Software Requirements Specification

for

Circlify

Version 1.0

Prepared by

Group Name: ***<place your group name here>***

Amartya Mishra	**60004210210**	**Amartyamishra2503@gmail.com**
Gaurang Bhogle	**60004210192**	**bgaurangan@gmail.com**
Darshit Sarda	**60004210208**	**darshitsarda10@gmail.com**
Shubham Mehta	**60004210191**	**shubhammehta2003@gmail.com**

Instructor:	***Dr. Kiran Bhowmick***
Course:	**Software Engineering**
Lab Section:	***Software Engineering Lab***
Teaching Assistant:	**NA**
Date:	**07-02-2024**

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

1 Introduction

1.1 Document Purpose

The Software Requirements Specification (SRS) document for Circlify serves as a foundational guide outlining the essential components and expectations for the development of the Circlify software product. It begins with an introduction providing context on Circlify's purpose and scope, followed by a delineation of its boundaries and included features within the scope section. Functional requirements are detailed, specifying the necessary capabilities and user interactions, while non-functional requirements outline quality attributes and constraints.
User interfaces and system architecture are described to illustrate how users will interact with the system and its underlying structure. Data requirements, assumptions, dependencies, and constraints are also documented, providing insights into Circlify's design considerations. Finally, acceptance criteria are defined to ensure stakeholder expectations are met. Overall, the SRS serves as a blueprint for development, facilitating communication between stakeholders and guiding the implementation process to ensure Circlify aligns with user needs and quality standards.

1.2 Product Scope

The Software Requirements Specification (SRS) for Circlify encapsulates the core features and expectations of a social media application designed for both individual and business users. Circlify enables seamless sharing of multimedia content such as audio, video, images, and files, fostering collaboration through posts, stories, and vertical videos. For businesses, Circlify offers a platform to promote ads and other targeted content to engage with their desired audiences effectively. The SRS outlines the application's scope, functionalities, user interfaces, system architecture, data requirements, and acceptance criteria. It serves as a blueprint for development, ensuring Circlify meets user needs and quality standards while facilitating communication between stakeholders and the development team.

1.3 Intended Audience and Document Overview

The Software Requirements Specification (SRS) document for Circlify is primarily directed towards the Circlify engineering and product team. Its secondary purpose is to align leadership and investors with the project vision and requirements. The detailed information provided within the SRS aims to provide the development team with a clear understanding of the expected functionality and characteristics of Circlify. This document serves as a comprehensive reference point throughout the development process, ensuring that all stakeholders are aligned and that the final product meets the defined requirements and expectations.

Following this introductory overview, Section 2 provides a high-level description of Circlify, its context, and primary components. Section 3 delves into the specific functional and behavioral requirements. Section 4 covers non-functional requirements around performance, security, and quality attributes. Finally, supporting appendices provide a data dictionary and project logs.

1.4 Definitions, Acronyms and Abbreviations

- API - Application Programming Interface
- SDK - Software Development Kit
- MVP - Minimum Viable Product
- SRS - Software Requirements Specification
- UI - User Interface

1.5 Document Conventions

In general this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1" margins found in this template. For Section and Subsection titles please follow the template.

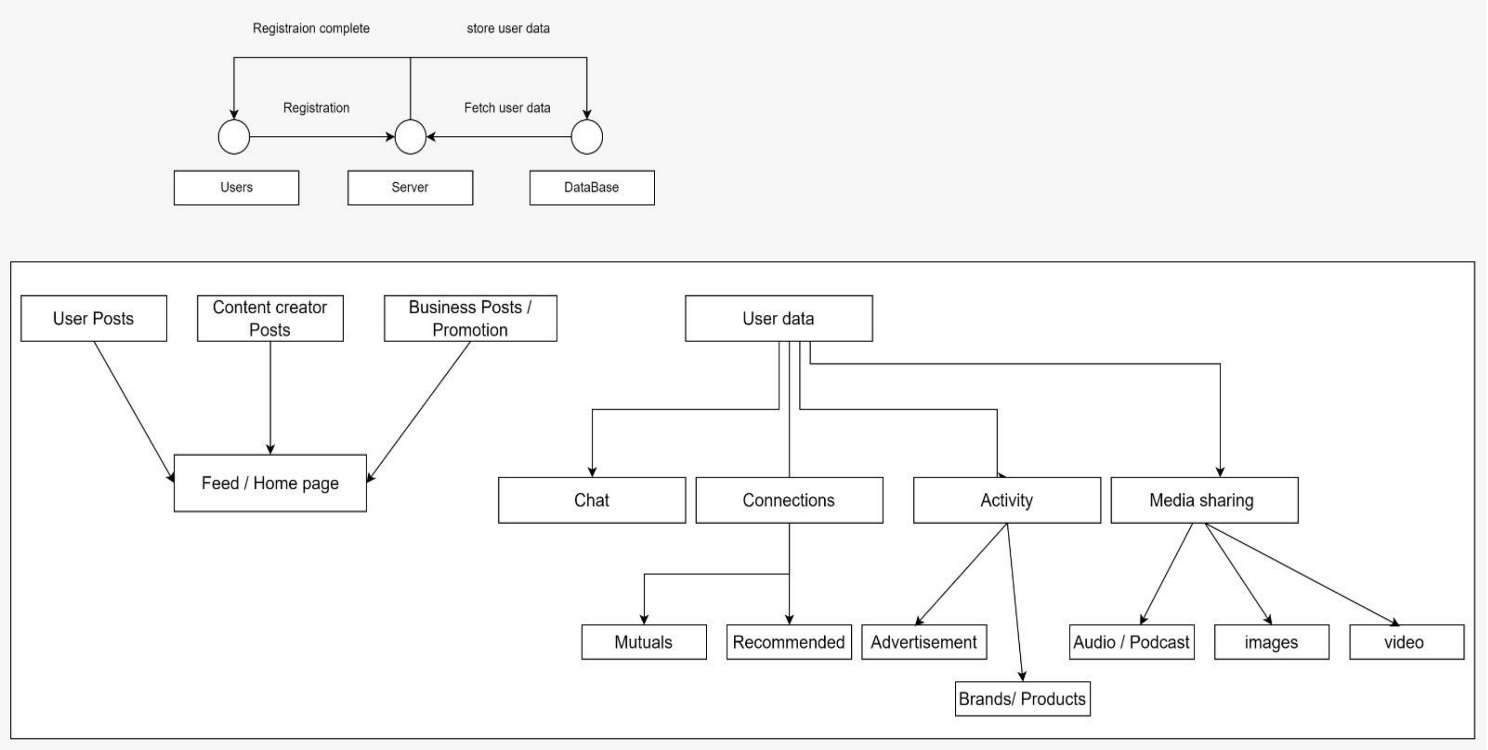
1.6 References and Acknowledgments

Our app Circlify takes inspiration from various different applications in the market like Instagram, Facebook, Whatsapp, etc.

2 Overall Description

2.1 Product Perspective

Circlify will be designed as a standalone social application, not requiring integration with existing platforms. Users will access the platform via a web-based client and native mobile apps on iOS and Android. The backend will consist of cloud-hosted servers providing API-based access to a relational database.



2.2 Product Functionality

- User profiles, feeds, and social connections - Allows users to create personal profiles, connect with others, and view content feeds

- Sharing of posts with rich text, images, videos, and files - Enables users to share diverse post content including text, visual media, videos, and documents

- Ephemeral stories and vertical video sharing - Offers Snapchat-like ephemeral stories and TikTok-style vertical videos

- Multimedia upload and optimization - Provides uploading and automatic optimization of images, videos and other media

- Hashtagging and social discovery - Implements hashtags and recommendations for discovering content and like-minded users

- Business tools for promotions and analytics - Gives businesses abilities to run targeted promotions and view engagement analytics

- Configurable privacy controls and content moderation - Offers users granular controls over content visibility and moderates inappropriate content

2.3 Users and Characteristics

- Individual users - Share personal experiences and connect with friends
- Business users - Share promotions, ads, reach target audiences; carry out business research on set target audiences.
- Age groups 13 and above – comprising of teenage and young adults majoritarily.
- Technically apt and novice users

Primary users are individuals looking to share personal content and experiences.

2.4 Operating Environment

The Circlify application is being developed as a cross-platform mobile app for Android and iOS devices. It will target a minimum SDK version of 21 for Android and iOS 13 for Apple devices to maximize accessibility across a range of hardware. Considering the goal of broad user adoption, the app is designed to function smoothly on devices with as low as 2GB of RAM. Along with native mobile apps, users can also access Circlify via any modern web browser for platform flexibility.

On the backend, Circlify will employ a hybrid data persistence architecture. A relational SQL database will efficiently handle structured user profile information and metadata. A NoSQL document store will provide the flexibility to capture dynamic social content like posts and media. Security is paramount for a social platform, so rigorous policies will be enforced for access control, data encryption, and user authentication using industry best practices like OAuth and multi-factor mechanisms. The backend will provide a robust API layer to securely expose functionality to clients. The overall technical approach aims to deliver an accessible, secure, and reliable experience across use cases.

2.5 Design and Implementation Constraints

Given the time and resource constraints of the initial development timeline, the product launch will strategically focus on implementing the core functionality and most critical user needs first. The engineering team will need to make careful design trade-offs and prioritize the MVP feature set guided by technical limitations. For example, while an advanced AI recommendation system is desired, it may be deprioritized for the first release in favor of essential capabilities like content sharing and user profiles. The cloud infrastructure will be scaled to efficiently support the target initial user base. Advanced monitoring and auto-scaling capabilities can be added later to manage growth. Where possible, the team will leverage existing open source libraries and reusable components to accelerate development. With agile methodology, the launch feature set will provide the essential user value while allowing for continuous enhancement and expansion in subsequent releases as future scope.

The other primary limitations are related to security of users and business personals . There will be a need of developing an adaptable security system that can adopt to age restricted content being circulated via the platform. There will bee need to verify the businesses that can actually promote and carry out business on the platform.

2.6 User Documentation:

The application's user documentation strategy encompasses various support channels, including in-app tooltips, FAQs, a help center, social community support, and a knowledge base on the website. In-app tooltips offer real-time guidance, FAQs address common queries, while the help center provides comprehensive resources. Social community support fosters engagement, and the knowledge base hosts detailed articles and guides. This multifaceted approach ensures users have diverse resources at their disposal, facilitating a seamless and productive user experience.

2.7 Assumptions and Dependencies

Key assumptions include:

- Projected user growth targets are accurate
- Core features can be built within budget and timeline
- Mobile app stores and ad networks will approve the application
- Target devices and browsers retain significant market share
- Users will be willing to adopt a new platform if it provides unique value
- Target demographics will respond positively to differentiating features

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface shall consist of three different GUI screens. They are Home Page, Explore Tab and Chat Tab.

3.1.1 1. Home Page

- Conversation List and Chat Interface:
  - Users access ongoing conversations displayed with avatars or profile pictures and usernames.
  - Tapping opens a threaded chat interface for message viewing and replies.
- Messaging Features:
  - Text and media sharing, including photos and videos.
  - Emojis, stickers, voice messages, reactions, and likes enhance communication.
- Notifications and Privacy:
  - Notifications keep users updated on new messages.
  - Users control messaging privacy settings, managing who can message them and see their online status.

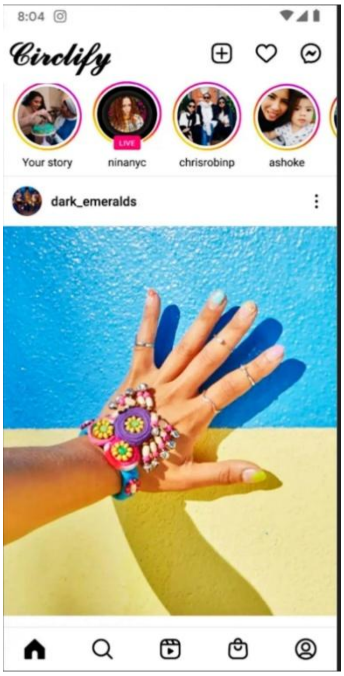
3.1.1.2 Explore/Search Tab

Personalized Content: Shows posts, stories, and updates from followed accounts.
Scrollable Interface: Allows vertical scrolling for continuous content viewing.
Engagement Features: Enables liking, commenting, and sharing directly from the feed.
Algorithmic Sorting: Prioritizes content based on user interests and engagement.
Search Page:

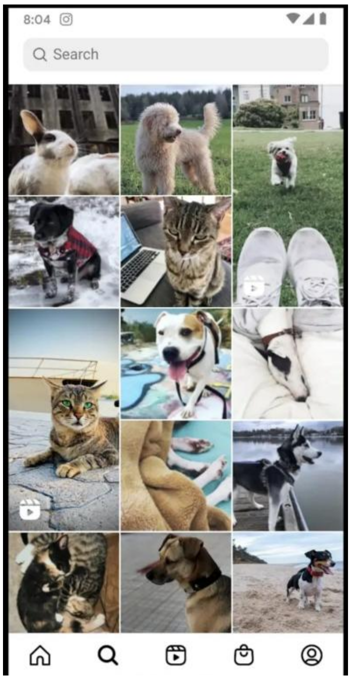
Content Discovery: Helps users find new accounts, posts, and relevant content.
Search Bar: Allows input of keywords, hashtags, or account names for specific searches.
Explore Tab: Highlights trending and popular content.
Filters and Categories: Offers filter options and content categories for browsing.
Suggested Accounts: Recommends accounts based on user interests and engagement.

3.1.1.3 Chat Tab

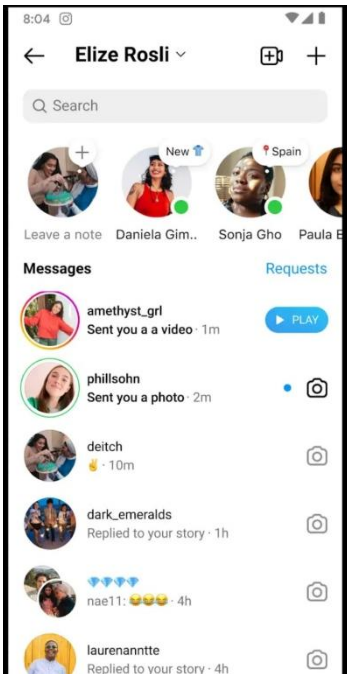
The Chat/Share tab serves as a dynamic hub for real-time text conversations and multimedia sharing. Users can exchange images, videos, and documents, utilize emojis and stickers for expression, engage in group discussions, search through chat history, manage privacy settings, and receive timely notifications, fostering seamless and personalized communication experiences.



3.1.1.1 Home Page



3.1.1.2 Search Tab



amethyst_grl
Sent you a a video · 1m

phillsohn
Sent you a photo · 2m

deitch
✌ · 10m

dark_emeralds
Replied to your story · 1h

💎💎💎💎
nae11: 😂😂😂 · 4h

laurenanntte
Replied to your story · 4h

3.1.1.2 Chat Tab

3.1.2 Hardware Interfaces

3.1.2.1 Touchscreen Interface: On mobile devices, such as smartphones and tablets, users interact with the Circlify app through touchscreens. They can navigate through menus, play/pause songs, and manage playlists using touch gestures.

3.1.2.2 Buttons and Controls: On various devices, including smartphones, tablets, and smartwatches, physical buttons and controls (e.g., play, pause, skip, and volume buttons) are used to manage playback without directly interacting with the touchscreen.

3.1.3 Software Interfaces

Operating System APIs

- The Circlify app interacts with operating system APIs on iOS and Android to perform platform-specific tasks like accessing the camera, storing data, and displaying notifications.

Web APIs

- Circlify provides Web APIs that enable third-party applications to integrate with the platform. This includes APIs for user authentication, accessing public profile data, and posting content.

Database APIs

- The app uses database APIs to interact with the SQL and NoSQL databases used on the backend. This includes CRUD operations for objects like users, posts, media, and comments.

Storage APIs

- Storage APIs are used to interface with cloud storage providers for managing user-generated media like images, videos, and files.

Messaging APIs

- Messaging APIs facilitate real-time communication by allowing the exchange of text, media, video calling, and other conversational content between users.

Notification APIs

- Notification APIs on mobile operating systems and web are leveraged to enable push notifications to user devices.

3.1.4 Communications Interfaces

Circlify will employ industry standard protocols and technologies for secure and reliable communication between clients and servers. Data transmission between Circlify apps and API servers will use HTTPS with the latest TLS encryption. Real-time messaging will leverage WebSocket or Socket.io protocols with optimized message serialization formats like Protobuf. For push notifications to devices, platform-specific services like Firebase Cloud Messaging or Apple Push Notification Service will be used.

Where applicable, common lightweight data exchange formats like JSON will be used to maximize interoperability and efficiency. Security measures like encryptions, access controls, and input validation will ensure integrity and privacy of communications. Performance will be optimized through connection pooling, caching, and other best practices.

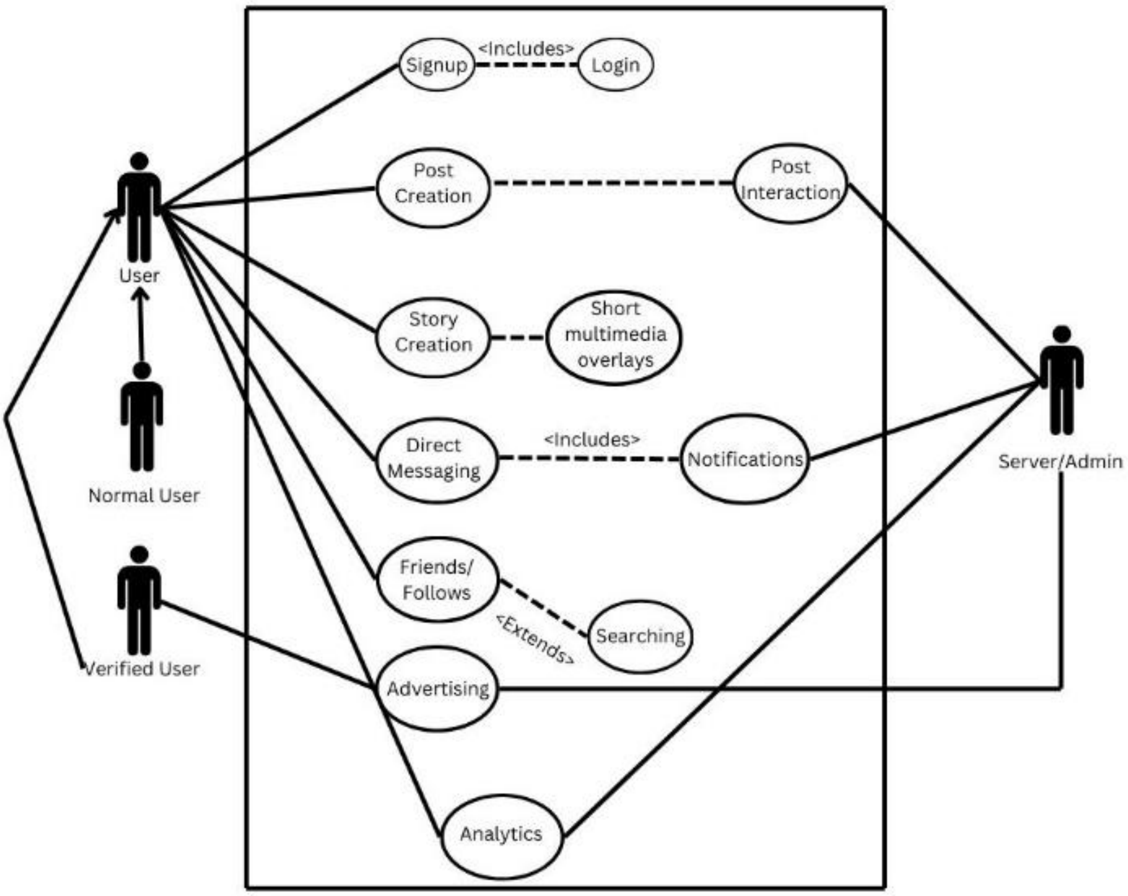
3.2 Functional Requirements

- User profiles, feeds, and social connections - Allows users to create personal profiles with photos, bios, and interests to represent themselves on the platform and connect with other users they can follow in customizable feeds.
- Sharing of posts with rich text, images, videos, and files - Enables users to make posts that contain formatted text, embedded images, uploaded videos, and attached document files for diverse multi-media sharing.
- Ephemeral stories and vertical video sharing - Offers ephemeral 24-hour stories combining multiple photos/videos with overlays and stickers as well as standalone vertical videos popularized by TikTok.
- Multimedia upload and optimization - Provides functionality to upload images, videos and files while automatically optimizing media for faster loading and streaming on the platform.
- Hashtagging and social discovery - Implements hashtags to tag topics and interests for discovering related content as well as user and content recommendations based on relevance to interests.

- Business tools for promotions and analytics - Gives business accounts abilities to create targeted promotions to relevant users and view analytics on reach, impressions, and engagement.
- Configurable privacy controls and content moderation - Offers users granular controls to set visibility of posts/profile and provides content flagging along with moderation teams and processes to review inappropriate content.

3.3 Behaviour Requirements

3.3.1 Use Case View



4 Other Non-functional Requirements

4.1 Performance Requirements

- Home feed and search results should load within 2 seconds for good UX
- Media uploads and compressed/optimized within 5 seconds
- Peak load of 100 concurrent users per server
- Push notifications delivered within 3 seconds
- 99.95% uptime for API and database servers
- Automated performance monitoring
- Media processing auto-scaled to maintain performance

4.2 Safety and Security Requirements

- Personally identifiable data encrypted at rest and in transit
- OAuth used for secure third party authentication
- Input validation to prevent code injection vulnerabilities
- Multi-factor authentication required for business accounts
- AI filters to detect inappropriate text, media, hashtags
- Content moderation team reviews flagged posts within 12 hours
- Age gating blocks users under 13 years' old
- Reporting mechanisms for objectionable users/content

4.3 Software Quality Attributes

Circlify will aim to achieve high quality standards guided by specific attributes:

Usability

- The user interfaces will be clean, intuitive, and easy to navigate.
- Onboarding flows will guide new users.
- Navigation and actions will be consistent across platforms.
- Accessibility standards will be followed to support assistive technologies.

To meet usability goals, user research studies and usability testing will be conducted. Feedback will drive iterative design improvements.

Reliability

- Infrastructure will implement redundancy and failover mechanisms.
- Input validation will catch bad data.
- Request throttling will prevent overload.
- Monitoring will enable issue diagnosis and rapid response.
- Disaster recovery provisions and data backup will minimize risk.

Reliability will be measured through uptime, mean time between failures, and ability to recover quickly.

Scalability

- Auto-scaling groups will adjust capacity based on demand.
- Database sharing will distribute load at high volumes.
- Caching and CDNs will reduce server processing.
- Code will be optimized to use resources efficiently.
- Modular services can scale independently.

Scalability will be evaluated under load tests simulating growth.

Maintainability

- Code will adhere to style standards with meaningful naming, comments.
- Version control will track changes and enable rollback.
- Code modularity will isolate components.
- Automated testing will catch regressions quickly.
- Monitoring will provide application insights.
- Developer documentation will capture institutional knowledge.

Maintainability will be measured by change failure rate and mean time to repair.

Academic Year: 2023_24

Appendix B - Group Log

Date	Actors	Work Done
07/02/2024	Amartya Mishra	Analyzed Requirements
07/02/2024	Darshit Sarda	Analyzed Requirements
07/02/2024	Gaurang Bhogle	Analyzed Requirements
07/02/2024	Shubham Mehta	Prepared SRS
14/02/2024	Amartya Mishra	Prepared SRS
14/02/2024	Darshit Sarda	Prepared SRS
14/02/2024	Gaurang Bhogle	Prepared SRS
14/02/2024	Shubham Mehta	Prepared SRS
21/02/2024	Amartya Mishra	Prepared SRS
21/02/2024	Darshit Sarda	Prepared SRS
21/02/2024	Gaurang Bhogle	Prepared SRS
21/02/2024	Shubham Mehta	Prepared SRS