** Select to resend a snap that failed to send initially.
- **Expected Result:** Snap is sent successfully on second attempt.

17. Receive Screenshot Notification

- **Input:** A friend takes a screenshot of the sent snap.
- **Expected Result:** Notification of the screenshot is received.

18. Send Snap with Sound Filter

- **Input:** Video snap with sound filter.
- **Expected Result:** Snap is delivered with the sound filter.

Non-Functional Test Cases

1. App Load Time

- **Input:** Open the app from a closed state.
- **Expected Result:** App loads within 5 seconds.

2. Battery Usage During Snap Sending

- **Input:** Send snaps continuously for 30 minutes.
- **Expected Result:** Battery consumption is within acceptable limits.

3. App Performance on Wi-Fi

- **Input:** Perform regular tasks on Wi-Fi.
- **Expected Result:** All functions perform smoothly without lag.

4. App Performance on 3G

- **Input:** Perform regular tasks on 3G.
- **Expected Result:** Functions work with minimal lag, no feature fails.

5. App Performance on 4G

- **Input:** Perform regular tasks on 4G.
- **Expected Result:** Functions work smoothly with optimal performance.

6. App Performance with Low Battery

- **Input:** Use the app with battery under 15%.
- **Expected Result:** Core functions continue to work without issues.

7. Device Compatibility with iOS (latest version)

- **Input:** Run the app on the latest iOS version.
- **Expected Result:** App operates without bugs or crashes.

8. Device Compatibility with Android (latest version)

- **Input:** Run the app on the latest Android version.
- **Expected Result:** App operates without bugs or crashes.

9. Device Compatibility with iOS (older version)

- **Input:** Run the app on an older iOS version (supported by Snapchat).
- **Expected Result:** App functions properly without significant issues.

10. Device Compatibility with Android (older version)

- **Input:** Run the app on an older Android version (supported by Snapchat).
-

- **Expected Result:** App functions properly without significant issues.

11. Memory Usage During Video Call

- **Input:** Make a video call lasting 10 minutes.
- **Expected Result:** Memory usage stays within reasonable limits.

12. Stability During Prolonged Use

- **Input:** Use the app continuously for 2 hours.
- **Expected Result:** App remains stable, with no crashes or freezes.

13. App Response with Heavy Media Content

- **Input:** Send multiple high-resolution snaps and videos.
- **Expected Result:** App performs smoothly with no significant delay.

14. Offline Mode Behavior

- **Input:** Open the app without network access.
- **Expected Result:** App displays network error and prevents snap sending

33 Testing schedule

Content

To guarantee thorough coverage, a Snapchat testing schedule should comprise defined phases. Planning, which establishes the goals and scope of testing; test case development, which focuses on creating particular test cases for functional, non-functional, and edge cases; test execution, which runs the developed test cases on various devices and OS versions to verify performance; and defect resolution, which tracks, fixes, and retests issues that are found. Functional testing (such as sending Snaps and watching Stories), non-functional testing (such as battery consumption and performance in different network conditions), and edge situations (such as device restrictions or network fluctuation) should all be given top priority in the timetable. Regression testing should also be considered after every upgrade or bug fix to make sure that new modifications don't interfere with already-existing functionality. To guarantee quality at every level, testing should be done continuously during the development cycle. This guarantees that Snapchat meets user expectations by offering a flawless user experience and operating well in a variety of scenarios.

Motivation

In order to guarantee that Snapchat's features are extensively tested within the necessary project dates, ultimately the diagnostic procedure schedule is essential. Comprehensive testing is made possible by a defined schedule, which lowers the possibility of delays and interruptions by recognizing and addressing problems early in the development process. Snapchat can maintain high quality, user friendliness, and performance by regularly testing the app. This promises that the stakeholder experience is uninterrupted and seamless across a range of devices, operating systems, and network circumstances. Regardless of the setting or device being used, this strategy helps guarantee that Snapchat maintains up to client requirements for dependability and effectiveness.

Considerations

It's important to consider things like device availability when planning a testing schedule because this guarantees that testing is done on a variety of hardware since some features like video calls or snaps could need more thorough testing than others. Feature complexity should also be taken into account to prevent bottlenecks. Resources like testers, tools, and testing environments should be arranged ahead of time to guarantee compatibility and consistent performance. Testing should encompass a range of device models, operating systems, and network types such as Wi-Fi, 4G, and 3G. Re-testing should be given enough time following bug patches and performance assessments, particularly for real-time features where user experience may be impacted by latency or interruptions. Last but not least, cooperation with other teams such as development and UI/UX is essential to guaranteeing that input is included as soon as possible and that the schedule fits in with the project's overall timelines, enabling seamless iteration and release.

Example

- **Weeks 1-2:** Test case development and preliminary test planning, with an emphasis on key functionalities such as sending Snaps, watching Stories, and applying filters. Outlining non-functional tests like battery usage and app launch times is another aspect of this phase.
- **Weeks 3-4:** Functional testing to verify compatibility and find issues on a range of devices and OS versions. Verifying fundamental features like video calls, Stories, and Snaps across several devices would be the main goal.
- **Weeks 5-6:** Conducting battery consumption tests and performance testing across various network conditions (Wi-Fi, 3G, and 4G) to assess Snapchat's behavior in low-bandwidth situations, real-time interactions, and overall resource use.
- **Week 7:** Regression testing following the introduction of new features or problem fixes. This stage makes ensuring that new features function as intended and that fixes don't interfere with already-existing functionality.
- **Week 8:** Complete testing to make sure all systems are ready for release, including app compatibility and backend service integration. Prior to the actual release, this entails verifying user experience, database integration, and server connectivity.

V Project Issues

34 Open Issues

Content

A list of factors that remain uncertain and could significantly impact the product if not addressed. These are issues that have been identified but lack a clear resolution.

Motivation

To bring potential uncertainties into the open, allowing for better visibility and a foundation for effective risk analysis, which supports informed decision-making.

Examples

Our assessment of the new processor's compatibility with our application is still ongoing. Additionally, upcoming government changes may alter responsibilities for motorway maintenance, potentially affecting operational requirements.

Considerations

Identify unresolved issues from requirements gathering that could impact development. Note any anticipated changes within partner organizations or systems directly linked to the product. Address pending legislative changes that could affect functionality or compliance. Monitor updates from hardware or software providers that may influence compatibility or performance.

35 OfftheShelf Solutions

35a ReadyMade Products

Open issues are like a list of “unknowns” Snapchat’s development team is keeping an eye on. These are questions or potential problems that don’t have clear answers yet, but could impact the app in significant ways. It’s all about being proactive—by bringing these uncertainties to light, the team can keep tabs on anything that could create challenges down the road.

Content

Here’s what’s typically included:

- **List of Uncertainties:** This could be anything that hasn’t been fully figured out yet, like whether Snapchat’s latest update will work smoothly with the newest iOS version or if the app’s servers can handle a sudden spike in users.
- **Possible Impact:** These uncertainties could be minor or major, depending on how things turn out. For example, if a new hardware release changes Snapchat’s performance, it might require a fix or even a redesign in certain areas.
- **Watch for Changes:** The team might also keep track of other things outside of Snapchat itself, like new laws on privacy or platform policies on Apple or Google that could influence how Snapchat functions.

Motivation

The main reason to keep a list of open issues is to make sure no surprises catch the team off guard. By identifying possible uncertainties early, the team can factor them into risk assessments and be prepared with solutions. This transparency also helps everyone stay informed on potential risks, so they’re not left scrambling if an issue becomes more pressing.

Examples

- **Compatibility Questions:** Let's say Snapchat is waiting to confirm that the new iPhone camera tech will work seamlessly with its filters. This is an open issue until they're sure it's compatible.
- **Changing Rules:** If there's talk of new data privacy laws that could impact Snapchat's location-sharing feature, it would be logged as an open issue until there's clarity on what Snapchat needs to adjust.
- **System Integration:** If other organizations Snapchat relies on (like cloud providers or content moderation tools) are planning big changes, the team would keep track of this to anticipate any adjustments.

Considerations

- **Unresolved Issues from User Research:** If users are experiencing a problem that the team hasn't found a clear solution for yet, it would be logged here until resolved.
- **External Changes:** Any possible shifts from outside companies or platforms Snapchat depends on (like iOS updates or new Google Play policies) might impact how Snapchat operates.
- **Legal and Regulatory News:** If there are rumors about new laws or rules that could influence Snapchat's features, they go on this list to ensure the team is prepared to respond.
- **Industry Developments:** The team might also track industry trends, like new processors or AR technology that could change how Snapchat's features perform.

35b Reusable Components

Reusable components are like Snapchat's "shortcut" parts—ready-made tools and bits of code that can be used to build different parts of the app without starting from scratch every time. Whether Snapchat's team created these in-house or they're reliable resources from outside, these components are tried, tested, and ready to go. Using them saves time and helps make sure everything works consistently across the app.

Content

Here's what's involved in picking reusable components:

- **Candidate Components:** These are the specific tools or libraries Snapchat can use. For example, if there's a module for processing images, it could be used for filters in Snaps and Stories, making things smoother and keeping the quality high.
- **Where They Come From:** Snapchat can pull these from their own internal code library or from trusted external sources that specialize in things like handling images, verifying user identities, or sending notifications.

Motivation

The main reason for using reusable components is pretty simple: efficiency. When Snapchat reuses code that's already been built, the team saves time and avoids doing the same work over and over. Plus, these components have already been tested, so the team knows they'll work well and stay consistent across different features.

Example

Some examples of reusable components Snapchat might use:

- **Login and Security:** If there's a secure way to verify users already available, Snapchat can plug it in to handle logins and make sure users' data is protected without creating a new system from the ground up.
- **Photo Filters and Effects:** Snapchat could use a photo and video processing library that already knows how to handle things like adding filters. This means users get a smooth, reliable experience with every Snap and Story.
- **Notifications:** Instead of building a notification system every time they need it, Snapchat could use an existing tool that knows how to send reminders, alerts, or messages in a way that's easy for users to receive and manage.

35c Products That Can Be Copied

"Products that can be copied" are basically a list of tools or apps similar to Snapchat that we could legally use as inspiration. If there's a product out there with a feature or setup that Snapchat could adapt or modify, it might save us a lot of time and effort. By looking at what's already out there, Snapchat's team can get ideas, see what works, and possibly find shortcuts to avoid reinventing the wheel.

Content

- **List of Similar Products:** This could include apps or features that Snapchat could adapt. For example, if there's a messaging feature from a similar app that's known for being secure and user-friendly, it might be worth exploring.
- **What Can Be Used or Modified:** The idea isn't to copy an entire product but to identify parts we could legally use or modify to save time. This could be a layout, a security feature, or an interface design.

Motivation

The motivation is to save development time by seeing if there's something ready-made we can learn from. If we can borrow a good feature or design and modify it for Snapchat, that means less work for our team and potentially a quicker route to a polished result.

Example

For instance:

- **Customer Service Features:** Let's say another app has a great customer service chatbot that's easy to interact with. Snapchat could look at using a similar setup to help answer common questions.
- **Story Functionality from Similar Apps:** If a competitor has found an efficient way to let users create stories, Snapchat's team might review it to see if there's something we can implement in a way that fits Snapchat's unique style.
- **Notifications and Alerts:** A feature like real-time notifications could be inspired by another app known for its alert system, so users never miss a message or Snap.

Considerations

- **Adaptability:** Sometimes, the base product might not be a perfect fit for Snapchat, but if it's adaptable, it might be worth modifying.
- **Quality Check:** Since we'd be relying on a different product as a base, it's crucial to make sure it's a high-quality system. Otherwise, we risk copying something with flaws.

- **Looking Broadly:** Even if a ready-made product doesn't exist, by checking out other solutions in the tech space, we might find something with a similar "essence" that could inspire a better approach to Snapchat's needs.

36 New Problems

36a Effects on the Current Environment

When Snapchat brings in a new feature or system, it's important to consider how this will impact the setup that's already in place. This means looking at both the positive effects and any potential issues the change could create. This section isn't just about what the new product will do—it's also about what it shouldn't do. The goal is to spot any possible conflicts or disruptions before they become real problems.

Content

Here's what goes into understanding the effects on Snapchat's current environment:

- **Impact on the Existing System:** We need to look at what parts of Snapchat might be affected by the new feature. For example, a new scheduling system for Snaps might affect how users or engineers interact with the platform.
- **Limitations:** It's also about understanding what the new feature shouldn't do. If it could accidentally interfere with other functions, that's something to flag early on.

Motivation

The main reason for assessing these effects early is to prevent surprises later. By understanding the impact of new features on the current environment, Snapchat's team can

identify conflicts before they become real obstacles. It's all about a smooth transition that doesn't disrupt user experience or existing workflows.

Examples

Here are a few scenarios:

- **Scheduling Changes:** If Snapchat rolls out a new scheduling feature, it might change how engineers manage content or influence the timing of Snap stories.
- **Potential Workflow Changes:** Adding new processes or tools could mean adjustments for the development or support teams, so the impact on their workflow needs to be considered upfront.

Considerations

- **Compatibility with Existing Systems:** Could the new feature create problems with the current systems in place? Any interference would need to be addressed.
- **Impact on People:** Is there a chance the change could affect anyone's work, either positively or negatively? For example, will new tools require additional training for support teams?
- **Clear Model of Changes:** To make sure everyone understands how the new feature will affect things, a model or outline can make it easier to visualize the impact across different areas of Snapchat's ecosystem.

36b Effects on the Installed Systems

When Snapchat introduces a new feature or system, it almost always has to work alongside what's already in place. This means looking at how the new and existing systems will interact and making sure they're compatible. The goal is to spot any issues that could come up when these systems work together so that everything runs smoothly without unexpected problems.

Content

Here's what's involved in checking the effects on installed systems:

- **Interface Specifications:** We need to look closely at how the new system will connect with Snapchat's current setup. For example, if we're adding a new messaging tool, we'd need to make sure it doesn't interfere with the existing notification system or profile management.
- **Compatibility:** The aim is to ensure the new and old systems communicate effectively without causing conflicts.

Motivation

The main reason for this evaluation is that very few new systems are created to operate in isolation. Snapchat's features usually need to fit in with older, trusted systems to keep the user experience seamless. By examining the existing setup and identifying any compatibility issues, Snapchat can make sure everything works in harmony.

Examples

A few examples could include:

- **Messaging Features:** If Snapchat adds a new messaging functionality, it needs to work well with the current messaging system. Otherwise, there could be issues with message delivery or duplicate notifications.
- **User Profile Integration:** Adding a feature that involves user data, like customized filters, would need to work smoothly with the existing profile setup. The last thing Snapchat would want is for user data to get out of sync.

Considerations

- **System Compatibility:** Will the new system be able to connect with the current Snapchat systems without causing disruptions?
- **Smooth Data Flow:** Will data need to be transferred between the new and existing systems? If so, ensuring accurate and secure data flow is crucial.
- **Conflict-Free Coexistence:** It's essential to verify that the new system won't interfere with the old one. If they're likely to clash, a workaround or modification might be necessary.

36c Potential User Problems

When Snapchat adds something new, we have to think about how it might affect users who are already used to the current version. Sometimes, changes can cause unexpected frustrations or confusion for people who rely on things working a certain way. This section

is about spotting any issues current users might experience and deciding what to do to help them adapt smoothly.

Content

Here's what we look at regarding potential user problems:

- **Possible Reactions:** We need to think about how users might react negatively. For instance, if Snapchat's layout or core functions are updated, regular users might initially struggle to find familiar features.
- **Adjustments and Warnings:** If a feature might confuse or frustrate users, we could plan to add in-app prompts, tutorials, or a way to switch back to the old setup (if possible).

Motivation

The motivation here is to ensure Snapchat's changes don't catch users off guard. Some users rely heavily on certain features or interfaces, and an unexpected shift could lead to a bad experience. By recognizing any potential issues ahead of time, Snapchat can address them proactively, helping users adjust and reducing any initial frustration.

Examples

Here are some examples of possible user reactions:

- **Interface Changes:** If Snapchat updates the look or layout, regular users might feel disoriented at first, struggling to find features they're used to.

- **New Features that Interrupt the Flow:** Adding a new step in the messaging process, for instance, could feel inconvenient to users who are used to quick and straightforward interactions.
- **Data Privacy Changes:** If a feature involves new data-sharing settings, some users might be concerned about privacy and need reassurance or an option to control these settings.

Considerations

- **User Comfort:** The new feature or design disrupt familiar parts of the app that users rely on.
- **Clear Guidance:** User must get clear guidance using tutorials, prompts, or help messages to make the transition easier
- **User Feedback:** It's a good idea to monitor feedback and see if users are facing unexpected issues, so we can respond quickly if needed.

36d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

When Snapchat introduces a new feature or updates existing technology, there can be some unexpected limitations in the tech setup or organization that could hold things back. This section is about identifying potential problems early on—before we're too deep into implementation—to prevent headaches later.

Content

- **Technical Constraints:** Is there anything about Snapchat's current infrastructure that might struggle to support the new feature? For instance, if the update demands a lot of processing power, we need to check if our servers can handle it.
- **Organizational Fit:** If the new feature changes how teams or departments operate, it's worth examining if they're set up to adapt smoothly.

Motivation

Spotting these limitations early allows Snapchat to prepare for or work around potential issues before they cause delays. By considering what might restrict the effectiveness of a new feature, we can make adjustments or develop alternatives that ensure the rollout goes smoothly.

Examples

- **Server Capacity:** If a new feature increases data usage or user engagement, Snapchat's current servers might not be powerful enough to support the added load. This would mean upgrading our infrastructure or optimizing the feature to work within existing limits.
- **Device Compatibility:** Certain features might not work well on older devices. If a feature relies on newer technology (like augmented reality), it's worth ensuring it doesn't make the app unusable for people with older phones.

- **Bandwidth and Power:** Some updates might use more bandwidth or battery power, impacting users on limited data plans or with low battery capacity.

Considerations

When planning new features, Snapchat must consider infrastructure needs to determine if any hardware or software upgrades are essential for smooth performance. Device limitations are also key, as some features might not work on older phones, potentially excluding a segment of users. Additionally, it's important to assess if the feature could affect overall app speed or stability, as this could impact the Snapchat experience for users across the board. Each of these factors helps ensure that new additions are accessible, efficient, and compatible with Snapchat's existing system.

36e FollowUp Problems

When Snapchat introduces new features, it's essential to look closely at any potential issues that could become stumbling blocks. This section is all about identifying situations where things might not go as planned, allowing the team to build a safety net around possible risks.

Content

Here are the main areas we'll cover:

- **Managing Demand:** Will the new feature create a surge in usage that Snapchat's servers and support systems might struggle to handle? It's important to forecast demand accurately and prepare infrastructure accordingly.
- **Legal Compliance:** The particular feature open Snapchat up to any unexpected legal challenges, especially in different regions with strict regulations on data or user privacy.

- **Compatibility with Hardware:** Is there a chance that certain devices, especially older models, might have trouble running the new feature smoothly? Ensuring compatibility helps us avoid issues where some users experience crashes or lag.

Motivation

Spotting these follow-up problems early on allows Snapchat to be proactive, making adjustments before issues arise. This is especially helpful for preventing large-scale disruptions or compliance issues that could be costly to fix later on.

Considerations

- **Capacity Planning:** Are our servers and support teams ready if there's a sudden demand spike? Preparing for scalability makes sure that users have a smooth experience even during heavy use.
- **Staying Legally Sound:** Could this new feature expose Snapchat to new legal requirements or scrutiny? As we expand and update, it's critical to keep compliance in mind to avoid any penalties or operational setbacks.
- **Device Testing:** Can we ensure that this feature will work well across all devices? Testing for compatibility helps prevent performance problems, especially on older or less common devices.

37 Tasks

37a Project Planning

When planning a new feature or update for Snapchat, we outline the full process—from development to delivery—so that everyone involved has a clear roadmap. This includes

details on each phase, resources required, and any specific adjustments based on Snapchat's unique needs. A visual process flow can help show the main steps and how each team or task connects, making the big picture easy to follow.

Content

We break down the project's life cycle to show:

- **Project Phases:** Each stage, from design and development to testing and release, is mapped out to ensure smooth progress and accountability.
- **Timeline and Resources:** We estimate the time and resources needed for each part of the project. These are linked to specific features or functions, allowing us to allocate support based on priority areas.
- **Additional Needs:** Elements like data migration, user training, and seamless integration are planned in advance. These are often forgotten but are crucial to making the feature launch as smooth as possible.

Motivation

A clear plan sets expectations across teams and helps everyone understand the overall approach. This ensures that we're all moving in the same direction and prepared for each step, reducing misunderstandings or delays later on.

Considerations

- **Standard vs. Custom Processes:** While Snapchat's standard development process is reliable, each new feature might come with unique needs requiring adjustments to our usual approach.

- **Time and Resource Estimates:** With requirements in mind, we make realistic time and resource estimates for each task, helping keep the project on schedule and fully supported.
- **Often Overlooked Needs:** Things like data conversion, user training, and phased rollout are essential for user adoption and long-term success, so we account for them early on.

37b Planning of the Development Phases

- **Phase 1: Design**
 - **Phase Name:** Design
 - **Target Completion Date:** [Insert Date]
 - **Operating Environment Components Included:** Server infrastructure, design tools (e.g., Figma, Sketch), user interface mockups, and AR testing environment.
 - **Functional Requirements Included:** User interface design, feature flow, interaction design.
 - **Nonfunctional Requirements Included:** System architecture alignment with scalability, initial security frameworks, and app responsiveness targets.
- **Phase 2: Content**
 - **Phase Name:** Content
 - **Target Completion Date:** [Insert Date]
 - **Operating Environment Components Included:** Content creation tools (e.g., Adobe Creative Suite, Canva), social media platforms for content distribution, database for storing multimedia content.

- **Functional Requirements Included:** Creation and integration of multimedia content (images, videos, filters, effects), integration of user-generated content.
 - **Nonfunctional Requirements Included:** Optimization of content for various devices, ensuring that content meets the app's guidelines and community standards, ensuring fast loading times for media.
-
- **Phase 3: Development**
 - **Phase Name:** Development
 - **Target Completion Date:** [Insert Date]
 - **Operating Environment Components Included:** Development environment, source control systems (e.g., GitHub), staging servers.
 - **Functional Requirements Included:** Core feature implementation, API integrations, data synchronization with backend.
 - **Nonfunctional Requirements Included:** Optimized code for performance, cross-platform compatibility (iOS, Android), and security protocols for user data protection.
-
- **Phase 4: Testing**
 - **Phase Name:** Testing
 - **Target Completion Date:** [Insert Date]
 - **Operating Environment Components Included:** Testing frameworks (e.g., Jest, Appium), staging servers, user beta testing groups.
 - **Functional Requirements Included:** Validation of features against requirements, user flow testing, feature performance under load.
 - **Nonfunctional Requirements Included:** Usability testing, stress tests on backend servers, load balancing, battery usage testing on various devices.

- **Phase 5: Launch**

- **Phase Name:** Launch
- **Target Completion Date:** [Insert Date]
- **Operating Environment Components Included:** Production servers, monitoring tools (e.g., New Relic), app stores (Google Play, Apple App Store).
- **Functional Requirements Included:** Full deployment of feature to all users, app update submissions.
- **Nonfunctional Requirements Included:** Monitoring for real-time performance, user feedback channels, scalability for increased traffic post-launch.

Considerations

- **Hardware Requirements:** The hardware needed for each phase will be determined based on the specifications of the feature. For example, AR functionality might require updated smartphones or specialized testing environments.
- **Software Requirements:** Tools for design, content creation, development, and testing must align with the feature's complexity. This includes app development platforms, design software, testing frameworks, and deployment systems.
- **Timing:** The deadlines for each phase are critical to ensure a smooth transition between stages and to keep the project on track. Adjustments may be made based on emerging requirements or unforeseen delays.

38 Migration to the New Product

38a Requirements for Migration to the New Product

Overview

This part covers the key actions and timeline needed for moving Snapchat to the new system

or feature update, aiming to keep things running smoothly with no major interruptions for users.

Purpose

By planning out the migration steps carefully, Snapchat can shift to the updated system seamlessly, ensuring that data is secure and that the process is efficient.

Key Areas to Address

- **Phased Rollout:** Break the migration into steps or phases, outlining which parts will go live at each stage. This phased approach lets us gradually roll out the new system without overwhelming the infrastructure.
- **Data Transfer:** Prepare programs or processes to securely move important data (such as user accounts, snaps, and settings) from the current system to the new one, making sure nothing important gets lost.
- **Safety Backups:** Set up temporary backup systems so that critical data is fully protected during the transition.
- **Timeline for Implementation:** Lay out the timeline for each key stage of the migration, from initial setup through to the final, complete switchover.
- **Parallel Running:** If needed, run the old and new systems side-by-side temporarily to catch any issues early and ensure the new system is stable before fully replacing the old one.
- **Additional Support Needs:** If extra hands-on help or specific expertise is needed for the transition, arrange this in advance to support a smooth migration.
- **Gradual Shutdown of the Old System:** Plan the retirement of the old system carefully, ensuring all necessary data and functions are transferred fully to the new setup before completely shutting it down, so that users experience a seamless transition.

38b Data That Has to Be Modified or Translated for the New System

Overview

This part outlines the key data tasks needed to ensure Snapchat's data is ready for the new system. It involves modifying or translating certain data so that it aligns with the requirements of the upgraded technology.

Purpose

Identifying the necessary data modifications helps Snapchat avoid unexpected challenges and ensures that all data is correctly prepared for the new environment.

Data Tasks

- **Current Data Setup:** A clear overview of how data is currently stored, including the technology Snapchat uses today to manage and organize this data.
- **New Data Requirements:** A description of the new system's structure, specifying how data will be stored, accessed, and maintained moving forward.
- **Data Translation Steps:** Outlines the specific tasks needed to convert existing data formats or structures into the new system. This includes transforming data fields, reformatting as necessary, and ensuring compatibility with the upgraded environment.

- **Challenges to Anticipate:** Recognize any foreseeable issues, such as data loss risks or compatibility problems, and plan steps to address them, maintaining data integrity throughout the process.

39 Risks

Every project has risks, and Snapchat's development is no exception. Risk doesn't always mean trouble—sometimes taking calculated risks is essential to move forward. The goal here is to turn risks into manageable pieces by predicting what might happen, understanding the likelihood, and setting up backup plans. Problems arise only if we ignore risks and let them surprise us down the road. Good risk management means spotting potential issues early, planning for them, and keeping a close eye as the project progresses.

Identifying Snapchat's Key Risks

This section is where we outline the biggest and most likely risks for Snapchat's project. Drawing from risk assessment guides, here are a few to keep an eye on:

- **Data and Metrics:** Misinterpreted or inaccurate data can skew project decisions.
- **Tracking Gaps:** If we don't measure effectively, it's hard to see if we're on track.
- **Schedule Pressure:** Deadlines that are too tight can lead to stress and impact the quality of work.
- **Management Gaps:** Challenges in leadership or planning can slow progress or create unnecessary complications.
- **Budget Estimation Issues:** Underestimating costs could lead to budget shortfalls.
- **Overreliance on One Solution:** Counting on a single approach or tool can leave gaps if it doesn't work as expected.
- **Scope Creep:** New ideas or requests added over time without adjusting plans can create delays.

- **Quality Control:** Meeting Snapchat's quality standards requires constant attention and adaptation.
- **Project Continuity:** There's always a risk of interruptions or cancellations due to unexpected changes.

We'll look at each of these to gauge how likely they are to cause issues and how serious their impact might be on the project's timeline and budget. By thinking through these risks now, Snapchat can handle challenges smoothly and keep the project on track.

40 Costs

When Snapchat introduces new features or updates, specific cost factors must be taken into account to ensure the project is completed efficiently and within budget.

- **Development Costs:**
 - **Engineering Salaries:** Costs associated with the development team required to build and implement new features, such as AR filters, security updates, or design changes. For example, if a team of engineers works on a feature for 6 weeks, the total salary expense for those engineers must be calculated based on their hourly or monthly rate.
 - **Design Costs:** Designers who work on the user interface (UI) and user experience (UX) to ensure the new feature is visually appealing and intuitive. This may also involve costs for graphic designers if custom visuals or animations are created for a feature like Discover content.
- **Infrastructure Costs:**
 - **Server Upgrades:** If the new feature increases user activity or engagement, Snapchat will need to evaluate if the current server infrastructure can handle the increased load. This might involve scaling up cloud services, adding more

servers, or upgrading database storage systems. For example, if a new feature results in a 30% increase in data usage, the server costs will rise accordingly.

- **Data Storage:** Additional storage costs for storing user data generated by new features (e.g., Snap Streaks, videos, and Stories), which require more cloud storage space. If new video or AR features are introduced, data storage requirements may increase significantly, thus raising the overall infrastructure cost.

- **Testing and QA Costs:**

- **Quality Assurance:** Testing the feature across different devices, operating systems, and scenarios to ensure everything works smoothly. This may involve using automated testing tools and manual testers. If Snapchat is rolling out a new filter or camera functionality, QA teams will need to run tests on various device models (iOS and Android) to ensure compatibility and reliability.
- **Bug Fixing:** There will also be costs associated with fixing bugs identified during testing and after the feature is released, including the hiring of additional developers or QA engineers to address issues that arise post-launch.

- **Marketing and Launch Costs:**

- **Advertising:** Costs for promoting the new feature through various channels, such as social media ads, influencers, and in-app notifications. If Snapchat is launching a new AR filter, it might partner with influencers to showcase how the filter works, which adds advertising costs. A campaign budget might include costs for ad placements, creative development, and influencer partnerships.
- **Launch Events:** Costs related to the launch event for a feature or system update, such as a press release, online event, or community outreach. If a

feature has broad public interest, Snapchat might need to allocate funds to host an online event or press briefings to generate user excitement.

- **Ongoing Maintenance and Support:**

- **Feature Updates:** Once the feature is launched, there will be continuous costs for updating and maintaining the feature, including security patches, bug fixes, and potential improvements based on user feedback. For instance, if a new feature creates unexpected performance issues on certain devices, Snapchat may need to fix the issue and release updates, which involves costs for development and testing.
- **User Support:** Snapchat may need to allocate resources to support teams that handle user inquiries or complaints related to the new feature. This might involve expanding customer service teams to address increased queries or issues related to the feature.

- **Training and Documentation:**

- **Internal Training:** If the new feature impacts how Snapchat employees interact with the app (e.g., customer support, moderation), there will be training costs. This could involve educating the customer service team on handling new user queries or technical training for engineers working with new systems.
- **Documentation:** Costs for creating user manuals, help center articles, or video tutorials explaining the new feature to users. If Snapchat introduces a complex new feature, the cost of developing clear and helpful documentation will be necessary for a smooth user experience.

41 Waiting Room

Sometimes great ideas don't make it into the next version of Snapchat, whether because of time constraints or complexity. The "Waiting Room" is where we keep track of these future ideas, ensuring they aren't forgotten. This section gathers potential improvements and innovative suggestions from users and teams that won't be in the current release but are kept as possibilities for later versions. This way, Snapchat captures user and team creativity without overloading the current development phase.

Why It's Important

Keeping these ideas organized helps Snapchat plan for future updates, offering a clear vision for ongoing improvements and managing expectations. It reassures users and stakeholders that their ideas are valued and considered for later releases, even if they're not included in the immediate rollout.

Prioritizing the Waiting Room

As we evaluate these waiting-room features, we often prioritize the “low-hanging fruit” – enhancements that are high-impact and low-cost – making them strong candidates for upcoming releases. Additionally, features with high user demand or competitive advantage potential will be at the top of the list. This strategy ensures that Snapchat’s development roadmap aligns with both user needs and practical project planning.

42 Ideas for Solutions

This section holds any innovative ideas or concepts that could enhance Snapchat, even if they're not yet prioritized for immediate development. We're keeping them here as future options to revisit, with each entry carefully captured for easy reference. Here's what might be included:

- **New Feature Concepts:** Descriptions of potential new features, like advanced augmented reality (AR) filters, voice-command Snap controls, or enhanced group chat functionalities that could deepen user engagement.
- **UI/UX Prototypes:** Initial sketches or wireframes for potential redesigns of high-use screens, like the Snap Map or Camera view, to improve user experience and accessibility.
- **Competitive Product Insights:** Links or notes on features from similar apps that perform well or solve common user pain points, offering inspiration for Snapchat's future updates.
- **Technology Enhancements:** Ideas for using tools like AI to automate moderation or cloud infrastructure adjustments for improved performance as Snapchat scales globally.
- **Internal Feedback:** Specific suggestions gathered from team discussions that highlight improvements Snapchat's core users or internal teams have found valuable, capturing the voice of our community.

By keeping these solution ideas documented separately, Snapchat can prioritize its current roadmap while having a well-organized set of options ready for future releases. This approach helps us systematically revisit and assess each idea based on evolving user needs and technical advancements, ensuring no promising solution is lost.

43 Project Retrospective

At the end of each Snapchat feature development or system upgrade project, we'll conduct a retrospective to analyze what worked well and what didn't, focusing on specific methods, tools, and strategies used during the project.

What We'll Document:

- **Effective Strategies:**
 - **Agile Sprint Planning:** The implementation of bi-weekly sprints and daily stand-up meetings helped keep the team aligned, ensuring that feature development like AR filters was on track and that blockers were addressed quickly.
 - **Cross-team Collaboration:** Using Slack and Jira to coordinate between engineering, design, and product teams led to faster feedback loops and reduced miscommunication when implementing changes to the user interface.

- **Challenges Faced:**
 - **Server Overload During Feature Testing:** During the rollout of a new feature, server capacity wasn't fully tested, leading to slow app performance and some crashes. This showed that server capacity testing needs to be prioritized in future projects before feature launches.
 - **Incomplete User Research:** A feature aimed at improving privacy settings didn't fully consider user experience feedback. This resulted in a design that was initially difficult for users to understand. We need to include more in-depth user testing earlier in the development cycle.

- **Recommendations for Future Projects:**

- **Increase Load Testing:** Prior to the launch of any feature with high user engagement potential (like new filters or Snapchat Plus), we need to allocate more time to stress test servers under anticipated load to avoid slowdowns and crashes.
- **Enhance User Research Phase:** In the next round of app redesigns, we should conduct early-stage usability testing with a wider sample of users to avoid rework and improve the accuracy of feature designs.

Why We Do This:

This retrospective process helps ensure that we're constantly refining our approach. Identifying specific improvements, such as improving server scalability testing or enhancing user experience research, will guide future projects, allowing us to avoid similar pitfalls and deliver high-quality features more efficiently. This approach supports our commitment to continuous improvement and keeps Snapchat's development cycles adaptable and responsive to user needs.

VI Glossary

Key phrases pertaining to Snapchat and its capabilities might be defined in a glossary in a paper on the app. This would ensure clarity for users who are not familiar with the platform's jargon, particularly if the material is meant for a wide audience, such as schoolchildren. Examples of terms that could be used are listed below:

- **Snap:** A snap is a picture or video that is shared on Snapchat and is usually only accessible to the receivers for a brief period of time.
- **Snapstreak:** A tool that keeps track of the days in which two users have exchanged Snaps, promoting further communication.
- **Story:** A set of Snaps that users may add to their profile and make public or friend-viewable for a full day.
- **Memories:** A private collection of saved Stories and Snaps that lets users view and distribute earlier work.
- **Lenses:** Augmented reality (AR) filters that can be used to add effects and animations to faces and the environment while taking pictures.
- **Bitmoji:** A user-created, customized cartoon avatar that may be used to represent oneself in profiles, messages, and Snaps.
- **Discover:** A place where visitors may see carefully chosen material from influencers, publishers, and celebrities.
- **Chat:** Snapchat's messaging software that lets users exchange movies, pictures, and text messages that vanish once they're viewed.

- **Friend Emoji:** Tiny icons that show the status or kind of friendship based on interaction patterns that show up next to friends' names in Snapchat.

VII References / Bibliography

- [1] E. Spiegel and I. Khan, *Snapchat and the Digital Life*, Second Edition, New York: Snap Inc. Publications, 2021.
- [2] A. Silberschatz, P. B. Galvin, and G. Gagne, *Operating System Concepts*, Ninth ed., Wiley, 2013.
- [3] J. Smith, *Social Media Interaction and User Engagement*, Third Edition, Pearson, 2018.
- [4] M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, Third Edition, Boston: Pearson Education, 2004.
- [5] Snapchat Inc., "Privacy by Design: How Snapchat Protects User Data," Snapchat Technical Documentation, 2020.
- [6] T. Luo, "User-Centric Design for Mobile Applications," *Journal of Social Media & Digital Interaction*, vol. 11, pp. 23-45, 2019.

Plagiarism Check

Project Description:

The screenshot shows the PapersOwl plagiarism checker interface. At the top, there's a navigation bar with links for Services, Writing Tools, How it Works, Support, About us, Log In, and an orange ORDER NOW button. On the left, there's a sidebar with social sharing icons (Facebook, Twitter, Pinterest, Email) and a share count of 44.5K. The main content area is titled "Free Online Plagiarism Checker". It displays a text sample about Snapchat, its word count (1173 words / 7278 characters), and two buttons for "Recheck this text after changes" and "Check another text". To the right, the similarity report is shown with "SIMILAR 0.0%" and "ORIGINAL 100.0%". A message says "Well done, your text is unique!". Below this, there's a promotional message for essay writing services.

This screenshot shows the same interface as the first one, but with a different text sample. The text discusses consistency in naming conventions across a context diagram and a data dictionary. It includes the word count (1054 words / 6156 characters) and the same "Recheck" and "Check another text" buttons. The similarity report shows 0.0% similarity and 100.0% originality, with the message "Well done, your text is unique!" and the same promotional essay writing message.

Requirements:

PapersOwl Log out My orders My balance \$0.00 Earn 35 ADD FUNDS PLACE ORDER Menu

Free Online Plagiarism Checker

44.5K Shares f X Pinterest Email

12 performance requirements 12a speed and latency requirements content response times defines a time limit for snapchats response to user interactions as an illustration sendingreceiving of snaps the duration of time required for a snap to be transmitted by a single user received by the server and shown to the intended recipient story loading the amount of time it takes for a user-tapped story to begin playing operating speed establishes how quickly snapchat should do different tasks

117 words (7359 characters) Recheck this text after changes Check another text

SIMILAR 5.2% ORIGINAL 94.8%

MAKE IT UNIQUE

Text matches these sources

Sources: 1. <https://blog.flutter.wtf/application-s...> 5.2% Exclude source View source

PapersOwl Log out My orders My balance \$0.00 Earn 35 ADD FUNDS PLACE ORDER Menu

Free Online Plagiarism Checker

4.5k Shares f X Pinterest Email

13 dependability requirements 13a reliability requirements content failure rate indicates the frequency of snapchat failures while maintaining user satisfaction and operational standards failure frequency in order to guarantee users continuous availability and performance snapchat should not encounter more than one major failure every day data integrity makes sure that even in the case of a malfunction user data is secure and unaltered fail-safe requirement no user data such as chat history stories or snaps shall be lost or corrupted in the event of a system failure mechanisms

1020 words (6727 characters) Recheck this text after changes Check another text

SIMILAR 5.5% ORIGINAL 94.5%

MAKE IT UNIQUE

Text matches these sources

Sources: 1. <https://www.atlassian.com/incident-...> 5.5% Exclude source View source

PapersOwl Log out My orders My balance \$0.00 Earn 35 ADD FUNDS PLACE ORDER Menu

Free Online Plagiarism Checker

44.5K Shares f X Pinterest Email

14b supportability requirements content support level indicates the kind and extent of support that snapchat needs as well as whether it should be integrated into the app or offered through a help desk motivation ensuring that the support requirements are well-defined and encompass all essential elements to properly maintain the application and aid users is known as adequate specification considerations support forms describe the types of assistance that snapchat will

1658 words (11003 characters) Recheck this text after changes Check another text

SIMILAR 0.0% ORIGINAL 100.0%

Well done, your text is unique!

Need an essay written but don't have the time? With PapersOwl you'll get it professionally researched, written and received right on time!

GET MY ESSAY DONE

Design:

The screenshot shows the PapersOwl plagiarism checker interface. At the top, there's a navigation bar with 'Log out', 'My orders', 'My balance \$0.00', 'Earn 35', 'ADD FUNDS', 'PLACE ORDER', and a 'Menu' icon. Below the navigation bar, the title 'Free Online Plagiarism Checker' is displayed. On the left, there's a sidebar with '44.5K Shares' and social sharing icons for Facebook, Twitter, Pinterest, and Email. The main content area contains a text editor with the following text:
21 system design 21a design goals content snapchat is a social app that's all about sharing quick real-time moments with friends. It's designed to feel playful and natural, encouraging people to capture their day as it happens. Photos, videos, and even texts shared through Snapchat disappear after being viewed, making interactions feel a little more personal and less permanent than on other platforms. It's packed with fun features like filters, bitmoji, and short stories that users can...
The text editor also shows '906 words (5628 characters)'. To the right, the similarity report shows 'SIMILAR 7.6%' and 'ORIGINAL 92.4%'. A large orange button says 'MAKE IT UNIQUE'. Below the report, it says 'Text matches these sources' and lists '1. https://clockwise.software/blog/top...' with a '7.6%' similarity score. Buttons for 'Exclude source' and 'View source' are also present.

This screenshot shows the same interface as the first one, but with a different text input. The sidebar still shows '14.5K Shares'. The main text input contains:
23 proposed software architecture snapshots new proposed design is all about making the app faster, smarter, and more personal for each user by breaking down features into smaller, separate components called microservices. Snapchat could easily update or scale parts of the app without affecting the whole system. This setup also allows for smarter recommendations, improved security, and a smoother experience during busy times. With this structure, Snapchat would rely on flexible...
The text editor shows '928 words (5876 characters)'. The similarity report on the right shows 'SIMILAR 0.0%' and 'ORIGINAL 100.0%'. A message below the report says 'Well done, your text is unique!'. At the bottom, there's a call-to-action button 'GET MY ESSAY DONE'.

Test Plans:

The screenshot shows the PapersOwl plagiarism checker interface. The top navigation bar includes 'Log out', 'My orders', 'My balance \$0.00', 'Earn 35', 'ADD FUNDS', 'PLACE ORDER', and 'Menu'. On the left, there's a sidebar with '44.5k Shares' and social sharing icons for Facebook, Twitter, Pinterest, and Email. The main content area is titled 'Free Online Plagiarism Checker'. It displays a text sample about test plans, a word count of 1444 words (9868 characters), and two buttons for 'Recheck this text after changes' and 'Check another text'. To the right, it shows similarity results: 'SIMILAR 0.0%' and 'ORIGINAL 100.0%'. A message says 'Well done, your text is unique!'. Below this, promotional text for essay writing services is visible.

This screenshot shows another instance of the PapersOwl plagiarism checker. The layout is identical to the first one, with the same top navigation bar, sidebar, and main content area. The text sample discusses approach content test cases, a word count of 2860 words (19065 characters), and the same set of similarity results ('SIMILAR 0.0%' and 'ORIGINAL 100.0%'). The 'Well done, your text is unique!' message and promotional text are also present.

Project issues:

The screenshot shows the PapersOwl plagiarism checker interface. The main heading is "Free Online Plagiarism Checker". On the left, there is a sidebar with social sharing icons (Facebook, Twitter, Pinterest, Email) and a share count of "44,5K Shares". The main content area displays a text sample about project issues, stating: "project issues 34 open issues content a list of factors that remain uncertain and could significantly impact the product if not addressed these are issues that have been identified but lack a clear resolution motivation to bring potential uncertainties into the open allowing for better visibility and a foundation for effective risk analysis which supports informed decision-making examples our assessment of the new processors compatibility with our application is still ongoing additionally". Below the text, it says "2017 words (12255 characters)". At the bottom of the content area are two buttons: "Recheck this text after changes" and "Check another text". To the right, the similarity report shows "SIMILAR 0.0%" and "ORIGINAL 100.0%". A green message says "Well done, your text is unique!". Below this, promotional text reads: "Need an essay written but don't have the time? With PapersOwl you'll get it professionally researched, written and received right on time!" and a button "GET MY ESSAY DONE".

This screenshot shows the same interface as the first one, but with a different text sample. The heading is "Free Online Plagiarism Checker". The sidebar shows "44,5K Shares". The main content area displays a text sample about good risk management: "good risk management means spotting potential issues early planning for them and keeping a close eye as the project progresses identifying snapchats key risks this section is where we outline the biggest and most likely risks for snapchats project drawing from risk assessment guides here are a few to keep an eye on data and metrics misinterpreted or inaccurate data can skew project decisions tracking gaps if we dont measure effectively its hard to see if were on track schedule pressure deadlines that are too tight can lead to stress and impact the quality of work". Below the text, it says "2128 words (13673 characters)". At the bottom of the content area are two buttons: "Recheck this text after changes" and "Check another text". To the right, the similarity report shows "SIMILAR 0.0%" and "ORIGINAL 100.0%". A green message says "Well done, your text is unique!". Below this, promotional text reads: "Need an essay written but don't have the time? With PapersOwl you'll get it professionally researched, written and received right on time!" and a button "GET MY ESSAY DONE".

This screenshot shows the same interface again, with a third text sample. The heading is "Free Online Plagiarism Checker". The sidebar shows "44,5K Shares". The main content area displays a text sample about the 41 waiting room: "41 waiting room sometimes great ideas dont make it into the next version of snapchat whether because of time constraints or complexity the waiting room is where we keep track of these future ideas ensuring they arent forgotten this section gathers potential improvements and innovative suggestions from users and teams that wont be in the current release but are kept as possibilities for later versions this way snapchat captures user and team creativity without overloading the". Below the text, it says "756 words (5029 characters)". At the bottom of the content area are two buttons: "Recheck this text after changes" and "Check another text". To the right, the similarity report shows "SIMILAR 0.0%" and "ORIGINAL 100.0%". A green message says "Well done, your text is unique!". Below this, promotional text reads: "Need an essay written but don't have the time? With PapersOwl you'll get it professionally researched, written and received right on time!" and a button "GET MY ESSAY DONE".

Glossary:

The screenshot shows the PapersOwl plagiarism checker interface. On the left, there's a sidebar with social sharing icons (Facebook, Twitter, Pinterest, Email) and a '44.5k Shares' counter. The main content area has a title 'Free Online Plagiarism Checker'. A text input box contains the following text:
vi glossary key phrases pertaining to snapchat and its capabilities might be defined in a glossary in a paper on the app this would ensure clarity for users who are not familiar with the platforms jargon particularly if the material is meant for a wide audience such as schoolchildren examples of terms that could be used are listed below snap a snap is a picture or video that is shared on snapchat and is usually only accessible to the receivers for a brief period of time snapstreak a tool
Below the text, it says '250 words (1436 characters)'. At the bottom of the input box are two buttons: 'Recheck this text after changes' and 'Check another text'. To the right, the similarity report shows 'SIMILAR 0.0%' and 'ORIGINAL 100.0%'. Below the report, a message says 'Well done, your text is unique!'. There are also promotional messages for essay writing and a button to 'GET MY ESSAY DONE'.

Reference/Bibliography:

This screenshot is identical to the one above, showing the same interface and results for a different piece of text. The text input box contains a bibliography entry:
vii references bibliography 1 e spiegel and i khan snapchat and the digital life second edition new york snap inc publications 2021 2 a silberschatz p b galvin and g gagne operating system concepts ninth ed wiley 2013 3 j smith social media interaction and user engagement third edition pearson 2018 4 m fowler uml distilled a brief guide to the standard object modeling language third edition boston person education 2004 5 snapchat inc privacy by design how snapchat protects user data
Below the text, it says '107 words (643 characters)'. At the bottom of the input box are two buttons: 'Recheck this text after changes' and 'Check another text'. To the right, the similarity report shows 'SIMILAR 0.0%' and 'ORIGINAL 100.0%'. Below the report, a message says 'Well done, your text is unique!'. There are also promotional messages for essay writing and a button to 'GET MY ESSAY DONE'.