



STREAMING INTERTAINMENT XTENSION

CSCE5430 – Software Engineering Project

Fall 2023 – Section 009 - University of North Texas

Professor: Lewis Sykalski

Team Members:

- 1) Ashoka Chakravarthy**
- 2) Bhavya Mammai**
- 3) Haritha Buthukuri**
- 4) Bhavani Navari**

1. Background	3
Software Overview.....	4
Organization / Environment Overview	6
Schedule / Effort Overview.....	8
Management Overview	8
2. Program Plan	9
Work Breakdown Structure (WBS).....	9
Organizational Breakdown Structure (OBS).....	12
Basis of Estimate	13
Schedule.....	19
Software Process Model	20
3. Problem Definition	21
Problem Statement.....	22
Root Causes.....	22
Stakeholders & Users.....	23
System Boundary Diagram.....	24
Constraints.....	25
4. Frame Diagram	27
5. Context Diagram.....	27
6. Entity Relationship Diagram.....	29
7. Use Cases.....	29
Use Case Diagrams.....	30
Use Cases	30
Misuse Case Diagram	36
8. Activity Diagram	39
Normal Scenario.....	40
Abnormal Scenario.....	41
9. Sequence Diagram	42
Normal Scenario	42

	3
Abnormal Scenario.....	43
10. Data Flow Diagram	43
11. Functional Requirements.....	44
12. Non-Functional Requirements	47
13. Traceability Tables.....	49
14. User Interface Design.....	
15. Class Diagram.....	
16. Quality Attribute Scenarios/Tactics.....	
17. CM Overview and Metrics Strategy.....	
18. A System Level Test Plan.....	
19. Glossary.....	
20. Bibliography.....	

1. Background

A high-tech project called SIX seeks to improve the intelligence of your standard TV. It resembles a unique device (like a USB stick) that you can use Wi-Fi to connect to your TV. We may use SIX to stream content from Netflix or Amazon Prime on your TV. To use it, we must register our device and create an account. Your device can be controlled via a website or directly from the device. We may update our gadget, connect it to your home Wi-Fi, look up movies and actors using a database like IMDb, and sign up for various streaming services. We may view and manage our subscriptions online or through SIX, where we can even cancel them. The SIX-partnering streaming providers will create a unique app for it while adhering to some guidelines laid down by the project's developers. Additionally, we'll require a special remote control that makes it simple to pause, play, rewind, and browse menus. The SIX-partnering streaming providers will create a unique app for it while adhering to some guidelines laid down by the

project's developers. Additionally, you'll require a special remote control that makes it simple to pause, play, rewind, and browse menus.

The basis of the SIX project involves the development of cutting-edge streaming technology in order to succeed in the fiercely competitive streaming services market. By offering an accessory (HDMI stick) that can be plugged into existing TVs to enable Wi-Fi streaming services, the initiative aims to provide cutting-edge streaming technology. Thanks to this technology, users have additional options for accessing streaming content. The company's efforts to gain traction in the streaming market led to SIX. This is the sixth and final effort due to the significant stakes at risk and the company's commitment to success. The project's main target market is Japan, where it will be put to the test. This choice was based on a strategic choice to test the product locally before thinking about going global. Users can create accounts to register their devices and manage their accounts online. Users can use IMDB, which provides detailed content information, to look for actors and films, perhaps. Through your platform, users can subscribe to a wide range of content providers who adhere to the API standards. Users can view, modify, or cancel their subscriptions through the SIX platform, among other subscription management options.

Software Overview

The SIX system was created because there is a pressing need to transform how people receive streaming information on their TVs. The SIX system's desire to reshape the streaming entertainment industry served as the driving force behind its development. By providing an innovative HDMI stick gadget, SIX intends to enable consumers to quickly turn their regular TVs into smart, Wi-Fi-enabled hubs for gaining access to a wide variety of streaming material. By offering viewers a seamless, practical, and enhanced viewing experience, this innovation aims to close the gap between conventional television and the rapidly developing world of digital entertainment.

What are the software's objectives?

The main objective of the SIX software is to make it possible for users to access Wi-Fi-enabled streaming services on their TVs using an HDMI stick. For content discovery, subscription management, and playback control, it attempts to offer a seamless and approachable interface.

Who will use it? And where/how?

The SIX software is used by people who have compatible TVs and want to access streaming content. When using it, they will connect the HDMI stick to their TVs and communicate with the system via either the device or an online website.

Why is the system being developed?

The SIX system is being created in order to meet the changing needs of the digital entertainment sector. It seeks to give people a cutting-edge method of watching streaming material on their current TVs, increasing ease and choice in the digital entertainment industry. Additionally, it is the firm's sixth and final effort to make a significant impact on the cutthroat streaming business.

What are the purposes of the software?

The main objective of the SIX software is to make Wi-Fi enabled streaming services through the HDMI stick device as seamless as possible. It offers a user-friendly interface and allows users to register their devices, manage subscriptions, find content, and manage playback.

What factors are likely to influence the economics of product development, choice of technology, etc.?

Costs associated with manufacturing hardware, those associated with creating software, collaborations with content providers, and pricing strategies are all factors that affect the economics of product development. Security needs, scalability, and compatibility with different TV models are a few examples of variables that will affect the technology chosen.

What platform will it run on?

Two main platforms will be used by the SIX software: the HDMI stick gadget itself, which customers put into their TVs, and an internet website accessed via web browsers. Users' accessibility and convenience are ensured by this dual-platform strategy.

Are there unique quality attributes (e.g., reliability) considerations to consider?

The SIX system places a high priority on reliability to guarantee less downtime and constant playback quality. Due to the processing of user data and subscriptions, data security and privacy are also crucial factors. Usability and user happiness must come first in the system's user interface.

How novel is this application? Are we breaking new ground?

Even though the idea of streaming services is not new, SIX adds innovative components. By supplying both a specialized remote control with distinctive features and a dedicated hardware device (the HDMI stick) to enable streaming, it sets new standards. It will soon have features like voice activation and recommendations for personalized content that will further set it apart from competitors.

Organization / Environment Overview

The organizational and environmental factors that affect the program must be thoroughly understood in the context of the SIX project.

Who is responsible?

The major team members of the SIX project share responsibility, with each having distinct roles and responsibilities.

- Architects
- Network Communication Team
- Database administrators

- UI/UX designers
- Software engineers
- Hardware engineers
- Business Analyst
- Marketing/Sales Team
- QA Team

While hardware engineers work on the HDMI stick, software engineers concentrate on software development. A user-friendly interface is guaranteed by UI/UX designers. The system's general design is overseen by architects. Business Analyst are gathering all the requirements of the project. The Marketing Team is very important for our project because they are the one who operates all the sales and marketing strategies.

Where are they organizationally located?

Some team members may work directly in the corporate or regional offices of the organization, while others may be dispersed across many locations. There may be different organizational structures for different external stakeholders, such as content producers. Having distinct roles and duties, the client and end users are also significant stakeholders. Involved parties who create and maintain their apps in accordance with SIX's API specifications are external content providers. Customers are the end customers who will utilize the SIX system and profit from its features.

Who are your developers? Where are they located? What's their experience level? Are they multi-site? What's their work week?

In our team the project will be developed by **Bhavya Mammai**. She will handle all the back-end operations. She will work from home because most of the teams do it remotely. She has almost four years of experience. She will work from Monday to Friday in business hours.

Who is responsible for building and testing, operating and maintaining the system?

In our team, **Haritha Butukuri & Bhavani Navari** will do all the responsibility for designing and testing both the hardware and software components. Possibly outside content suppliers who make use of the SIX marketplaces may work together to operate and maintain the system.

Who is responsible for system development and operation?

System development operation will be handled by **Ashoka Chakravarthy**. Developing the infrastructure, hardware, and software required to support the SIX system. Some parts of operation, while content providers oversee the applications housed on the SIX marketplaces.

What are the roles and responsibilities of the main groups of people working on system development (e.g., architects, engineers)?

Architects: Architects are in charge of creating the overall system architecture and making sure it complies with user needs and business objectives.

Software engineers are responsible for creating the user interface, streaming capabilities, and backend services for the SIX system's software.

Database administrators are responsible for controlling the storage and retrieval of user and content data.

Schedule / Effort Overview

In the "SIX - STREAMING ENTERTAINMENT XTENSION" project, cutting-edge technology will be created that will enable customers to plug a device into their TV and access Wi-Fi-enabled streaming services. The project is scheduled to be launched in 18 months after being piloted in Japan. The endeavor is the business's sixth and final effort to succeed in this sector.

The User Interface, Database Backend, Network Communication, Sound, and Marketing & Sales are just a few of the components listed in the hierarchically organized Work Breakdown Structure (WBS) for this project. Every task has been delegated to accountable people or teams, and projected expenses have been given. The Wideband Delphi and COCOMO II estimation

methodologies, which take into consideration elements like requirements analysis, design, coding, unit testing, and other project-specific aspects, were used to create the estimates.

The WBS provides a systematic method for managing and carrying out the project and acts as the basis for project planning and resource allocation. It aids in defining the project's scope, breaking it down into manageable parts, and allocating duties. This WBS can be used by the project manager to monitor development, manage resources, and make sure the project stays on track and within budget.

Management Overview

A distinct technical and management strategy must be established in order to accomplish the objectives of the SIX project. This includes describing how the project will be managed to ensure the project's success as well as how the technology will be developed. The program plan section will go into more detail on these elements.

The development and integration/testing phases will call for a variety of software development environments and technologies. These tools will help with UI design, database backend development, network connection coding and testing, and sound functionality. The program plan will include specifics on the tools and technology that will be employed.

Technical hazards are inherent to any project utilizing cutting-edge technology, such as SIX. These difficulties can include managing the complexity of sound transmission, assuring compatibility with a wide range of TV models, resolving network security and performance issues, and integrating with different streaming service providers. The program strategy will include a thorough risk assessment to effectively identify and reduce these dangers.

Please refer to the program plan section for further information on the technical and administrative strategies, tools, and risk mitigation plans for the SIX project. This part will go into greater detail on these topics.

2. Program Plan

Work Breakdown Structure (WBS)

The hierarchical division of a project into smaller, easier-to-manage components or work packages is known as a work breakdown structure (WBS). It is a visual and organized tool for project management that helps define the scope of a project, arrange its components into manageable tasks, and improve planning, tracking, and control.

Work Breakdown Structure Table

Provide basic information about the project including: Project Title – The proper name used to identify this project; Project Manager - the person with responsibility for the successful delivery of the project to time cost and quality. Prepared by – The person(s) preparing the Date/Control Number – The date the plan is finalized and the change or configuration item control number assigned.

Project Title: SIX – STREAMING ENTERTAINMENT XTENSION **Prepared by:** BHAVYA MAMMAI
Project Manager: ASHOKA CHAKRAVARTHY SANJAPU **Date / Control Number:** 09/07/2023

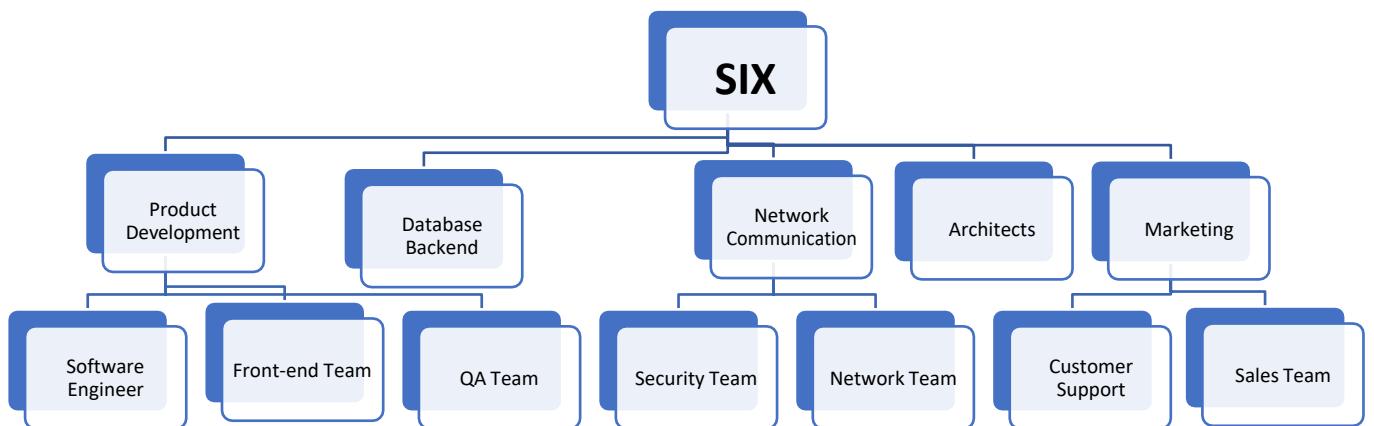
Element Number	WBS Elements Activity, Task, or Sub-Task Name	Definition of Activity or Task (Description)	Responsible Person or Group	Estimated Cost (Crore budget)
The unique reference ID for the activity or task.	Enter the name or title of the Task, Sub-task, Activity or Deliverable.	Provide a brief description of this Activity, Task or Sub-Task.	Enter the person or group(s) who are responsible	Enter the estimate of time and add a cross reference
1.0	All Software	Description	From OBS	\$20000
1.1	User Interface			\$4000
1.1.1	Requirements Analysis			\$1500
1.1.1.1	Task 1	Gather User Interface Requirements	UI Designers	
1.1.1.2	Task 2	Analyze user interface usability	UI Designers	
1.1.1.3	Task 3	Define UI Design guidelines	UI Designers	
1.1.2	Design			\$1500
1.1.2.1	Task 1	Create UI Design mockup	UI Designers	
1.1.2.2	Task 2	Prototype user interfaces	UI Designers	
1.1.2.3	Task 3	Review UI designs	UI Designers	
1.1.3	Code & Unit Test			\$1000
1.1.3.1	Task 1	Implement UI code based on designs	Development team	
1.1.3.2	Task 2	Perform unit testing of UI components	QA team	
1.1.3.3	Task 3	Resolve the UI related defects	Development team	
1.2	Database Backend			\$4000
1.2.1	Requirements			\$1500

	Analysis			
1.2.1.1	Task 1	Gather database requirements	Database team	
1.2.1.2	Task 2	Defining the database schema and structure	Database team	
1.2.2	Design			\$1000
1.2.2.1	Task 1	Create database design	Database team	
1.2.2.2	Task 2	Optimizing the performance of database	Database team	
1.2.3	Code & Unit Test			\$1500
1.2.3.1	Task 1	Implement logic	Development team	
1.2.3.2	Task 2	Perform unit testing of database	QA team	
1.3	Network Communication			\$3800
1.3.1	Requirements Analysis			\$1000
1.3.1.1	Task 1	Define network communication requirements	Business Analyst	
1.3.1.2	Task 2	Plan network architecture	Network team	
1.3.1.3	Task 3	Address security and encryption	Security team	
1.3.2	Design			\$800
1.3.2.1	Task 1	Design network protocols	Network team	
1.3.2.2	Task 2	Implement Security measures	Security team	
1.3.2.3	Task 3	Review the protocols and measures	Security team	
1.3.3	Code & Unit Test			\$2000
1.3.3.1	Task 1	Develop network communication code	Development team	
1.3.3.2	Task 2	Test network communication functionality	QA team	
1.3.3.3	Task 3	Resolve the defects		
1.4	Sound (P4)			\$3000
1.4.1	Requirements Analysis			\$750
1.4.1.1	Task 1	Gather sound requirements	Sound team	
1.4.2	Design			\$750
1.4.2.1	Task 1	Design sound requirements	Sound team	
1.4.3	Code & Unit Test			\$750
1.4.3.1	Task 1	Implement the sound code and test sound functionality	QA team	
1.5	Marketing & Sales			\$ 5200
1.5.1	Requirement Analysis			\$2000
1.5.1.1	Task 1	Gathering the products to market and sell	Business Analyst	
1.5.2	Design			\$1200
1.5.2.1	Task 1	Planning the various strategies to market the products	Marketing team	
1.5.3	Testing			\$2000
1.5.3.1	Task 1	Analyzing the specific markets	Sales team	

Organizational Breakdown Structure (OBS)

A hierarchical representation of an organization's structure called an organizational breakdown structure (OBS) aids in defining and illuminating the relationships and duties of various organizational units, departments, teams, or individuals within a business. It is a management technique used to make the reporting and communication channels inside an organization more clear, making it simpler to comprehend how the various organizational components work together and who is in charge of what.

The OBS for the project Streaming Intertainment Xtension is given below. This broadly includes development and marketing. Which further includes the development of UI, backend, and testing activities. We assigned teams for them. Marketing & sales will have their role after product development.



Basis of Estimate

Wideband Delphi :

In order to estimate the amount of time, effort, or other project-related factors, software developers and project managers utilize the Wideband Delphi Estimation Method, which is a structured technique. It is a development of the original Delphi technique, which was created for forecasting and decision-making in the middle of the 20th century. When handling complicated projects and including a group of specialists or stakeholders in the estimation process, the Wideband Delphi approach is especially helpful. The calculation is based on estimates of SLOC provided by team members.

Here Item 1 – User Interface

Item 2 – Database Backend

Item 3 – Network and communication

Item 4 – Sound

Item 5 – Marketing and Sales

Below is the wideband Delphi estimate for the project.

GR&A				Round 1						Round 2					
Bhavya	Bhavani	Haritha		Bhavya	Bhavani	Haritha	Average	STD DEV	CV	Bhavya	Bhavani	Haritha	Average	STD DEV	CV
This task uses a well-known standard	Need for specialized interface	This task uses a specialized interface	Item 1	10	20	15	15	5	0.333333	20	15	19	18	2.645751	0.146968
This task will require more data to analyze	No understanding	This task requires more relational attributes	Item 2	70	2	28	33.3	32.2	1.137248	81	15	15	17	3.464102	0.203771
No Understanding	Need for different strategy	Must use various security protocols	Item 3	15	15	12	14	14.14	0.123718	15	15	15	15	5	0.3333
Uses well-known standard	Uses well-known standard	No Understanding	Item 4	11	10	26	15.6	6.03	2.237	18	24	18	20	2.4	0.134
This task will require a marketing strategy	No understanding	Must use sales strategies	Item 5	30	6	8	14.6	5.12	2.112	40	10	13	21	3.126	0.19863

COCOMO II :

Barry Boehm created the software cost estimation method known as COCOMO II (COnstructive COst MOdel II). It incorporates three sub models for different stages of the software development life cycle: the Application Composition Model, Early Design Model, and Post-Architecture Model. COCOMO II provides estimates for software development effort, duration, and cost based on a variety of variables, including product, platform, staff, and project-specific features. It provides a methodical approach to resource allocation and project planning, assisting with budgeting and decision-making throughout the software development process.

Below is the COCOMO II estimation for the project.

BOE ID: WBS: ALL			
Author Name:	Level N/A	Period of Performance: 09/07/2023 to 11/29/2023	
Ashoka Chakravarthy	High level	Total Hours	2500
Bhavya Mammai			
Haritha Buthukuri			
Bhavani Navari			

Task Descriptions:

- 1.1.1 – Requirement Analysis of User Interface
- 1.1.2 – Design of User Interface
- 1.1.3 – Unit testing of User Interface
- 1.2.1 – Requirement Analysis of Database Backend
- 1.2.2 – Design of Database Backend
- 1.2.3 – Unit testing of Database Backend
- 1.3.1 – Requirement Analysis of Network & Communication
- 1.3.2 – Design of Network & Communication
- 1.3.3 – Testing of Network and Communication
- 1.4.1 – Requirement Analysis of Sound

1.4.2 – Design of Sound code

1.4.3 – Testing of Sound code

1.5.1 – Requirement Analysis of Marketing & Sales

1.5.2 – Design of Marketing & Sales

1.5.3 – Testing of Marketing & Sales

Basis of Estimate:

Labor Estimate:

WBS	Task Description	Calculation	Period	Hours
				2420
1.1.1	Requirement Analysis of User Interface	COCOMO II 10 Hours per day	11 days	110
1.1.2	Design of User Interface	COCOMO II	23 days	230
1.1.3	Unit testing of User Interface	COCOMO II	17 days	170
1.2.1	Requirement Analysis of Database Backend	COCOMO II	12 days	120
1.2.2	Design of Database Backend	COCOMO II	21 days	210
1.2.3	Unit testing of Database Backend	COCOMO II	26 days	260
1.3.1	Requirement Analysis of Network & Communication	COCOMO II	13 days	130
1.3.2	Design of Network & Communication	COCOMO II	18 days	180
1.3.3	Testing of Network and Communication	COCOMO II	17 days	170
1.4.1	Requirement Analysis of Sound	COCOMO II	15 days	150
1.4.2	Design of Sound code	COCOMO II	17 days	170
1.4.3	Testing of Sound code	COCOMO II	15 days	150
1.5.1	Requirement Analysis of Marketing & Sales	COCOMO II	10 days	100

WBS	Task Description	Calculation	Period	Hours
1.5.2	Design of Marketing & Sales	COCOMO II	12 days	120
1.5.3	Testing of Marketing & Sales	COCOMO II	15 days	150

Material Estimate:

WBS	Purchase Description	Calculation	Cost
			\$2 million
1.1.1	16 developer Boards	10 developers*\$800	\$8000
1.3.1	12 Microchips	12*\$20k= \$240k	\$240000
1.4.1	7 smart appliances (for sound testing)	7*\$10k = \$70k	\$70000
1.5.1	Marketing Purchases	20 Ad Campaigns* \$85k/campaign= \$1.7 million	\$1700000

Trips Estimate:

WBS	Trip Description	Calculation	Cost
			\$67500
1.5.2	Marketing planning	5 media consulting trips*3 members*\$1500 /week	\$22500
1.5.3	Trips to field test location	10 weeks*3 test team members*\$1500/week	\$45000

Rationale/Estimating Methodology:

A popular model for calculating the effort and duration of software development is COCOMO II.

It depends on several factors, such as the size of the software, the difficulty of the project, and the development team's experience.

Reasons for Using an Estimating Methodology: it appears that the COCOMO II model calculates the time and effort needed to complete various project phases, particularly those involving the

user interface, database backend, network and communication, sound, and marketing and sales components. The process for each subtask is as follows:

1. Requirement analysis: In this stage, the requirements for each component—such as the user interface and database backend—are recognized and recorded. The COCOMO II model estimates the effort (in person-days) required for requirement analysis by considering variables such as the number of necessary function points, complexity, and other project-specific characteristics.
2. Design: Each component's thorough design specifications are created during the design process. Once more, the effort is estimated using COCOMO II based on the scope and complexity of the design work.
3. Unit testing: It is part of the testing phase for each component. Considering the unique testing requirements and the complexity of each component, COCOMO II is used to estimate the testing effort needed.

Basics of Estimate – Wideband Delphi:

Here, we estimate the software items such as External Inputs (EI), External Outputs(EO), External Query(EQ), Internal Logic Files(ILF), External Interface Files(EIF), degrees of influence and weighting factor.

		Round 1						Round 2					
		Bhavya	Bhavani	Haritha	Average	STD DEV	CV	Bhavya	Bhavani	Haritha	Average	STD DEV	CV
EXTERNAL INPUTS (EI)	Item 1	20	25	18	21	3.126	0.49863	20	15	19	18	2.645751	0.146986
EXTERNAL OUTPUTS (EO)	Item 2	15	20	12	15.6	1.53	1.637248	11	15	15	17	2.464102	0.103771
EXTERNAL QUERY (EQ)	Item 3	10	15	8	11	1.23	0.123718	15	15	15	15	5	0.3333
INTERNAL LOGIC FILES (ILF)	Item 4	25	30	20	25	6.3	4.237	18	24	18	20	2.4	0.134
EXTERNAL INPUT FILES	Item 5	10	12	8	10	1.12	3.112	10	10	13	21	3.126	0.19863
14 DEGREES OF INFLUENCE	Item 6	4	3	4	3.89	0.123	0.004	5	3	3	3.89	0.123	0.004
WEIGHTING FACTOR	Item 7	1.2	1.1	1.3	1.2	0.0012	0.0034	1.4	1.2	1.1	1.2	0.0012	0.0034

RATIONALE:

- Due to the project's substantial data conversions and calculations, complex data processing is required.

- There is a sizeable amount of data, but it is not an excessive amount, and effective data management solutions are in place.
- The project necessitates integrating with numerous external systems, which makes interfaces more complex.

Basic of Estimate – FUNCTION POINTS:

1. Productivity Rate - Construction (FP/day):

Novice: 0.1 FP/day

J Journeyman: 0.2 FP/day

Expert: 0.3 FP/day

Rationale: Based on Function Points, these rates show how productive workers can be expected to be during the project's building period. It is assumed that beginners will generate work more slowly than specialists, who work more quickly.

2. Testing Time FP (FP/workday):

Novice: 2 FP/workday

J Journeyman: 3 FP/workday

Expert: 4 FP/workday

Rationale: These rates show how many Function Points testers of various levels of experience can test in a single workday. In testing, beginners require more time while specialists test more quickly.

3. Errors / FP (Error/5 FP):

Novice: 3 Error/5 FP

Journeyman: 2 Error/5 FP

Expert: 1 Error/5 FP

Rationale: The predicted error rate per five function points found during testing is specified by this option. Experts are more error-conscious, but novices are more likely to introduce errors.

4. Time to Resolve / Error (workdays):

Novice: 3 workdays

Journeyman: 2 workdays

Expert: 1 workday

Rationale : This option shows how long it typically takes to fix a mistake. Experts can fix problems more quickly while novices may take longer.

Schedule

The SIX project, which entails the creation of cutting-edge streaming technology, is described in the IMS (Integrated Master Schedule) that was delivered. This includes key tasks and their duration dates. Basically, it is a way of scheduling the tasks in a much more formatted way.

TASK ID	TASK DESCRIPTION	START DATE	END DATE	DURATION
1.	User Interface			51 days
	Requirement Analysis	09/08/2023	09/19/2023	11 days
	Design	09/22/2023	10/15/2023	23 days
	Code & unit test	10/16/2023	11/1/2023	17 days

2.	DataBase Backend			59 days
	Requirement Analysis	09/10/2023	09/22/2023	12 days
	Design	9/12/2023	10/3/2023	21 days
	Code & unit test	10/4/2023	10/30/2023	26 days
3.	Network & Communication			48 days
	Requirement Analysis	9/29/2023	10/11/2023	13 days
	Design	10/9/2023	10/27/2023	18 days
	Code & unit test	10/28/2023	11/15/2023	17 days
4.	Sound			47 days
	Requirement Analysis	10/1/2023	10/15/2023	15 days
	Design	10/9/2023	10/26/2023	17 days
	Code & unit test	10/26/2023	11/11/2023	15 days
5.	Marketing & Sales			37 days
	Requirement Analysis	11/11/2023	11/20/2023	10 days
	Design	11/12/2023	11/25/2023	12 days
	Code & unit test	11/15/2023	11/30/2023	15 days

Software Process Model

The Agile development methodology is the best development process methodology for the SIX project. Agile's iterative development methodology adapts to the project's complexity, enabling the production of incremental features and adaptation to changing customer requirements and technological developments. Agile's customer-centric approach and adaptability fit the project's many components and multiple stakeholders' engagement. Continuous integration and testing are facilitated, which is important for network communication and specific remote capabilities. The project's 18-month time to market and the requirement for early risk identification are both compatible with Agile's adaptability. In conclusion, Agile enables the SIX project to manage complexity, lower risks, and successfully provide incremental value.

RATIONALE:

For a project like SIX, which has numerous components including User Interface (UI), Backend Development, Quality Assurance (QA), Network and communication, Sound, and Marketing and sales, Agile's iterative and incremental approach is quite useful. Agile makes it possible to continuously improve these elements, adapting to shifting client needs and new technical developments.

In a project with several stakeholders and a broad user base, Agile's customer-centric methodology is essential. Through regular customer contact and feedback throughout the development process, it promotes user expectations alignment in the finished product.

Agile approaches, which include continuous integration and testing, are especially helpful for components like network and communication, which need seamless integration and thorough testing to guarantee stability and performance.

Agile can be used to manage the SIX project's 18-month time to market. It is suitable for a project that requires flexibility and a quick response to new obstacles because to its adaptability and capacity to reprioritize work.

Additionally, Agile emphasizes early risk identification and mitigation, which is consistent with the project's need to quickly identify and handle any potential issues.

3. Problem Definition

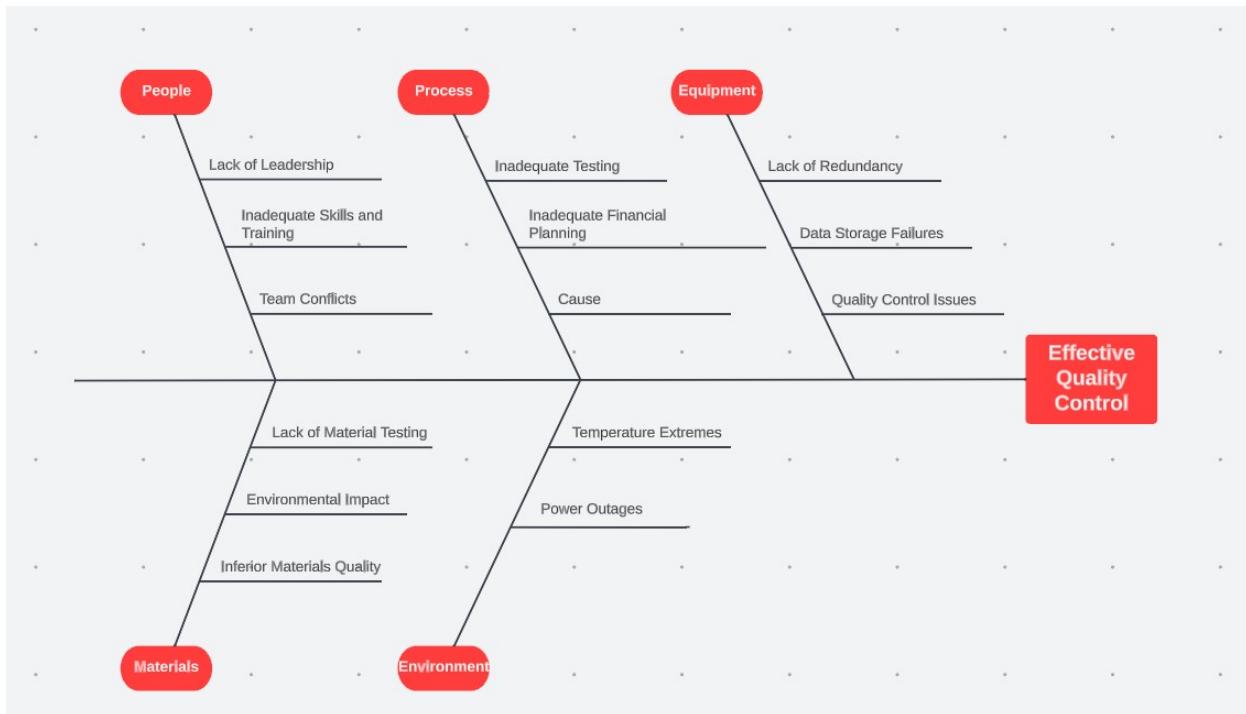
Problem Statement

Element	Description
The problem of	The issue of traditional TVs lacking smart capabilities having limited access to streaming content.
Affects	Relates to customers who have access to a variety of streaming services and own conventional televisions.
the result of which	Users lack a cohesive interface for managing various streaming subscriptions, necessitating the purchase of pricey smart TVs or other external devices.
benefits of	The advantages of finding a solution to this issue include expanding user comfort, democratizing access to streaming content on any TV, and maybe upending the status quo of traditional TV by presenting a radical idea.

Root Causes

The root causes for the project are explained here using a fishbone diagram. This explains the reason for our sixth and final attempt at this project. All the causes of failure in the last five attempts are shown here.

The fishbone diagram for the project is shown below.



Stakeholders & Users

Users	Other stakeholders
Viewers of content	Regulatory Organizations
End Users	Content Creators
Developers	Investors/Shareholders
Client Services Representative	Advisory Board
UI Testers	Competitors

Viewers of Content: These users watch videos or read articles on the platform.

End users: These are those who work with the system on a daily basis, posting comments and creating accounts.

Developers: They create and manage the system, developing code and resolving problems as they arise.

Customer service representatives: They help users and make sure they have a good time

UI testers: They assess the user interface of the system and report any problems.

Regulatory Organizations: Organizations charged with enforcing regulations on systems are known as regulatory organizations.

Creators of material: Individuals or organizations who produce material.

Investors: Financially motivated parties are called investors or shareholders.

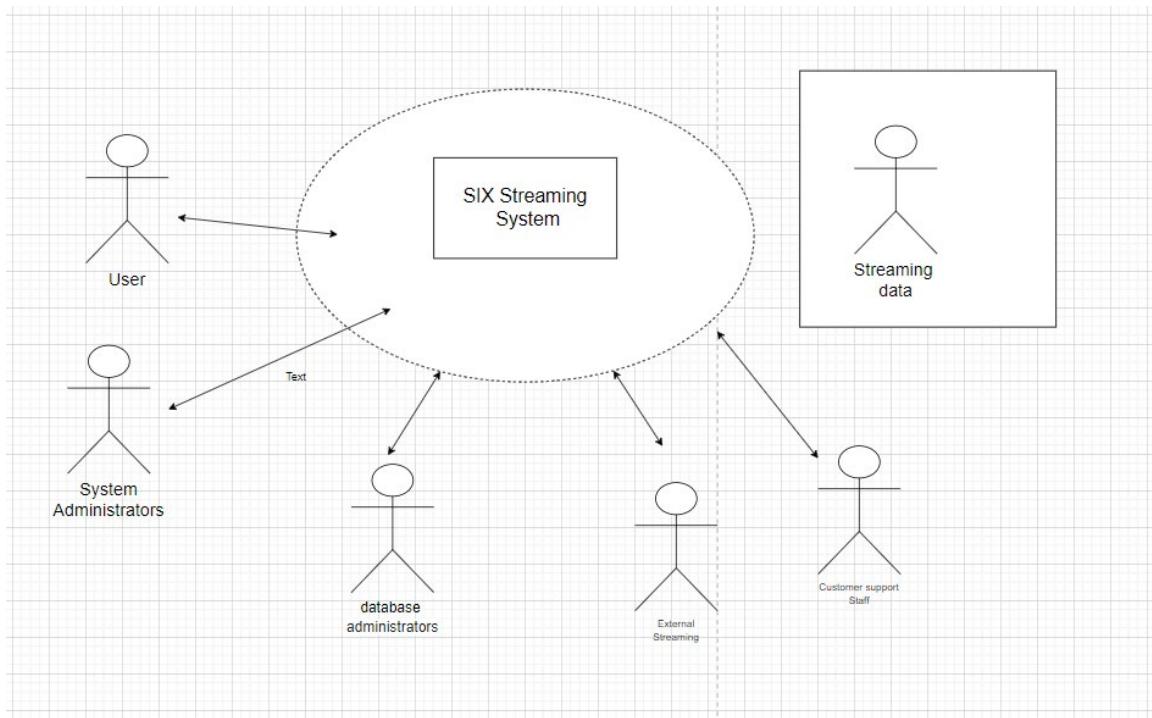
Advisory Board: Provides leadership of the organization with strategic advice.

Competitors: Rival businesses operating in the same sector are competitors.

System Boundary Diagram

An illustration of the scope and limits of a system or process is a system boundary diagram. To illustrate how the system interacts with outside parties and to specify the environment in which the system operates, it is frequently used in systems engineering, software development, and business analysis.

Below is the system boundary diagram for the project.



Constraints

Source	Considerations
System	<ul style="list-style-type: none"> The SIX system is meant to be compatible with and seamlessly integrated with existing systems. To ensure a seamless transition for users, compatibility with current solutions is of the utmost importance.
Environmental	<ul style="list-style-type: none"> Constraints related to the environment and regulations are not important for this project. To protect user data and privacy, the system will comply to strict security standards.
Schedule and resources	<ul style="list-style-type: none"> In order to maximize effectiveness and financial control, the project is mostly restricted to already-existing firm resources. For specific jobs, a limited amount of outside labor may be employed with careful consideration of the budget.

	<ul style="list-style-type: none"> ■ To fulfill project objectives and deliver on time, plans are in place to augment resources as needed.
Economic	<ul style="list-style-type: none"> ■ Project planning takes financial and budgetary restrictions into account, resulting in effective resource allocation. ■ To reduce any influence on the project's budget, licensing concerns will be assessed and dealt with.
Political	<ul style="list-style-type: none"> ■ No internal and external Political Issues ■ Internal collaboration are part of our team approach
Technical	<ul style="list-style-type: none"> ■ The project has the freedom to choose the finest technologies within predetermined limits. ■ develop to be compatible with current platforms and technologies and to operate fluidly with them. ■ The use of new technology is not prohibited, so this is not a problem.

System: The SIX project's infrastructure and core technologies are referred to as the system aspect.

Environmental: Environmental issues include variables that are beyond of the project's direct control yet could influence its progress.

Schedule and Resources: Project planning, budgeting, and resource allocation all include taking schedule and resource factors into account.

Economic: Factors relating to money and spending are included in economic issues.

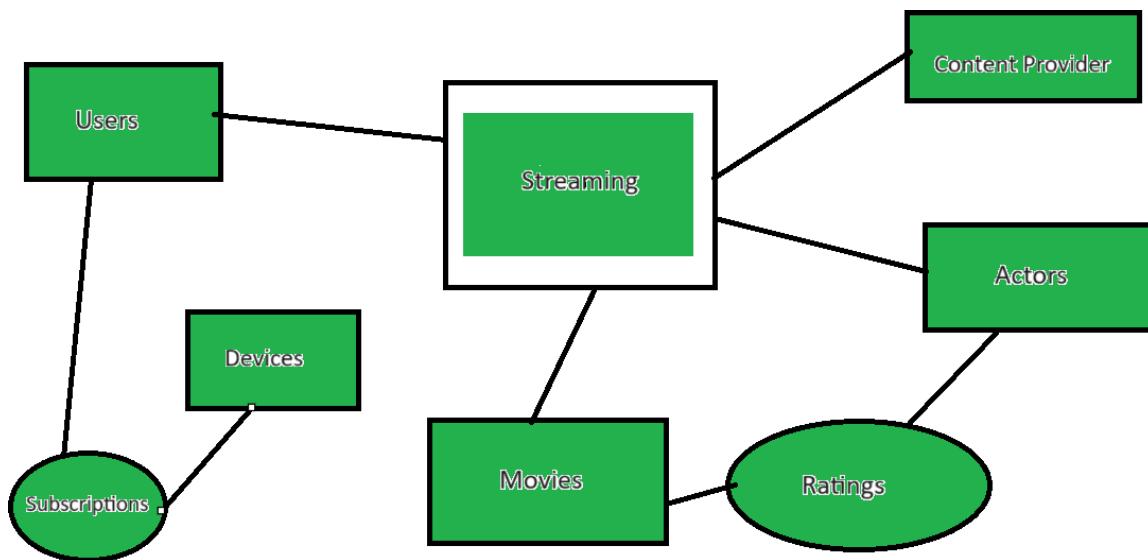
Political: Organizational and team dynamics are considered while making political decisions.

Technical: Technical considerations relate to innovation, compatibility, and technology decisions.

4. Frame Diagram

A diagram that depicts the general architecture or organization of a software application is known as a "frame diagram" in most contexts. This kind of diagram is used to show how different software system layers, modules, or components interact with one another. For purposes of high-level design and planning, frame diagrams are frequently employed.

Below is the Frame diagram for the project.

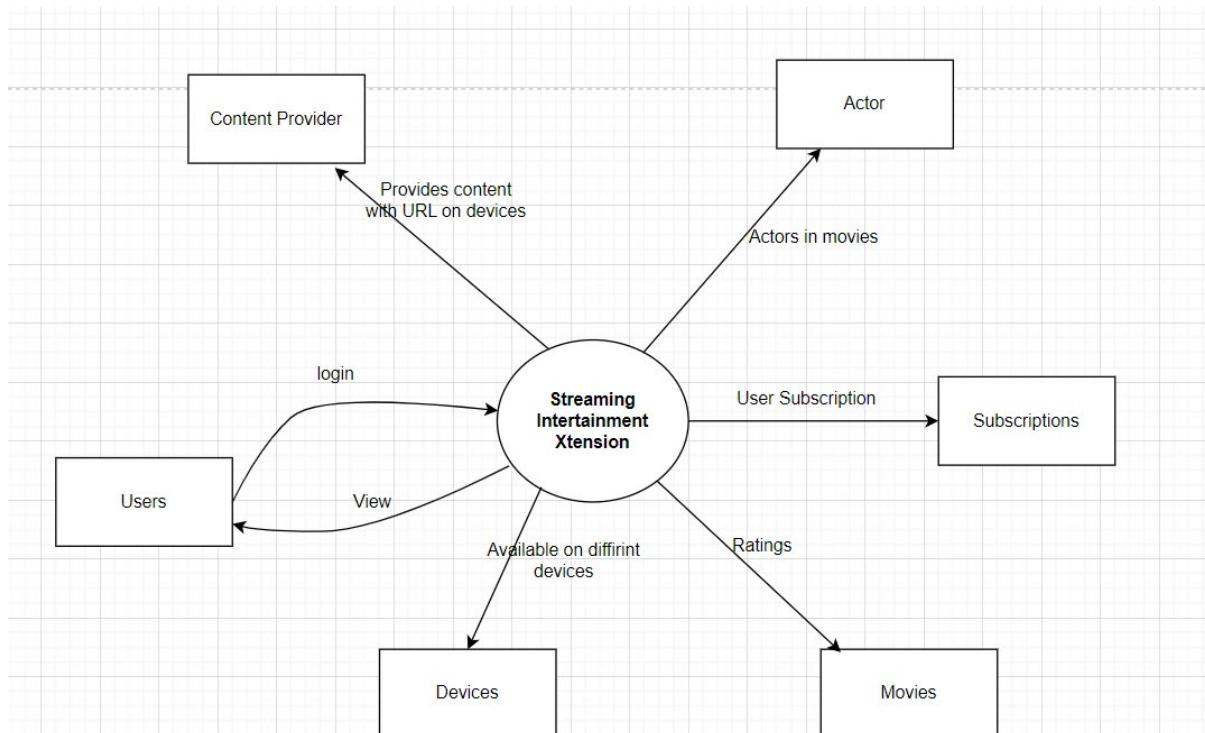


5. Context Diagram

An overview of a system or process and its relationships with outside parties can be seen visually in a context diagram. The system's borders and its connections to outside actors or entities are

depicted in this high-level diagram, which is used in systems engineering, software development, and business analysis. Understanding the scope of a system and how it interacts with its environment is made easier by using context diagrams.

Below is the context diagram for the project.

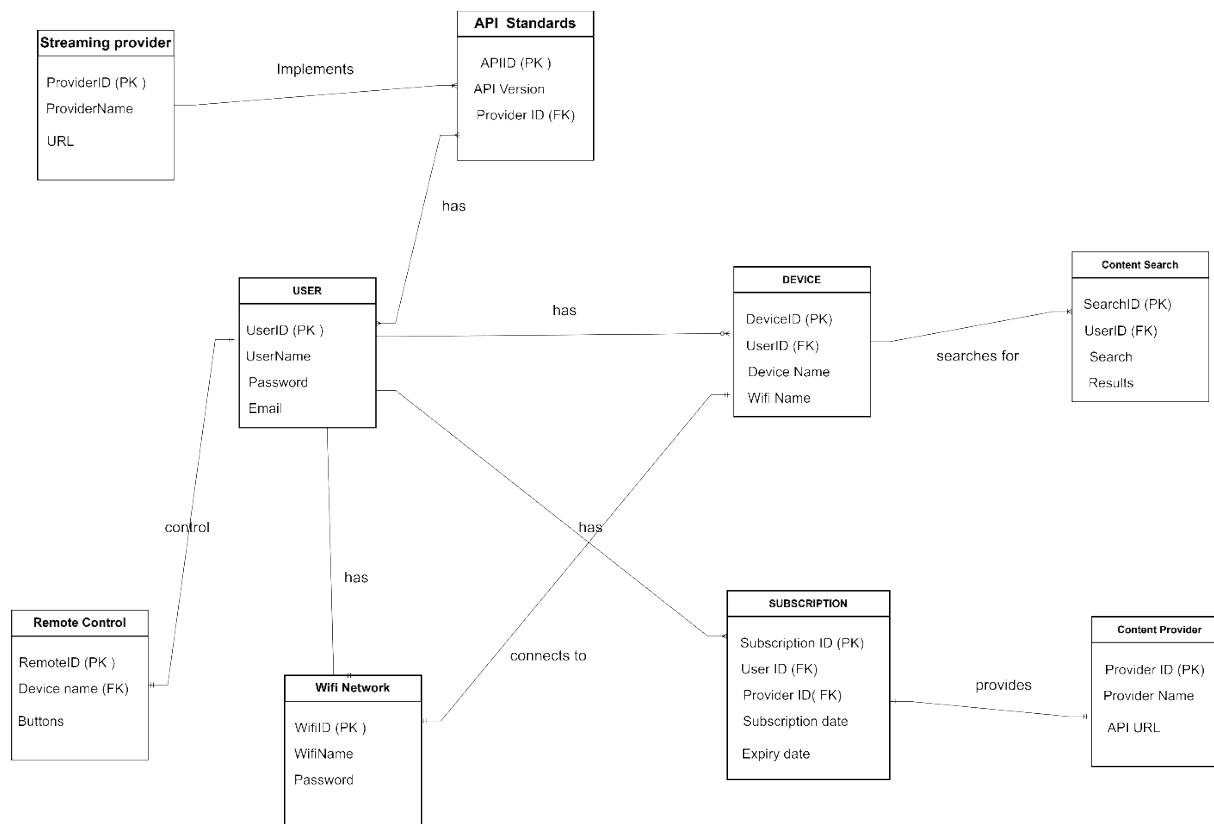


6. Entity Relationship Diagram

A visual representation of the database's structure's data model is an entity-relationship (ER) diagram. It is a graphical tool that is used in database design to represent the entities (objects about which data is stored), their attributes (properties or characteristics), and the connections between them.

The attributes in the ER diagram are User, device, Subscription, Content provider, Wifi Network, Content search, Streaming provider etc. Here, we can see that the user has one to many relationships with device and subscription. As user can log in to different devices and can have many subscriptions.

Below is the ER diagram for the project.

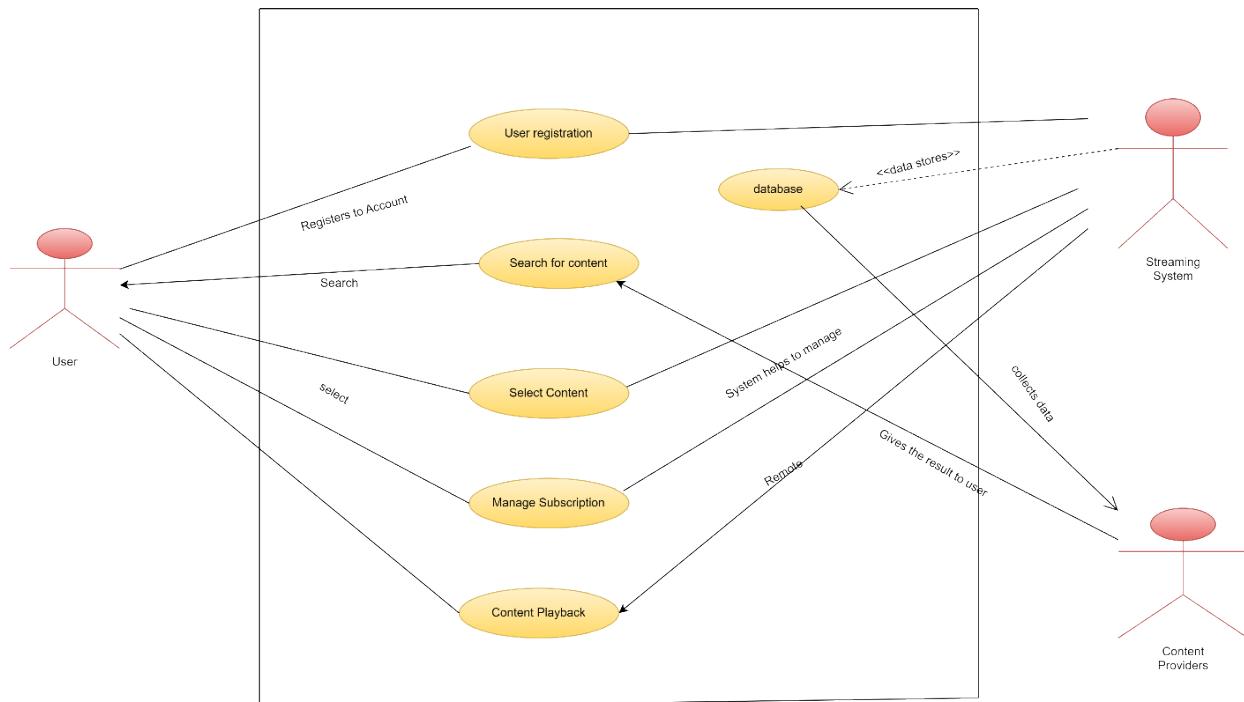


7. Use Cases

One kind of UML (Unified Modeling Language) diagram that shows how actors interact with a system is a use case diagram. It helps to show the relationships between the actors (users or external systems) and the various use cases (particular functionalities or behaviors) of the system.

by giving a visual overview of the functional needs of the system. Use case diagrams are mostly employed to represent a system's functional needs from the viewpoint of its users.

Below is the use case diagram.



UC ID	UC Name
UC1	User Registration
UC2	Search for movie/Actors
US3	Select Content
UC4	Manage Subscription
UC5	Content Playback

UC6	Database
------------	----------

Normal Scenarios

Use Case: User Registration
ID: UC1
Actors:
User, SIX system
Pre-conditions:
The user has internet access and a functional SIX device.
Flow of Events:
<ol style="list-style-type: none"> 1. On the SIX gadget, the user chooses the "Register" option. 2. The user is prompted by the system to set a password and input their email address. 3. The user completes the registration form by entering the necessary information. 4. After confirming the user's details, the system emails a verification to the recipient. 5. By clicking the link in the email verification, the user validates their registration
Post-conditions:
The user has a registered account linked to their device.

Use Case: Search for Movie / Actors
ID: UC2

Actors:
User, SIX system, Content Providers
Pre-conditions:
The device is connected to internet.
Flow of Events:
<ol style="list-style-type: none"> 1. On the SIX gadget, the user chooses the "Search" option. 2. The user can enter their question into the search box displayed by the system. 3. The user starts the search by typing in the name of the chosen film or actor. 4. The integrated IMDB database is accessed by the system to retrieve pertinent content. 5. To view relevant streaming alternatives, the user picks a particular film or star.
Post-conditions:
The user can view selected content from various content providers.

Use Case: Manage Subscriptions
ID: UC3
Actors:
User, SIX system, Content Providers
Pre-conditions:
The user has active subscriptions linked to their account.
Flow of Events:
<ol style="list-style-type: none"> 1. On the SIX device, the user chooses the "Subscription Management" option. 2. A list of all active subscriptions linked to the user's account is shown by the system. 3. To see a subscription's details, the user chooses one.

- | |
|---|
| 4. The user selects and validates the cancellation of a certain subscription. |
| 5. In accordance, the system modifies the subscription status. |

Post-conditions:

The user can view and manage their subscriptions effectively.

Use Case: Content Playback

ID: UC4

Actors:

User, SIX system

Pre-conditions:

The selected content is available for streaming.

Flow of Events:

1. On the SIX gadget, the user chooses a video from the list of selections.
2. The chosen content is started to playback by the system.
3. The user operates the playback features (such as pause, play, rewind, and fast-forward) with the unique remote.
4. The user modifies the other settings, including the volume, as needed.
5. By shutting off the device or leaving the video, the user terminates the playback session.

Post-conditions:

The user can control the video playback seamlessly.

Abnormal Scenarios

Use Case: User Registration
ID: UC1
Actors:
User, SIX System
Pre-conditions:
The user has internet access and a functional SIX device.
Flow of Events:
<ol style="list-style-type: none"> 1. On the SIX gadget, the user chooses the "Search" option. 2. The user can enter their question into the search box displayed by the system. 3. The user starts the search by typing in the name of the chosen film or actor. 4. The integrated IMDB database is accessed by the system to retrieve pertinent content. 5. To view relevant streaming alternatives, the user picks a particular film or star.
Post-conditions:
The user can control the video playback seamlessly.
Extensions / Alternate Flows:
3a: Network Interruption:
3a1: The system displays the message indicating the network interruption. 3a2: User attempts to reconnect to internet. 3a3: Return to step 3a2.
Post-conditions:
The user can control the video playback seamlessly.
3b: Error in Password Creation:
3b1: The password entered by the user does not match the requirements set forth by the system.

<p>3b2: The required password is displayed in an error message by the system.</p> <p>3b3: The user modifies the password in accordance with the specifications of the system.</p> <p>3b4: After changing the password, the user submits it.</p>
<p>Post-conditions:</p> <p>The user has successfully created a password that meets the system's requirements.</p>

<p>Use Case: Content Playback</p>
<p>ID: UC4</p>
<p>Actors:</p>
<p>User, SIX System</p>
<p>Pre-conditions:</p> <p>The selected content is available for streaming.</p>
<p>Flow of Events:</p> <ol style="list-style-type: none"> 1. On the SIX gadget, the user chooses a video from the list of selections. 2. The chosen content is started to playback by the system. 3. The user operates the playback features (such as pause, play, rewind, and fast-forward) with the unique remote. 4. The user modifies the other settings, including the volume, as needed. 5. By shutting off the device or leaving the video, the user terminates the playback session.
<p>Post-conditions:</p> <p>The user has a registered account linked to their device.</p>
<p>Extensions / Alternate Flows:</p>
<p>3a: Buffering Issue:</p>

- | |
|--|
| 3a1: The system tries to buffer the video, but it takes a long time. |
| 3a2: A notification indicating a buffering problem is displayed by the system. |
| 3a3: The user gives the video some time to properly buffer before continuing. |

Post-conditions:

The video playback resumes smoothly without further interruptions

3b: Playback Error:

- | |
|--|
| 3b1: When the playing process is underway, the user receives an error notice. |
| 3b2: The computer recommends either reloading the page or pausing the playing. |
| 3b3: The user tries to restart or refresh the playback by adhering to the system's instructions. |

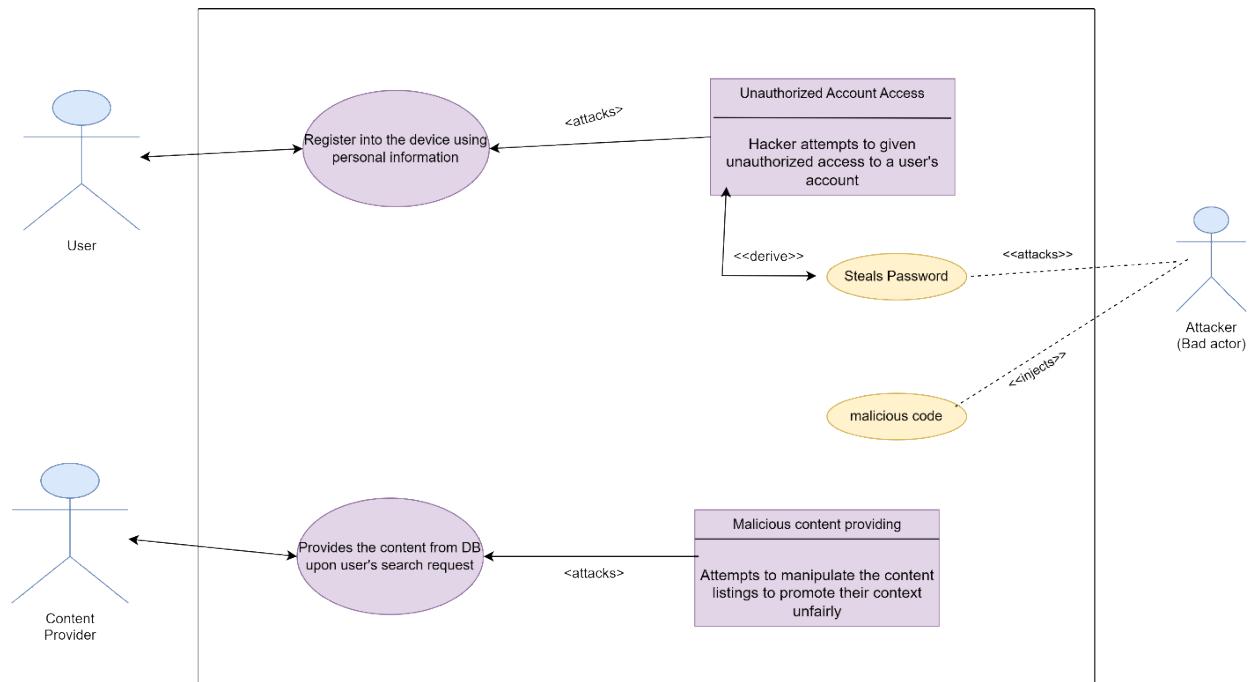
Post-conditions:

The user successfully resolves the playback error and continues watching the content without any further interruptions.

Misuse Case Diagram (Extra Credit):

In software engineering, a misuse case diagram is a graphical depiction that shows how an application might be abused. Its foundation is the idea of misuse cases, which are intended to spot and stop possible abuse situations that can endanger the system. Misuse case diagrams facilitate the understanding of the hazards or dangers that a system may encounter and the creation of mitigation or prevention solutions.

Below is the misuse case diagram:



Scenarios:

MisUse Case: Unauthorized Account Access
ID: MC1
Actors:
Hacker, Six system
Pre-conditions:
Information about user accounts is kept in the system's database.
Flow of Events:
<ol style="list-style-type: none"> 1. The hacker uses a brute force assault to try and guess the user's password. 2. When an account is attempted to log in more than once, the system locks it for a while. 3. To get past the security barriers and access the user's personal data, the hacker employs highly skilled techniques. 4. Although the hacker's attempt to access sensitive data is recorded by the system, it is unable to stop them.

5. After retrieving the user's personal data, the hacker may utilize it for nefarious reasons

Post-conditions:

The attacker(hacker) gains access to user's personal information.

MisUse Case: Manipulating Content Listings

ID: MC2

Actors:

Hacker, Six system

Pre-conditions:

Content providers can submit and manage their content listings on the platform with the help of this system.

Flow of Events:

1. The content management tools of the system are accessed by the malicious content provider without authorization.
2. To increase the prominence of their content in the listings, the supplier modifies the material's visibility settings and metadata.
3. The modified information is updated in the content listings by the system, which is unable to identify the unauthorized alterations.
4. On the platform, the modified content appears more prominently than other authentic content from other sources.
5. Increased traffic and interaction benefit the harmful material provider, which could harm the platform's legitimacy and impartiality.

Post-conditions:

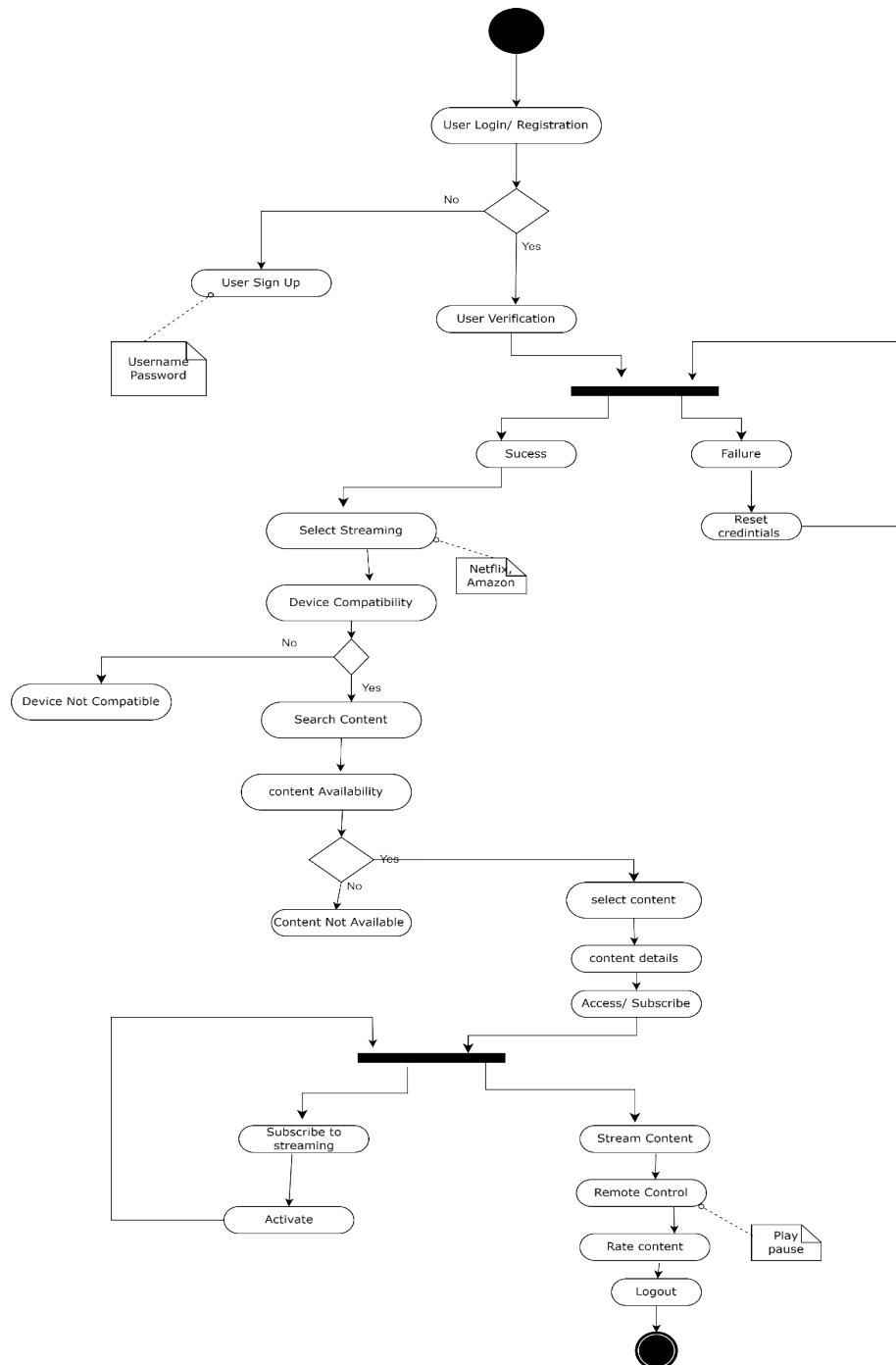
The manipulated content gains visibility on the platform

8. Activity Diagrams

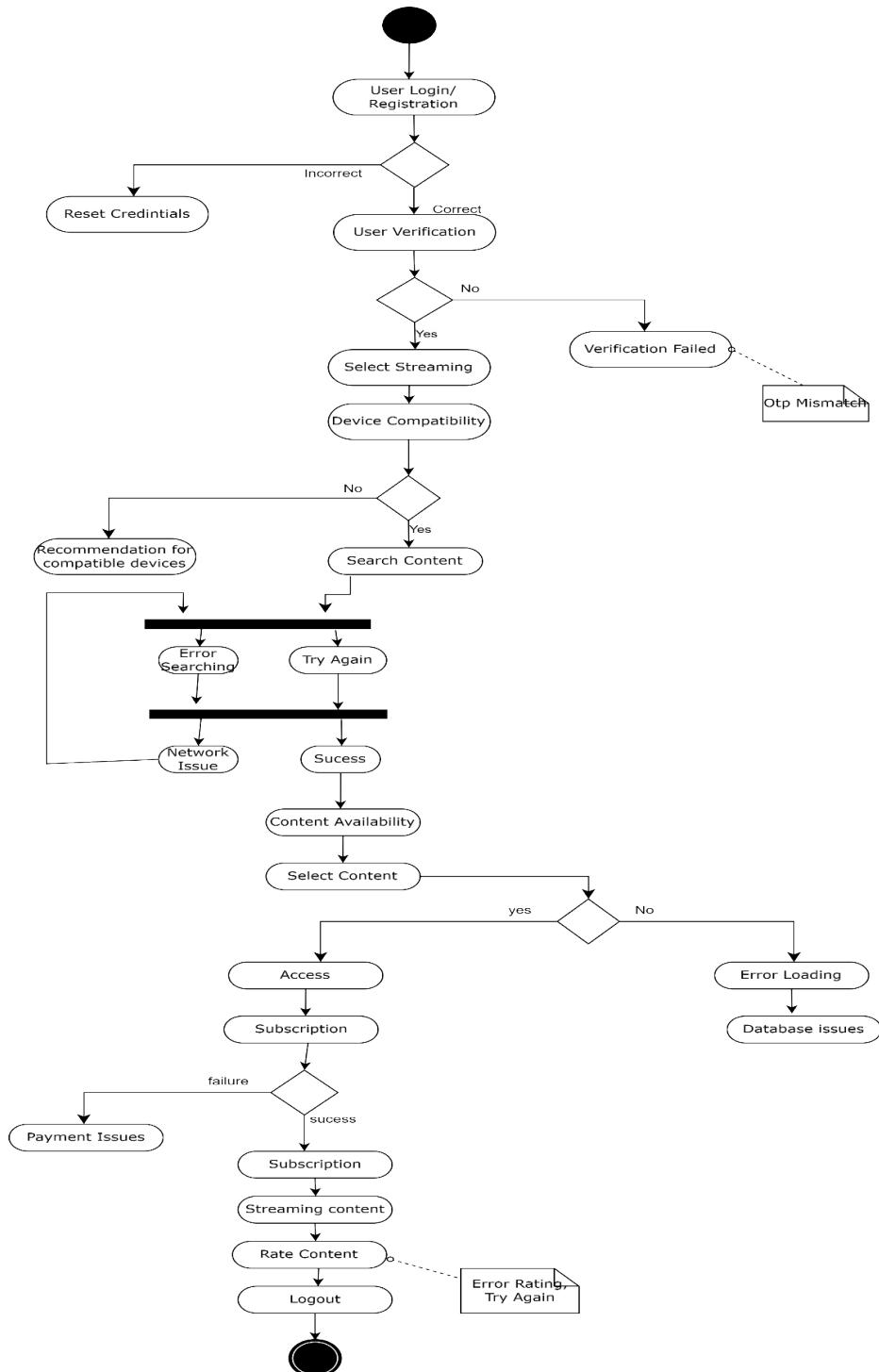
One kind of UML (Unified Modeling Language) diagram that helps you see how activities flow through a system is an activity diagram. It is especially helpful for simulating a system's dynamic features. Activity diagrams are frequently used to represent the flow of activities in a software system, the operational workflow of a specific system component, or the workflow of a business process.

Below are the activity diagrams for normal and abnormal scenarios.

Normal Scenario



Abnormal Scenario

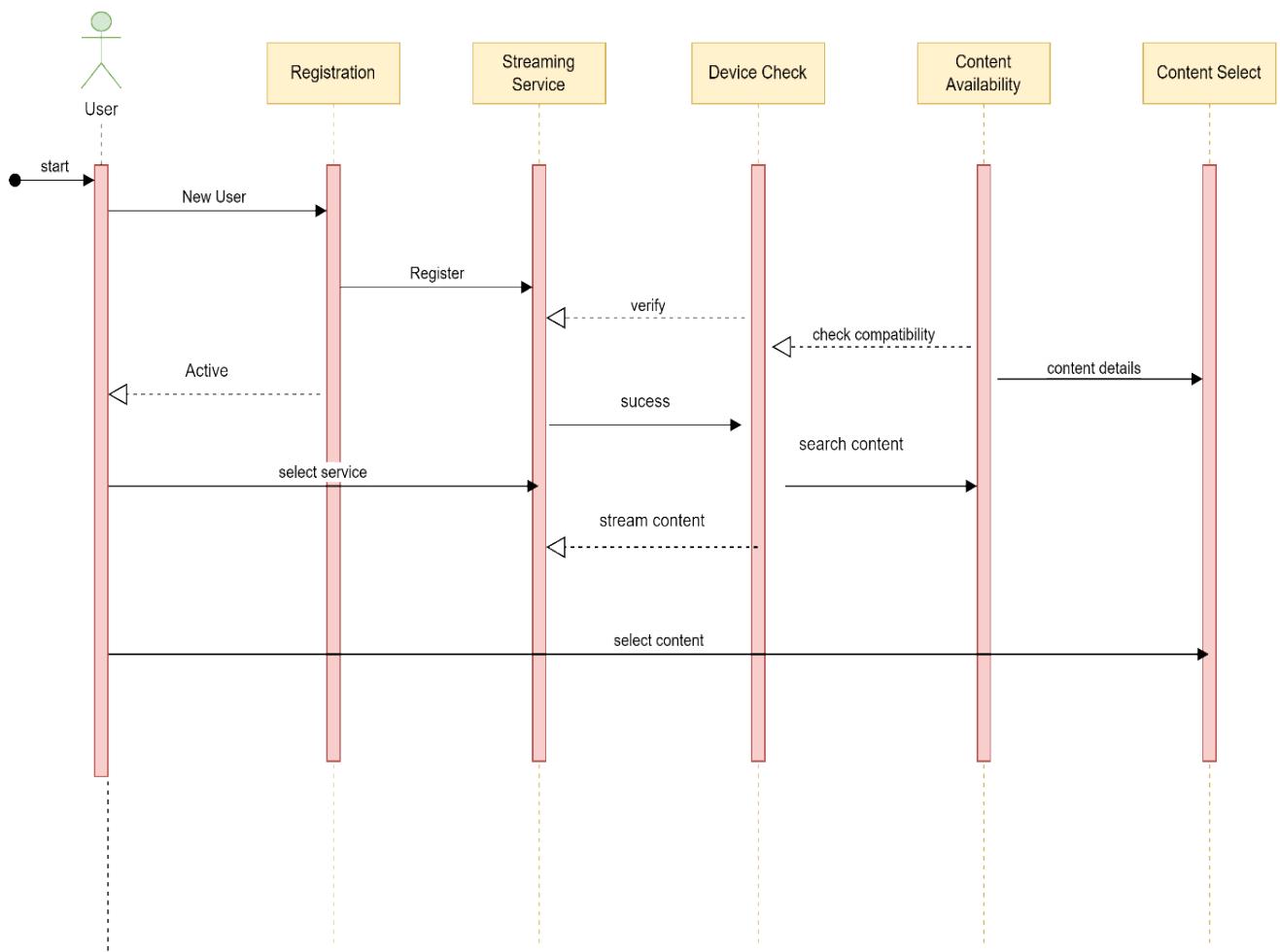


9. Sequence Diagrams

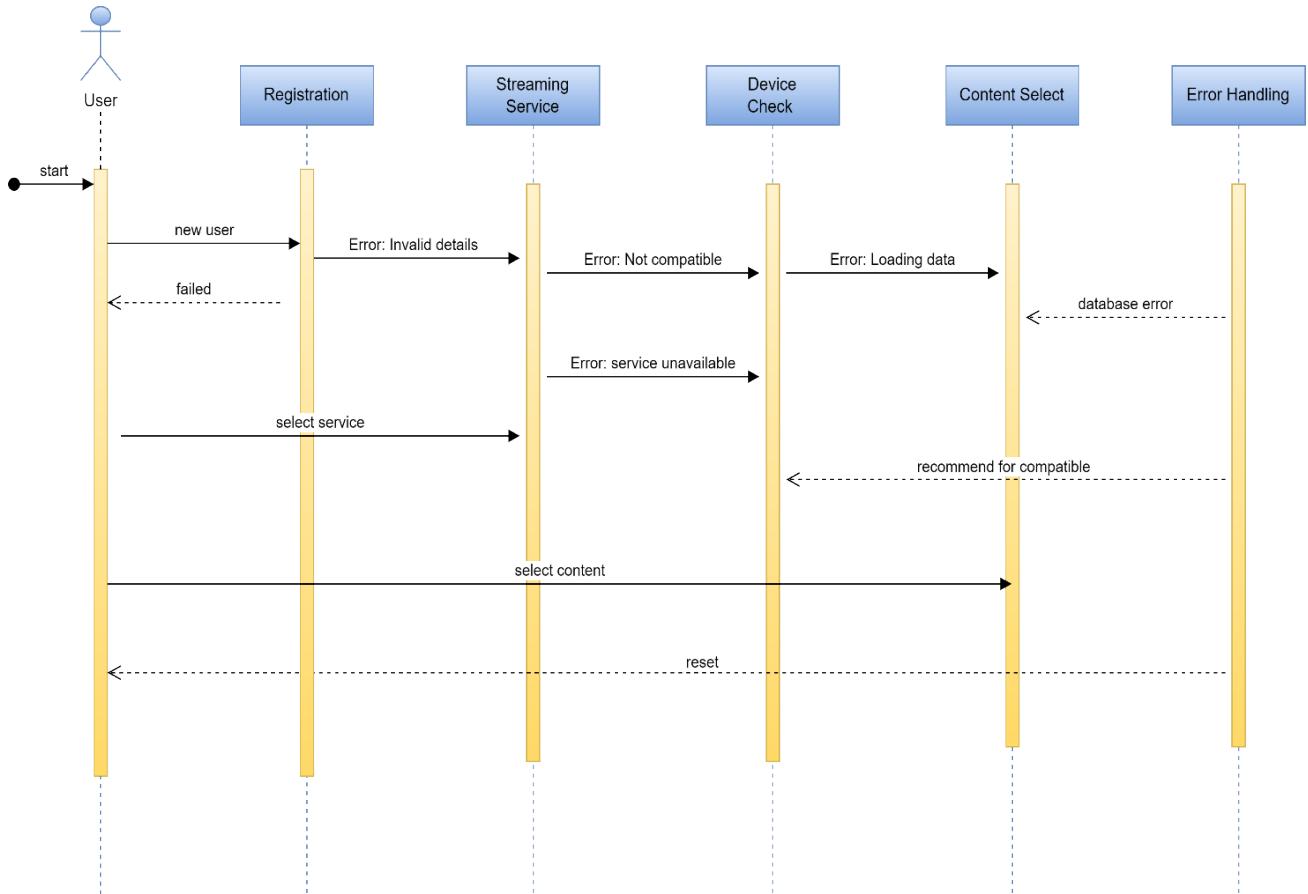
One kind of UML (Unified Modeling Language) diagram that shows the relationships and sequence of interactions between objects or components in a system is the sequence diagram. It enables you to see the system's dynamic behavior by displaying the messages that are sent over time between various objects or components. Sequence diagrams are very helpful for illustrating how various components of a system cooperate and transfer control.

Below are the sequence diagrams for normal and abnormal scenarios.

Normal Scenario



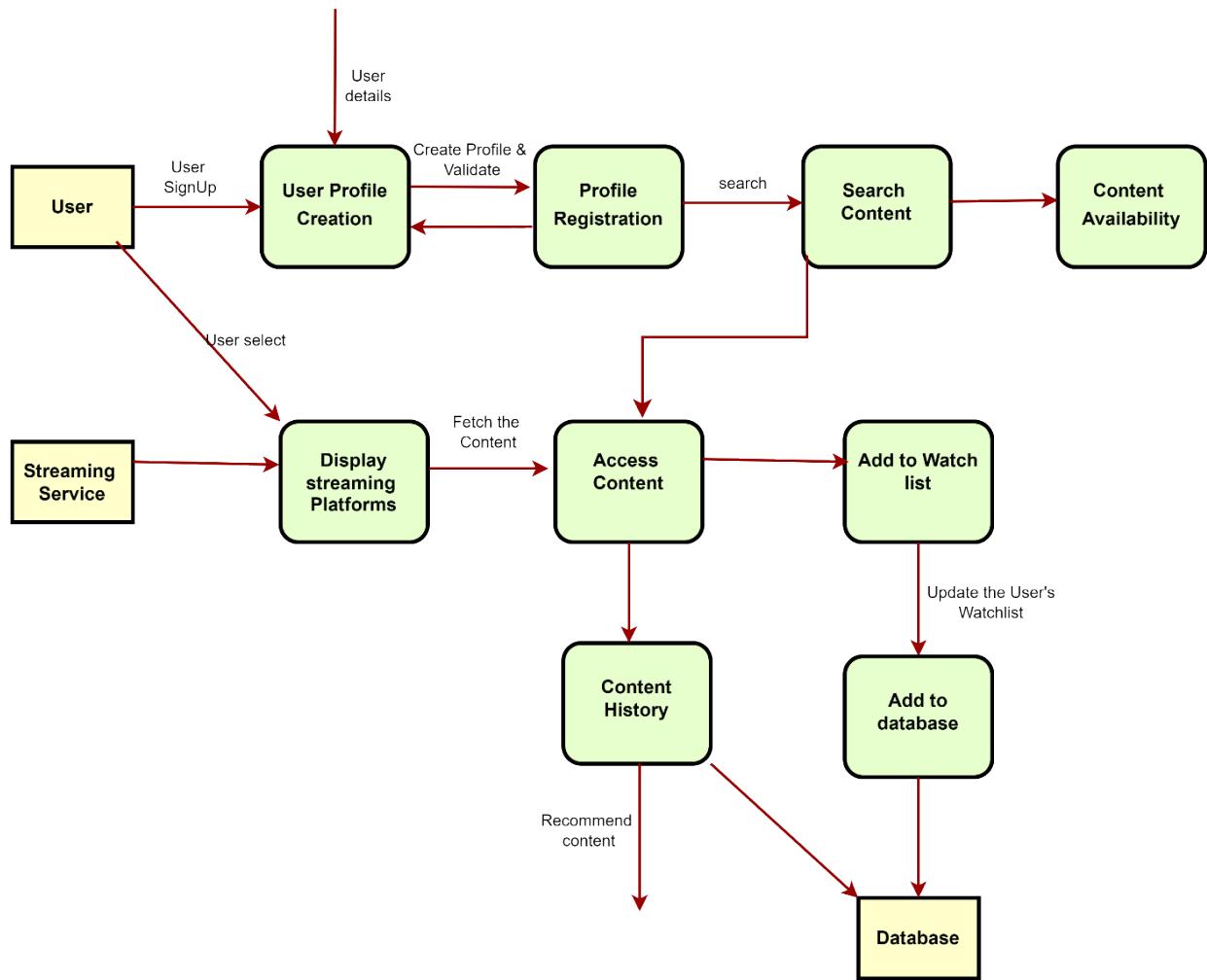
Abnormal Scenario



10. Data Flow Diagram

A data flow diagram (DFD) shows how data moves through an information system graphically. It provides an example of the intake, processing, storing, and output of data within a system. DFDs are used to illustrate the connections between different system components and are a component of the structured analysis and design approach. They offer a succinct and straightforward method of illustrating the data flow and the operations that modify the data within a system.

The data flow diagram for the entire project is as shown below.



11. Functional Requirements

Specific capabilities that a system or component needs to have in order to comply with a formal document such as a contract, standard, specification, or other legally binding agreement are known as functional requirements. They specify the goals of a system, how it should operate in specific scenarios, and how it should respond to specific inputs. In essence, these specifications outline the features or services the system must offer. Usually, functional requirements provide a response to the "what" the system should be able to perform.

The functional requirements for this project are specified below.

FR1 : ACCOUNT MANAGEMENT

	FR1.1	In order to register their gadget, users will be able to create an account on the system.
	FR1.2	Users will be able to control their account settings via the online website or the device interface thanks to the technology.
	FR1.3	Users will be able to connect to their WiFi networks using the system.
	FR1.4	The system must provide the device software update functionality.

FR2: CONTENT SEARCH AND MANAGEMENT

	FR2.1	Users will be able to use the interface of the system to search for stars and movies.
	FR2.2	In order to cross-reference content that is available on streaming platforms, the system will interact with IMDB.
	FR2.3	The User will search content and select from the streaming services
	FR2.4	Through the marketplace, the technology would enable consumers to download and subscribe to different content providers.
	FR2.5	Through the gadget and internet website, the system will enable management of subscription services, including viewing and cancellation.

FR3: INTEGRATION WITH CONTENT PROVIDERS

	FR3.1	A SIX application that complies with the company's API guidelines must run on the system.
	FR3.2	Content providers will be able to use the SIX application for streaming services thanks to the system.

	FR3.3	The SIX applications will be hosted on a marketplace made possible by the system.
	FR3.4	Secure data communication between the device and content providers will be guaranteed by the system.

FR4: REMOTE CONTROL FUNCTIONALITY

	FR4.1	The system needs a custom remote
	FR4.2	The most must contain basic buttons for video playback
	FR4.3	The remote shall allow navigation through menus on the streaming platform.
	FR4.4	The remote shall include custom buttons for seamless redirection to high-profile partners.

FR5: SECURITY AND PRIVACY

	FR5.1	User accounts and personal data protection will be guaranteed by the system.
	FR5.2	User data must be protected by the system both during transmission and storage.
	FR5.3	Secure authentication and authorization mechanisms must be implemented by the system.
	FR5.4	The system must abide with applicable privacy and data protection laws.

FR6: CUSTOMER SUPPORT AND FEEDBACK

	FR6.1	The system will answer questions and offer customer support for technical problems.
--	-------	---

	FR6.2	Users will be able to offer suggestions for enhancements and feedback on the user experience through the system.
	FR6.3	The technology will enable a feedback mechanism so that user complaints and issues may be quickly addressed.
	FR6.4	The system will have an easy-to-use help section for general questions and issues.

12. Nonfunctional Requirements

Instead of focusing on particular behaviors, non-functional requirements (NFRs) are the standards that can be used to assess how well a system functions. Non-functional requirements concentrate on the system's operation, quality features, and operating limitations as opposed to functional requirements, which mainly outline what the system should be able to perform. They are essential for assessing a system's overall effectiveness and user experience. Features like usability, security, performance, and other qualitative features are examples of non-functional requirements.

The non-functional requirements for this project are specified below.

NF1: PERFORMANCE

	NF1.1	The system will buffer and delay-free stream video as little as possible.
	NF1.2	User interactions must have a response time from the system of no more than two seconds.
	NF1.3	Multiple users can stream simultaneously on the system without experiencing any performance issues.
	NF1.4	Peak loads during periods of high traffic must be handled by the system without causing service degradation.

NF2: USABILITY

	NF2.1	The user interface must be simple to use and intuitive for people with different levels of technology experience.
	NF2.2	Clear setup and usage instructions for the device will be supplied by the system
	NF2.3	The method will guarantee that the design is the same on all platforms and gadgets.
	NF2.4	The system must handle several languages in order to serve a wide range of user demographics.

NF3: COMPATIBILITY

	NF3.1	The system must work with a range of gadgets and operating systems.
	NF3.2	A variety of TV models and configurations that are on the market must work with the device.
	NF3.3	Compatibility with a wide variety of streaming service providers and their content libraries will be guaranteed by the system.
	NF3.4	The solution will allow for integration with different payment gateways and subscription management techniques.

NF4: RELIABILITY

	NF4.1	There must be at least 99.9% uptime for the system.
	NF4.2	The system is responsible for guaranteeing uninterrupted and constant streaming service availability.
	NF4.3	During normal operation, the gadget must be resistant to software malfunctions and system crashes.
	NF4.4	The system must provide an automated data backup feature to guard against losing user data.

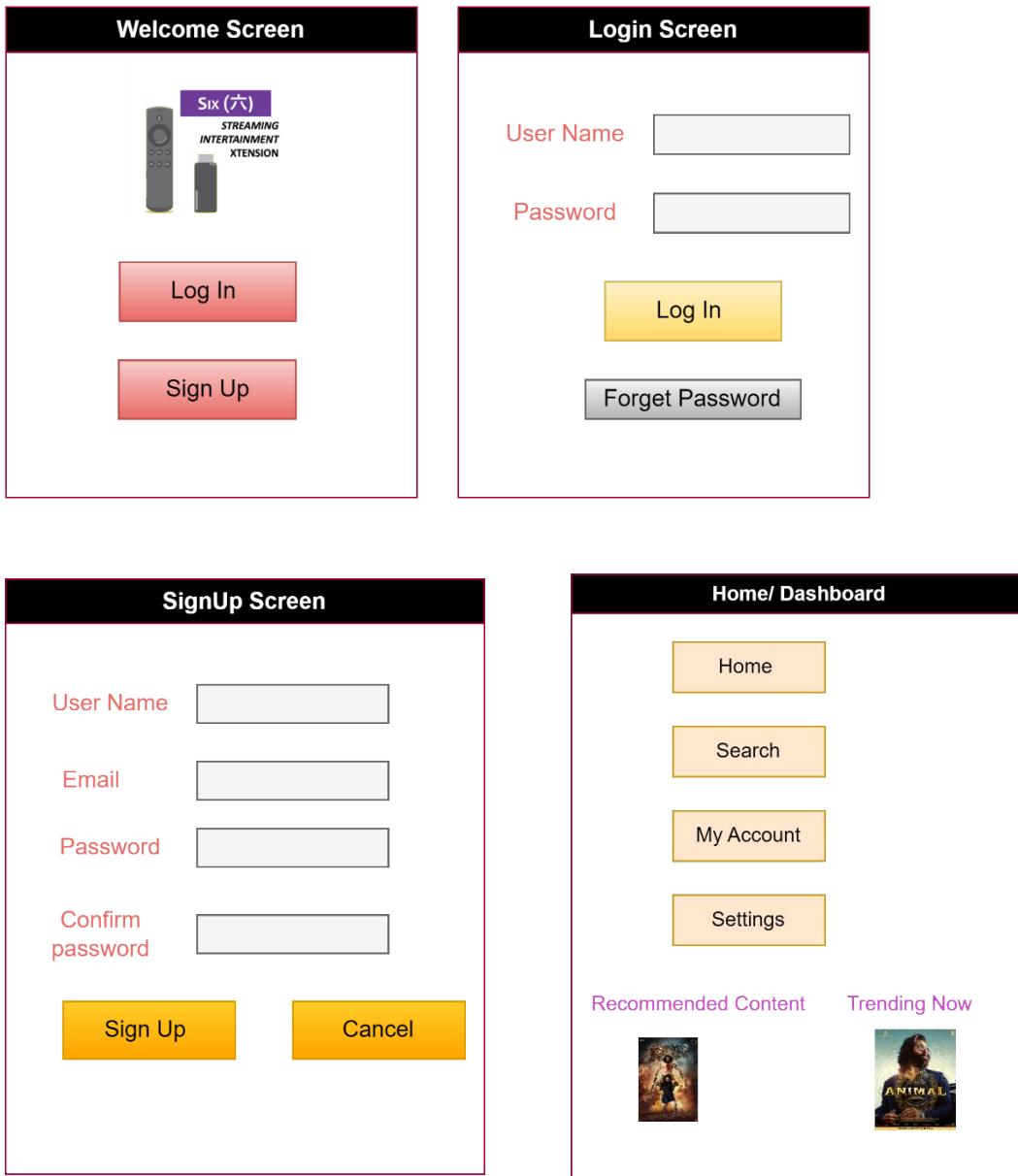
13. Traceability Tables

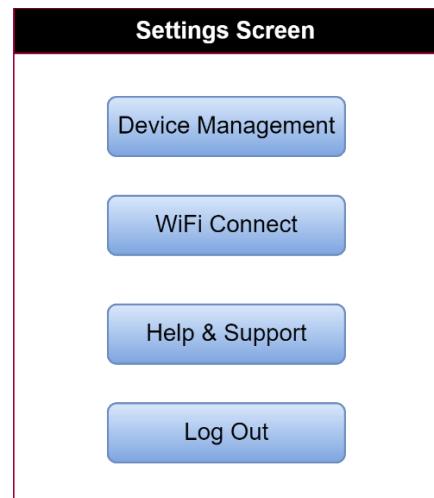
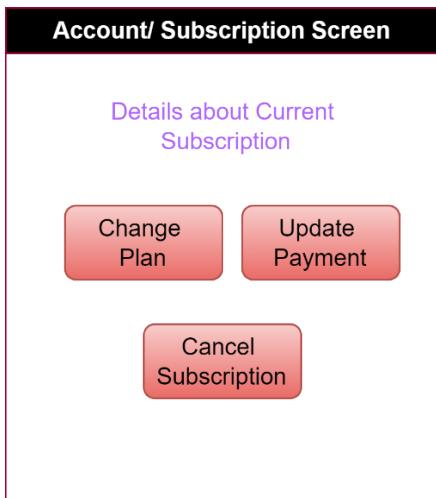
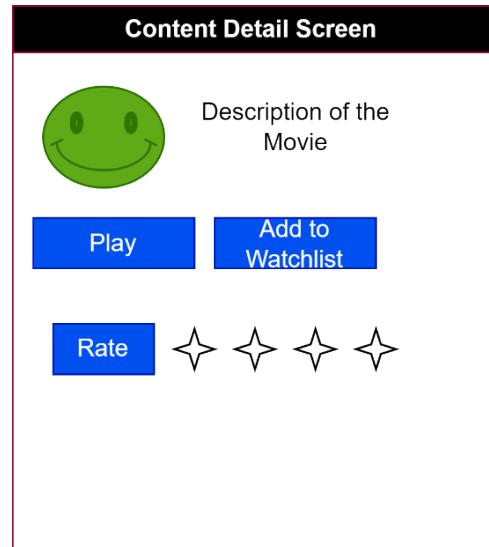
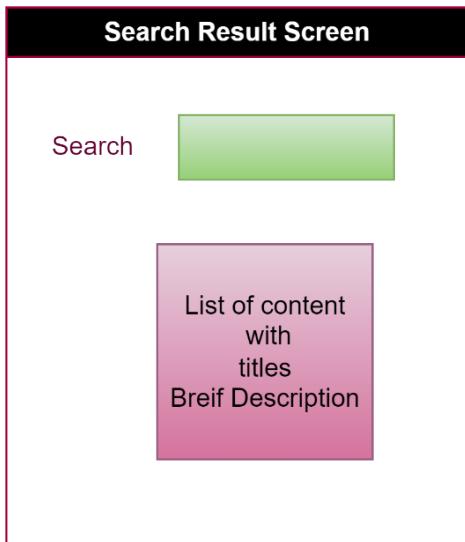
Below is the traceability for all the use cases and their corresponding functional and non-functional requirements.

USE CASE	FUNCTIONAL REQUIREMENTS
UC1	FR1.1, FR1.2, FR1.3, FR1.4
UC2	FR2.1, FR2.2
UC3	FR2.3, FR2.4
UC4	FR3.1, FR3.2, FR3.3, FR4.1, FR4.2, FR4.3, FR4.4
UC5	FR6.1, FR6.2, FR6.3, FR6.4

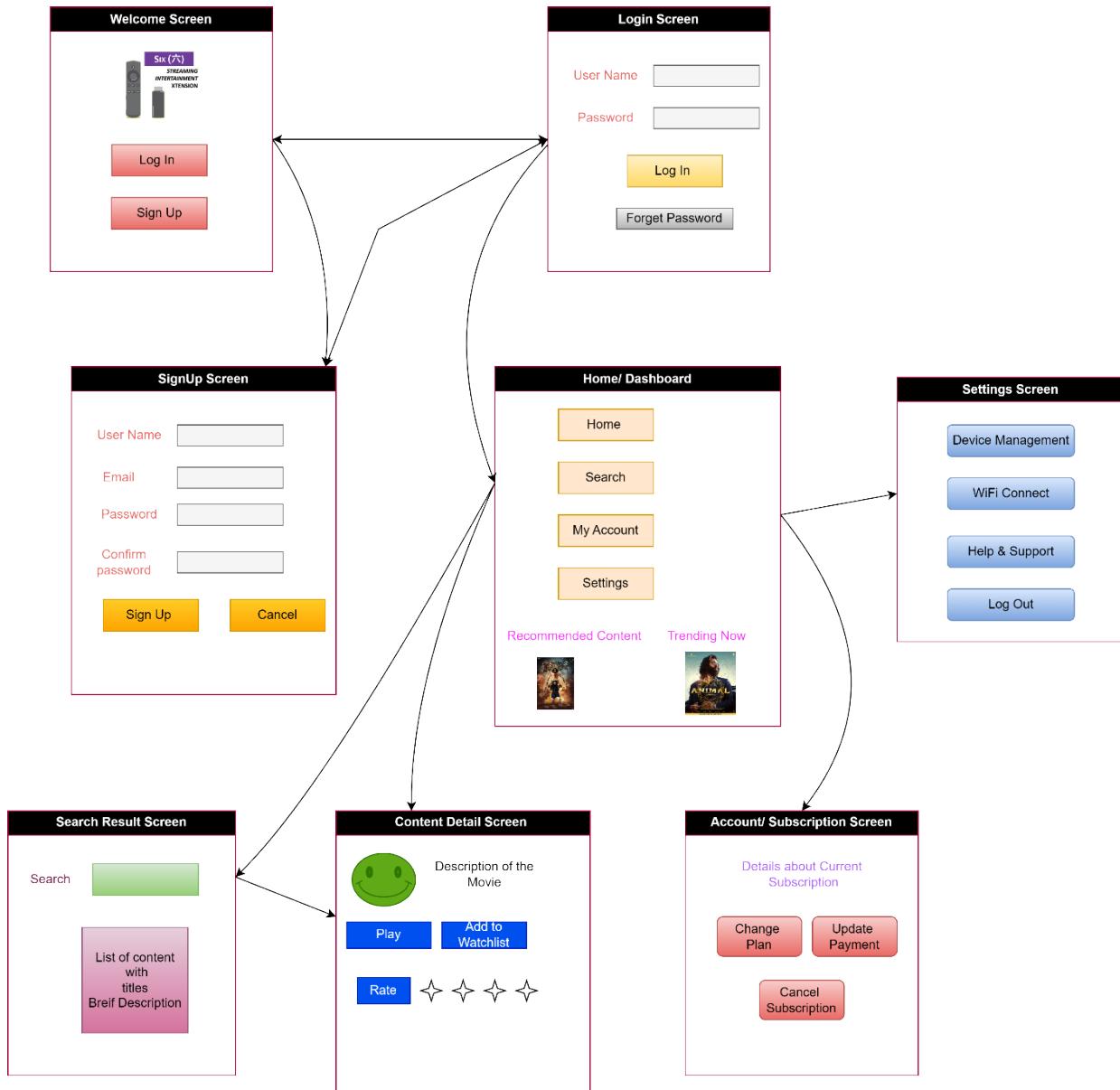
14. User Interface Design

Storyboards





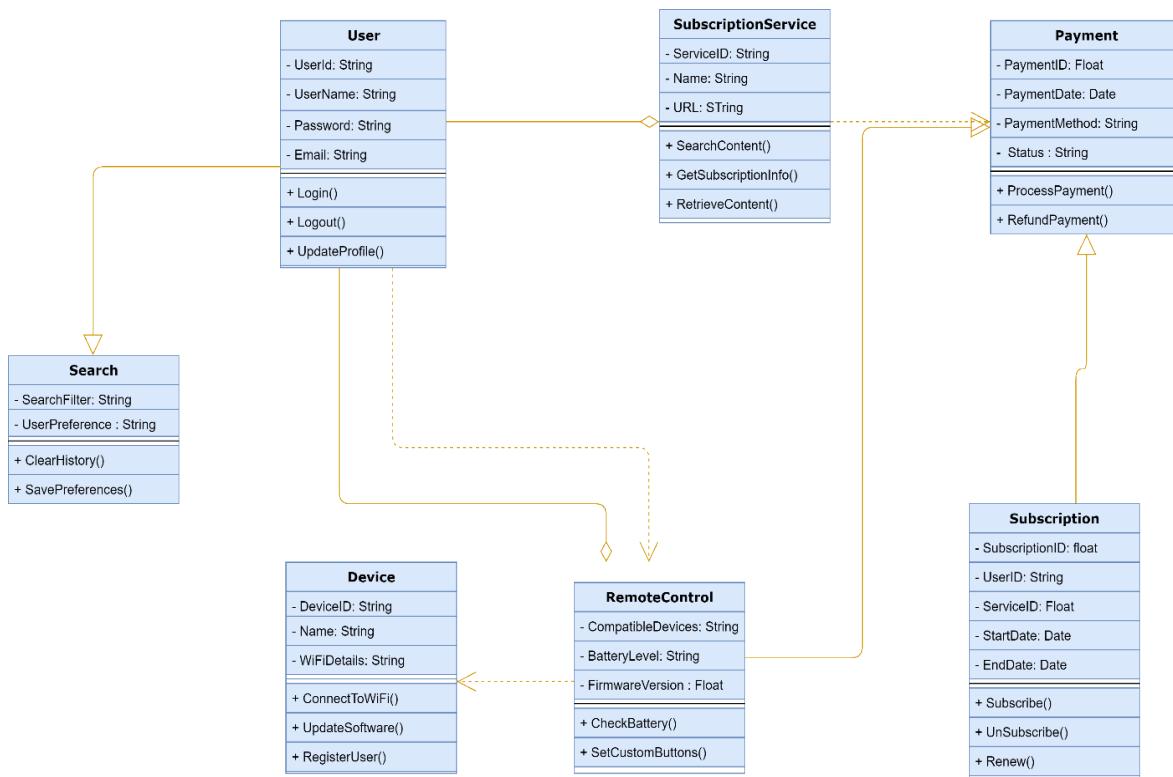
Link Analysis



15. Class Diagram

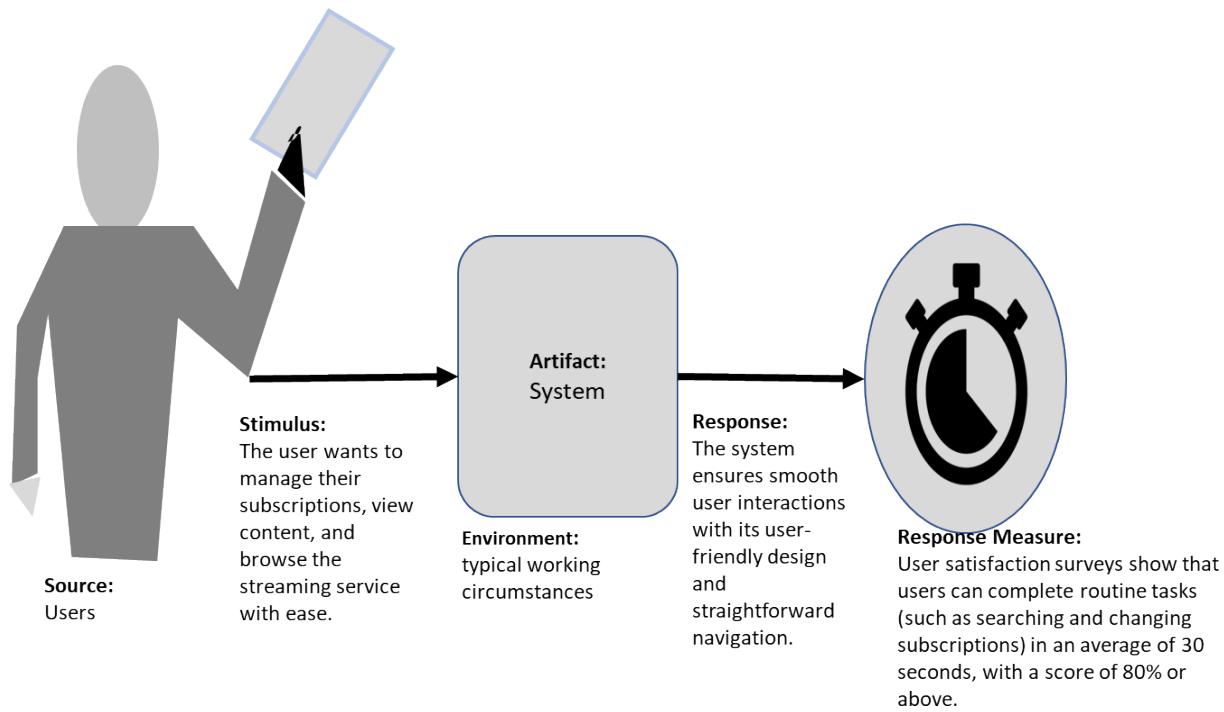
An essential part of the Unified Modeling Language (UML) is the class diagram, which is used in software engineering to graphically depict a system's static structure. The main components are the classes, which are represented as rectangles. Each class has three sections: one for the class name; another for attributes (data members); and a third for methods (functions or operations). Associations indicate how examples of one class

connect to instances of another, illustrating the relationships between classes. The number of instances involved in these relationships is indicated by multiplicity notations. Dependencies are shown as dashed lines with arrows, which indicate class interdependencies, but inheritance is shown as a solid line with a triangle arrowhead, signifying the "is-a" relationship between classes. Additional details about an association are captured by association classes.



16. Quality Attribute Scenarios / Tactics

USABILITY SCENARIO



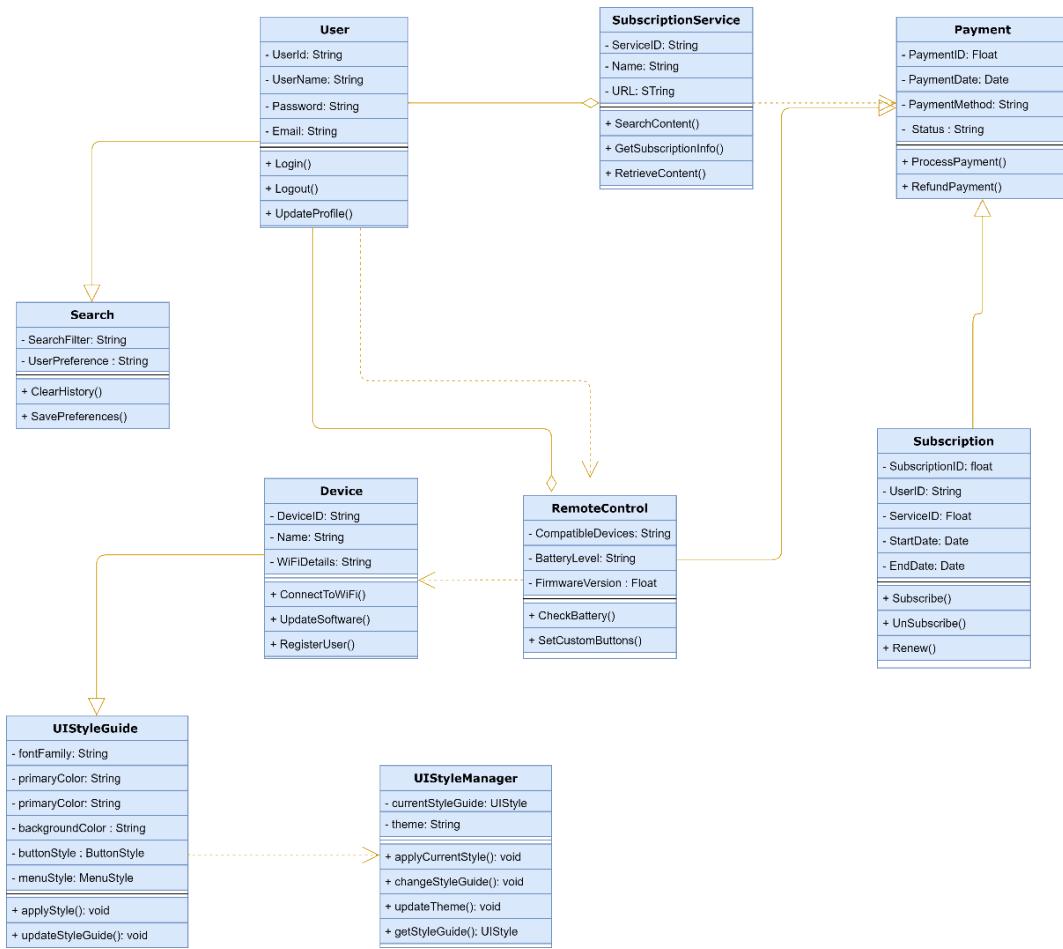
TACTICS

1. Uniform User Interface Design

Rationale: A unified user interface (UI) design makes it easier for people to comprehend and use the system.

Detailed Description of Implementation: Make a UI Style Guide that outlines the patterns of interaction and visuals. Utilize this guidance uniformly with every user interface.

Affected Artifacts: User Interface Storyboard and Class Diagram

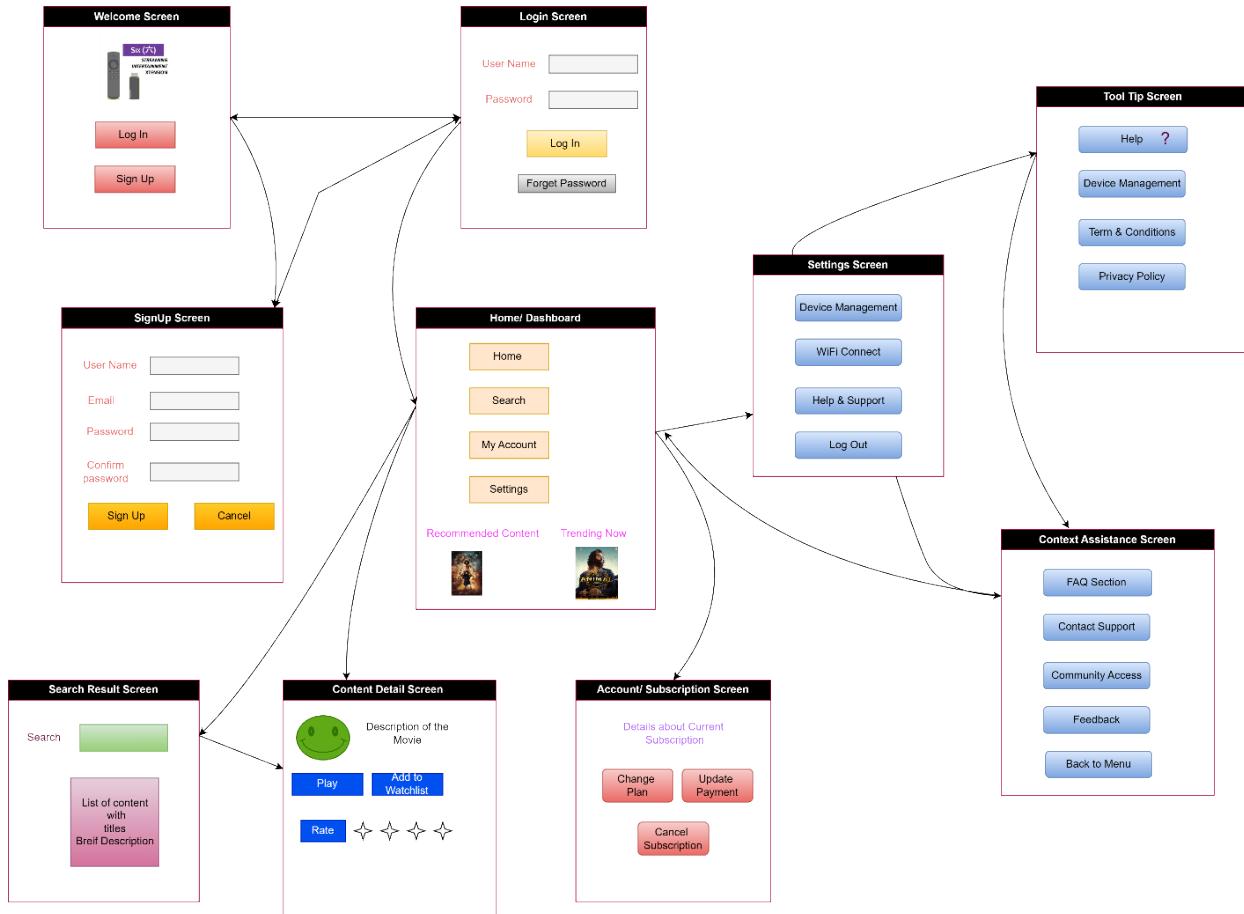


2. Contextual Guidance and Quick Tips

Rationale: Contextual assistance and tooltips improve user comprehension and lower the learning curve.

Detailed Description of Implementation: Include tooltips and contextual assistance in the user interface activated by hover events or user actions.

Affected Artifacts: User Interface Storyboard and Sequence Diagram.

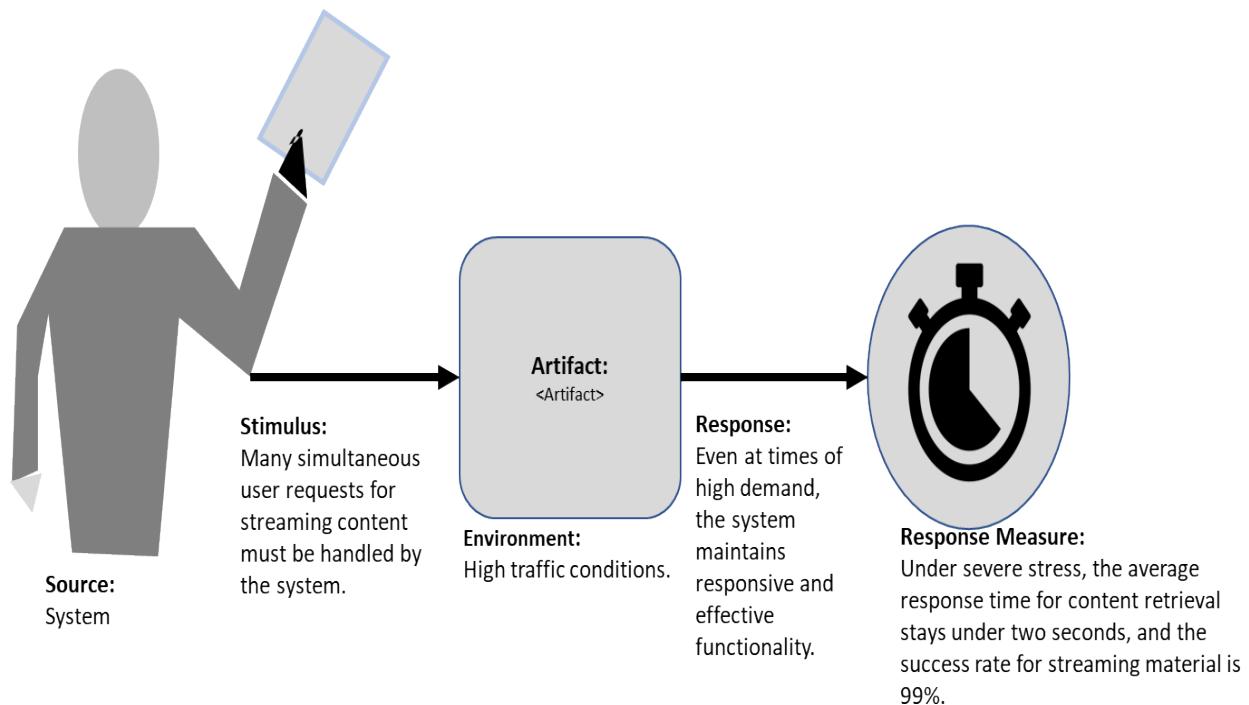


ACTIONS:

Go-Backs: revised and reviewed the storyboard for the user interface.

Looking Forward: Include the Class Diagram with the UI Style Guide.

PERFORMANCE SCENARIO:



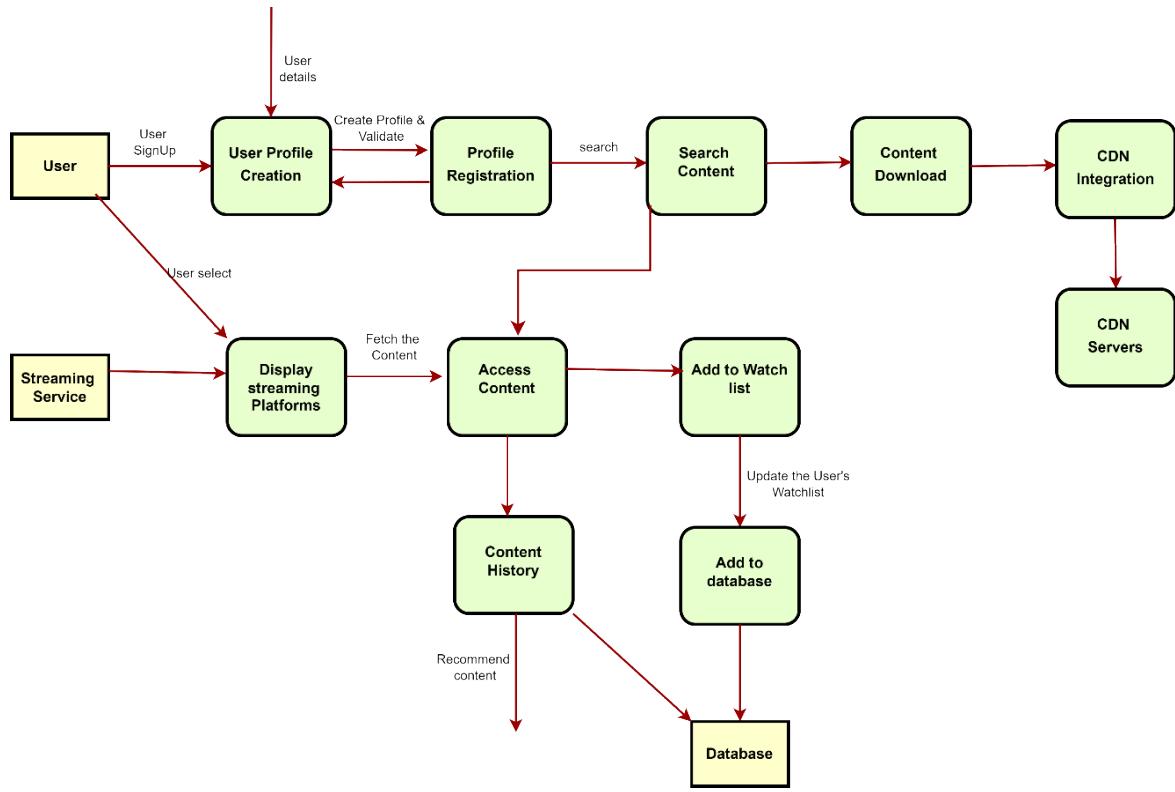
TACTICS

1. Integration of Content Delivery Networks (CDNs)

Rationale: By distributing material globally, **CDN integration** lessens the strain on the primary server and speeds up response times.

Detailed Description of Implementation: Use a CDN integration to cache and deliver commonly requested material from user-proximate servers.

Artifacts: Sequence diagram, Data Flow Diagram.

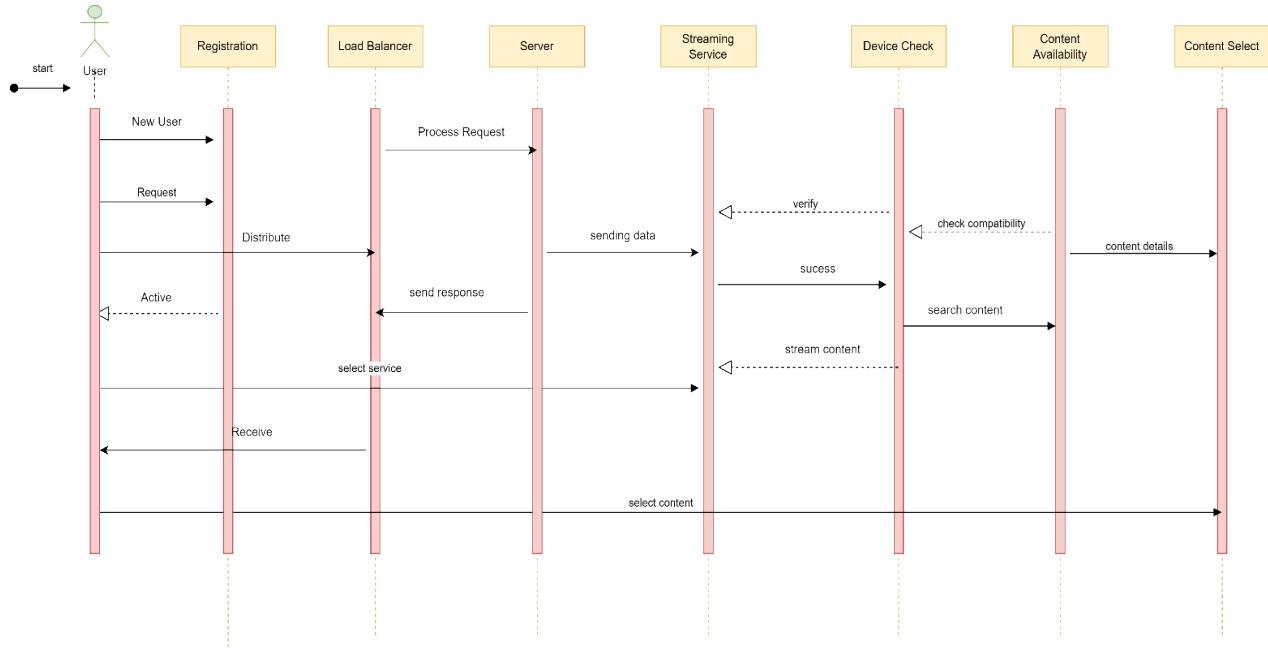


2. Balancing loads

Rationale: By distributing user requests across several servers, overload on a single server is avoided, guaranteeing steady performance.

Detailed Description of Implementation: To uniformly distribute incoming requests across multiple servers, implement load balancing algorithms.

Artifacts: Class and Sequence Diagrams



Take Actions:

Go-backs: revised and reviewed the diagram of data flow.

Looking forward: Adapt the Class Diagram to load balancing.

17. Config Management Overview & QA Metrics

Config Identification

An essential component of software system development and maintenance is software configuration management, or SCM. Throughout the software development lifecycle, it entails monitoring and regulating changes to the program, especially to its configuration items (CIs). This is a summary of the SIX project's SCM:

1. Configuration Items (CIs)

Hardware: Custom remote and an HDMI stick.

Software components: device firmware and SIX applications.

Documentation: API documentation and user guides.

Configuration files: preferences and settings.

Database schemas: Details about subscriptions and user accounts.

External interfaces: Third-party service APIs (such as those offered by streaming services and IMDB).

2. Versioning Identification Strategy

a) Software Versioning: For the SIX application, use a three-part versioning strategy (major, minor, patch).

- Patches are used to correct bugs, major version upgrades are for major releases, and minor versions are for feature additions.
- For traceability, include commit hashes or build numbers.

b) Versioning of Firmware:

- Apply a comparable versioning method to the firmware of the HDMI stick.
- Verify that the firmware and application versions are compatible.

c) Versioning of Documentation:

- Keep the versioned documentation up to date with the releases of firmware and software.
- To prevent confusion, make sure the documentation version is clearly indicated.

d) Versioning of Hardware:

For simple identification, give hardware parts (such an HDMI stick or a customized remote) version numbers or codes.

3. Traceability

Establish traceability between test cases, code modifications, and requirements.

Make use of tools to associate tickets or issues with certain configuration modifications.

4. Cooperation and Communication

Help the teams working on development, testing, and operations to communicate with one another.

To convey modifications and record decisions made by the CCB, use collaboration tools.

5. Backup and Recovery: Make regular backups of related data and configuration elements.

In the event of a failure, carry out recovery operations to return the system to a known state.

6. Change Control

Create a Change Control Board (CCB) to oversee and authorize modifications.

Utilize a version control system (like Git) to monitor modifications made to firmware, source code, and documentation.

For simultaneous feature releases and development efforts, use a branching model.

7. Build and Release Management

Automate build procedures to guarantee uniformity in various settings.

When code changes, use continuous integration (CI) technologies to start builds.

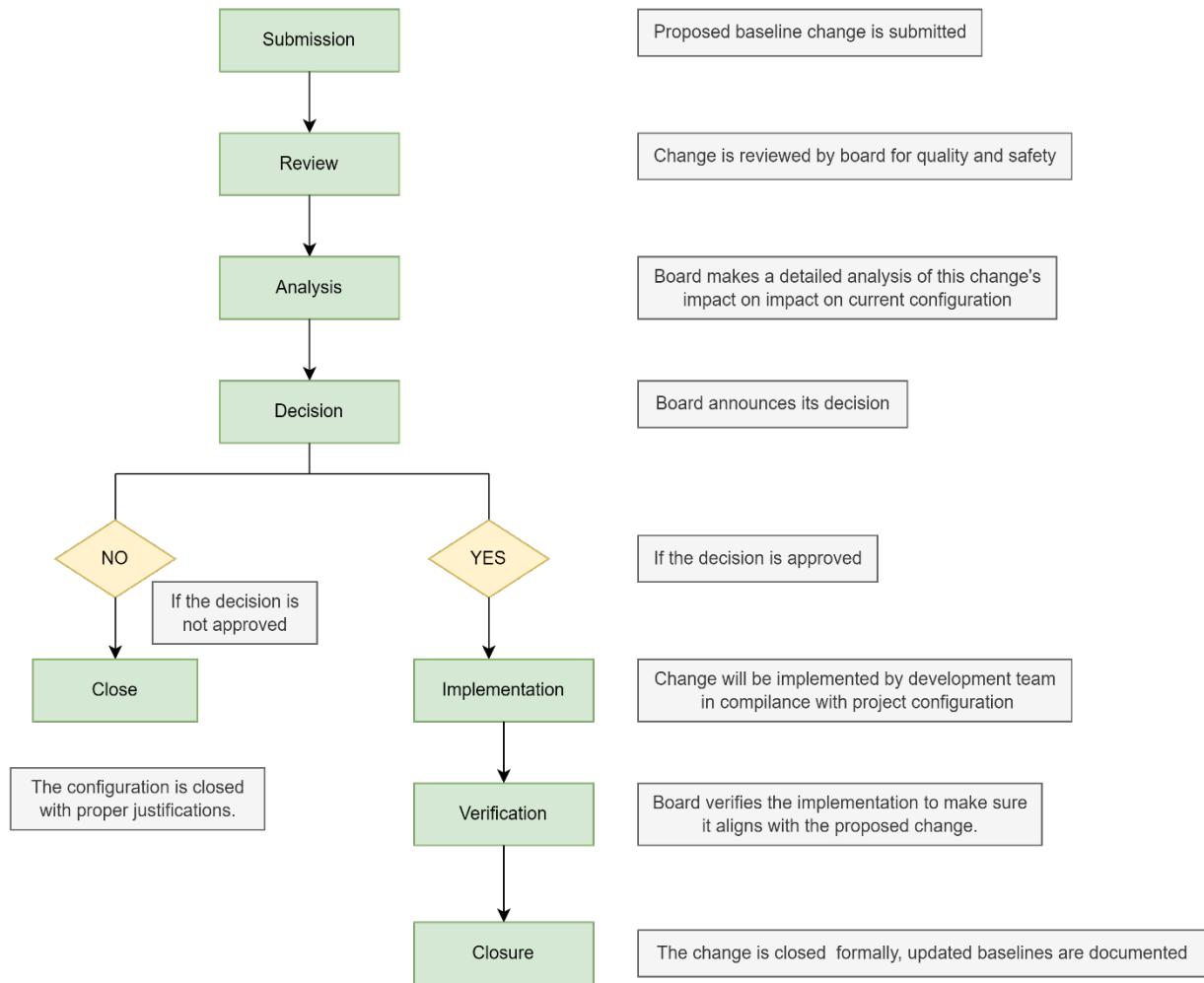
Continue to release content according to the 18-month time-to-market timetable.

Configuration Control

1. Baseline Change Board Workflow:

An organization's Baseline Change Board (BCB) is the governance or decision-making body in charge of examining and approving modifications to project baselines that have been set. A baseline is a snapshot of the project's essential elements (such as requirements, design, or configuration) at a certain moment in time that is used in project management. Baselines offer a point of reference for overseeing and managing modifications during the course of a project.

In order to make sure that suggested baseline modifications are in keeping with the project's objectives, specifications, and limitations, the Baseline Change Board is essential to the process. The board usually consists of important stakeholders and subject matter specialists who are well-versed in the goals of the project and the effects of any modifications that are suggested.



2. Key Stakeholders and Governance Strategy of Baseline Change Board

Stakeholders:

1. Product managers: They are essential in establishing the overarching vision and objectives of the SIX streaming service. They are in charge of setting feature priorities, making sure that features are in line with business goals, and making choices that have an effect on the product roadmap. Product managers collaborate closely with other key players to direct the process of development and fulfill the demands of users.
2. Development Teams: The technological components of the SIX streaming service are implemented by the Development Teams. This entails developing the firmware for the HDMI stick, the software components, and other technical fixes. They follow best practices and coding standards while collaborating with product managers to implement the envisioned features.
3. Quality Assurance Teams: QA Teams are responsible for guaranteeing the dependability and

efficiency of the SIX streaming service. They carry out testing, find anomalies and errors, and collaborate with development teams to resolve problems. QA Teams are essential to preserving the product's quality and guaranteeing a satisfying customer experience.

4. Configuration Management Team: Throughout the development lifecycle, the Configuration Management Team is in charge of overseeing and managing modifications to the configuration items. They are in charge of configuration status accounting, baseline management, and version control. This group makes certain that the product configuration is well-recorded, under control, and compliant with project specifications.

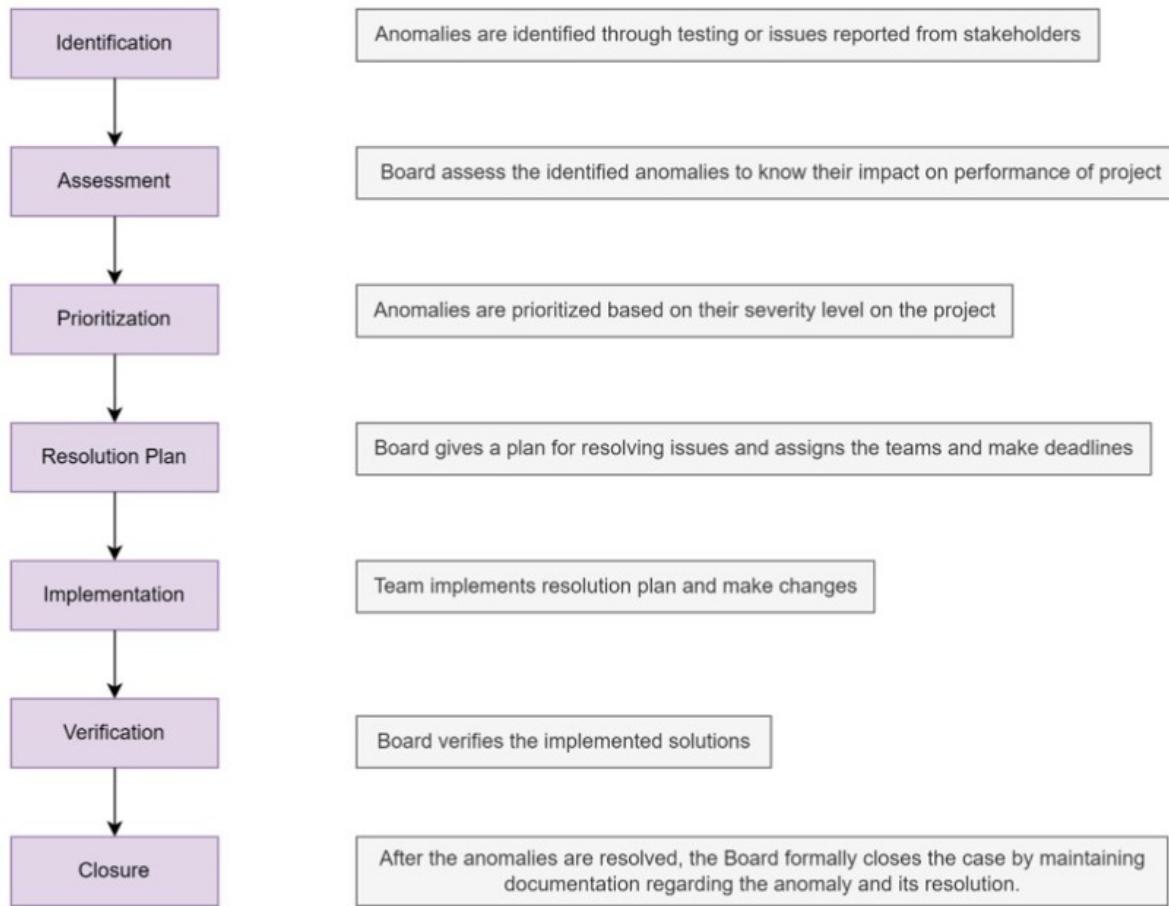
Governance:

1. Configuration Manager: In charge of the complete Software Configuration Management (SCM) process, the Configuration Manager is a crucial leadership position. They are in charge of creating and carrying out SCM best practices, protocols, and policies. The Configuration Manager makes sure that all teams follow set standards and that the configuration management plan is in line with the overall objectives of the project.

2. Baseline Change Board (BCB): Changes to the project baseline must be reviewed and approved by the BCB, a governing body. It is made up of personnel from multiple teams, such as the Configuration Management Team, Development Teams, Quality Assurance Teams, and Product Managers. The BCB makes sure that any modifications respect the stability and integrity of the baseline while also being in line with project goals.

3. Anomaly Resolution Board (ARB) Workflow

An organization or committee known as the Anomaly Resolution Board is in charge of identifying and resolving irregularities in a project, procedure, or system. Anything that differs from what is typical or expected is referred to as an anomaly. For a system to remain functioning and intact, anomalies must be found and fixed using a variety of techniques, including testing, monitoring, and analysis.



4. Key Stakeholders and Governance Strategy of ARB

Stakeholders:

1. Development Teams: These groups are in charge of designing and putting into action the SIX streaming service's technical elements. To deliver features and enhancements, they collaborate with other stakeholders and work on software design and code. The development teams play a pivotal role in transforming the project's concept into actual outcomes.
2. Quality Assurance Teams: QA Teams work to guarantee the dependability and caliber of the SIX streaming service. They work with development teams to fix problems, find and report bugs, and carry out testing. QA Teams are essential to preserving the integrity of the product and providing a satisfying user experience.
3. Product Managers: They oversee outlining the SIX streaming service's goals, plans, and features. Assisting the Development and Quality Assurance Teams closely, they prioritize

features according to business objectives and market requests. Product managers are essential in making sure that the finished product is in line with the overarching goals of the business.

4. Configuration Management Team: In the development process, the Configuration Management Team oversees overseeing and managing modifications to the configuration items. They are in charge of version control, accounting for configuration status, and making sure that the product configuration complies with project specifications. This group is essential to preserving traceability and consistency between versions.

Governance:

1. **Configuration Manager:** In charge of supervising the Software Configuration Management (SCM) process, the Configuration Manager is a leadership position. They make sure that every team follows set standards by defining and putting into practice SCM policies, procedures, and best practices. Control over the configuration components of the project depends on the Configuration Manager.
2. **Anomaly Resolution Board (ARB):** Tasked with addressing anomalies and problems found during development and testing, the ARB is a governing body. Representatives from Product Management, Testing, and Development are present. The ARB evaluates the degree and importance of abnormalities, organizes and supervises their remediation, and guarantees that the remedies meet project objectives.

CONFIG STATUS ACCOUNTING

CSA Strategy for As-Design Baseline

In order to make sure that the project stays within the defined baseline, the Configuration Status Accounting (CSA) technique for the As-Design Baseline is made to monitor and control changes in requirements, design documents, and program plans.

Key Attributes / Artifacts:

Record Versioning: Give requirements, design papers, and software plans version numbers.

To ensure traceability and rollback capabilities, manage changes using a version control system.

Establishing a baseline:

At significant turning points (e.g., early planning, design completion), establish baselines for program plans, requirements, and design documents.

To give a point of reference, each baseline should be precisely defined and documented.

Change Control:

Establish a change control procedure for suggested adjustments to requirements, design documents, and program plans.

Make sure that any modifications are vetted, authorized, and accurately recorded.

Audit Trail:

Ensure that every document has an audit trail that documents modifications, approvals, and dates.

Review the audit trail on a regular basis to make sure baseline standards are being followed.

Sample log

Item ID	Date	Version	Document	Changes
P001	2023-10-10	1.0.0	Program Plan	Initial Release
R105	2023-15-10	1.1.0	Requirement specification	Updated Section 3: User Authentication
D303	2023-20-11	2.0.0	Document design	Major Revision: Architecture(updated version)

CSA Strategy for Software Product Elements :

In order to guarantee a stable and dependable software product, the CSA strategy for software product elements concentrates on managing changes at several tiers, including hardware, operating systems, and applications.

Key attributes /Artifacts

1. Hardware Tier:

Versioning of the hardware attributes.

Documentation on the configuration of every piece of hardware.

Change control for hardware updates.

2. Tier of the Operating System:

Versioning of operating systems.

Update and patch the logs.

matrix of compatibility with hardware elements.

3. Applications Tier

Versioning of applications.

Release notes for every change to an application.

Monitoring dependencies for external libraries.

Sample log:

Item ID	DATE	Version	Component	Changes
H012	2023-10-10	1.0.0	HDMI Stick Firmware	Initial Release
H020	2023-15-10	1.2.0	HDMI Stick Firmware	Bug fix: Connectivity Issue
O129	2023-29-10	2.0.0	Operating system	OS upgrade: Version 2.0.0
A235	2023-21-10	1.5.0	SIX Application	Adding Feature: Search function

Version Control

We will utilize a version control system to oversee the development of SIX and monitor updates, problems, and teamwork. Git is a robust version control system that is frequently used. Your Git repositories can be hosted in a collaborative environment with services like GitHub or GitLab. Here's an example of how version control could be organized:

1. Strategy:

Master: The stable, production-ready version is referred to as the master.

Develop: An integration branch for future work in progress.

Feature branches: Before being merged into development, every new feature or enhancement is worked on in a different branch.

Release branches: are made in advance of a release to accommodate impromptu problem patches.

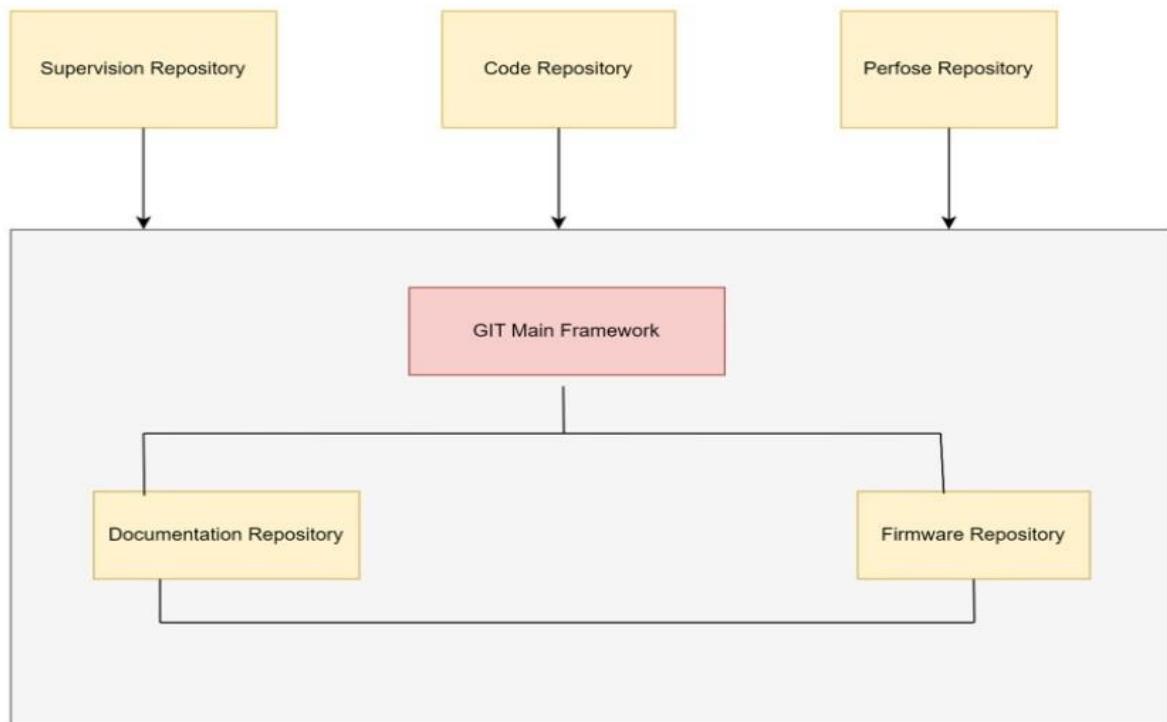
2. Commit Messages:

To improve readability and traceability, commit messages should adhere to a clear, uniform format.

Tags: For versioning milestones like v1.0, v1.1, etc., use tags.

Pull Requests:

Before integrating changes into the develop or master branches, conduct a code review using pull requests.



Configuration Auditing Checklist

Configuration Files:

Check each software component's configuration file integrity.

Examine any unapproved modifications or alterations to the default setups.

Access Control:

Make sure that the access restrictions are set up correctly for databases, APIs, and other resources that are sensitive.

Examine the settings for authentication and user rights.

Network Configuration:

To find any weaknesses or incorrect setups, audit network configurations.

Verify network policies, port setups, and firewall settings.

Security Configurations:

Make sure the encryption settings are set correctly for both data at rest and in transit.

Make sure that robust encryption algorithms and safe protocols are being used.

Integrities with Third Parties:

Examine the settings for third-party APIs and services.

Make sure credentials and API keys are handled safely.

Configuration for Logging and Monitoring:

Verify that the settings for logging and monitoring are audited to make sure pertinent events are recorded.

Check to see if notifications are set up for important occurrences.

Configuration for Backup and Recovery:

Make sure that the configurations for backup and recovery are established and verified.

Check the backups' consistency and frequency.

Major Events and Frequency:

Regular Audits: To proactively find and fix any possible problems, perform periodic configuration audits on a quarterly basis.

Major System Audits: Configuration audits should be carried out both before and after major system releases or modifications.

Security Incidents: In the event of a security incident or breach, an audit should be carried out right away.

Key Metrics for Software Project / Process / Product:

Project Metrics:

Metric: Code Churn Rate (Added, Modified, and Deleted Lines of Code)

Frequency: Regularly observed and reported on a weekly basis.

Rationale: Shows how stable the codebase is and how much work is still being done on it.

Process Metrics:

Metric: Lead Time for Changes (Duration between Concept and Production Deployment)

Frequency: Tracked for every cycle of release.

Rationale: Indicates how quickly new features or modifications are put into production as well as how effective the development process is.

Product Metrics:

Metric: User Satisfaction Score (obtained via user feedback or surveys)

Frequency: Every month is collected.

Rationale: Offers information on the general level of customer satisfaction with the product as well as its usability and user experience.

Thresholds and Programmatic Responses:

1. Rate of Code Churn:

Threshold: An abrupt increase in churn of code.

Response: Launch an automated alarm for prompt examination, enlisting the development group to evaluate the origin and consequences.

2. Lead Time on Modifications:

Threshold: A significant lead time increase is the threshold.

Response: Start a retrospective investigation to find inefficiencies and bottlenecks in the development process.

3. User Contentment Index:

Threshold: A steady decline in user satisfaction is the threshold.

Response: Will start a comprehensive investigation of usability and feedback, and work with the UX/UI team to resolve any issues found.

Key Software Quality Assurance (QA) Metrics:

1. Density of Defects:

Metric: Defect count per code unit (per KLOC, for example).

Frequency: During testing phases, observed once a week.

Rationale: Offers information about the codebase's quality and the efficiency of the testing procedures.

2. Test Coverage:

Metric: The proportion of code that is tested automatically.

Frequency: Recorded throughout testing stages and tracked with every code commit.

Rationale: Guarantees that crucial components of the codebase undergo sufficient testing.

Thresholds and Programmatic Responses:

1. Density of Defects:

Threshold: A significant rise in defect density is the threshold.

Response: Send an automated alert to the QA and development teams, requesting prompt investigation and resolution.

2. Test Extent:

Threshold: Dipping below an established level.

Re: To alert the development team, halt the deployment process until test coverage increases.

18. System Level Test Plan

Test Case 1:

Test Case: USER REGISTRATION – Successful Registration		
Goal: Verify that the user can register successfully to SIX application.		
Impacted Modules: User Registration Module, Database		
Pre-Conditions/Dependencies:		
1. User has access to a valid email address.		
2. The user owns a working SIX device.		
3. The internet is accessible through the device.		
Test Sequence:		
Step	Expected Result	Actual Result
1. Enter valid user details (name, email, password) and submit registration form.	Leave mandatory fields blank and submit the form.	PASS
2. Attempt to register with an already registered email	Error message indicates the email is already in use	PASS

Final Conditions: User is registered successfully with a unique account.

Test Case 2:

Test Case: Search for Movie/Actors**Goal:** To verify that users can search for movies or actors on SIX and receive relevant results.**Impacted Modules:** Search Functionality, IMDB Integration**Pre-Conditions/Dependencies:**

1. user is logged in and on the main search screen.
2. The device is connected to the internet.

Test Sequence:

Step	Expected Result	Actual Result
1. Search for a known movie title.	Relevant movie information is displayed.	PASS
1. Search for a known actor's name.	Relevant actor information is displayed.	PASS

Final Conditions: Users can successfully search for movies or actors**Test Case 3:****Test Case: Select Content****Goal:** To verify that users can select and access content from various streaming providers.**Impacted Modules:** Content Selection, Streaming Providers Integration**Pre-Conditions/Dependencies:**

1. User is logged in and on the content selection screen.
2. The device is connected to the internet.

Test Sequence:

Step	Expected Result	Actual Result
1. Browse through available content categories and select a movie.	Movie details and streaming options are displayed.	PASS
2. Attempt to access content from a provider without a valid subscription	User is prompted to subscribe to access the content.	PASS

Final Conditions: Users can successfully select and access content from subscribed providers.**Test Case 4:**

Test Case: Manage Subscription**Goal:** To verify that users can manage their subscriptions through the SIX platform.**Impacted Modules:** Subscription Management, Online Account**Pre-Conditions/Dependencies:**

1. User is logged in and on the subscription management screen.
2. The device is connected to the internet.

Test Sequence:

Step	Expected Result	Actual Result
1. View the list of current subscriptions.	All active subscriptions are displayed.	PASS
2. Cancel an existing subscription	Confirmation message is displayed, and subscription is canceled.	PASS

Final Conditions: Users can successfully manage their subscriptions through the SIX platform.**Test Case 5:****Test Case: Content Playback****Goal:** To verify that users can play, pause, and navigate through content using the custom remote.**Impacted Modules:** Content Playback, Custom Remote**Pre-Conditions/Dependencies:**

1. User is logged in and has selected content for playback.
2. The device is connected to the internet.

Test Sequence:

Step	Expected Result	Actual Result
1. Use the custom remote to play content	Content starts playing without issues	PASS
2. Pause the content using the custom remote.	Content pauses and the playback position is retained	PASS

Final Conditions: Users can successfully control content playback using the custom remote.**Test Case 6:**

Test Case: Database Check		
Goal: To verify the integrity and functionality of the database.		
Impacted Modules: Database		
Pre-Conditions/Dependencies:		
1. The database is in a stable state.		
2. The device is connected to the internet.		
Test Sequence:		
Step	Expected Result	Actual Result
1. Insert new user registration data into the database.	User data is successfully stored in the database.	PASS
2. Retrieve movie and actor information from the database.	Relevant information is retrieved accurately	PASS

Final Conditions: The database functions correctly and stores/retrieves data as expected.

Traceability Table:

Requirement	Use case	Test case
FR1.1	UC1	TC1
FR1.2	UC1	TC1
FR1.3	UC1	TC1
FR1.4	UC1	TC1
FR2.1	UC2	TC2, TC3
FR2.2	UC2	TC2,TC3
FR2.3	UC3	TC2
FR2.4	UC3	TC2
FR3.1	UC4	TC6
FR3.2	UC4	TC6
FR3.3	UC4	TC6
FR4.1	UC4	TC5
FR4.2	UC4	TC5

FR4.3	UC4	TC5
FR4.4	UC4	TC5
FR6.1	UC5	TC4, TC6
FR6.2	UC5	TC4, TC6
FR6.3	UC5	TC4, TC6
FR6.4	UC5	TC4, TC6

19. Glossary

20. Bibliography

