Project System Design Document

Project Title: Bingescape (Web Streaming Service)

Date: 27.10.2021

Subject: Software Engineering Lab

Contributors:

Ashish Nehra (2019IMT-022)

Kaushal Waghela (2019IMT-049)

Uditansh Patel (2019IMT-108)

INDEX

1. Introduction

Bingescape is a Web based application and it is developed to allow our customers to watch a wide variety of award-winning TV shows, movies, documentaries, and more on thousands of devices, without the hassle of managing multiple subscriptions. Also, our product allows the customer to binge watch content from several other streaming services, all under one app, hence reducing the struggle of wasting time to find the perfect content to watch. With Bingescape, you can enjoy unlimited viewing of our content without having to watch a single commercial.

2. Design Considerations

This section describes many of the issues that are needed to be able to addressed or resolved before embarking on a complete design solution. This document is based on the version v1.0 as in SRS document. There is a need for reference in case any part is not understood or felt incomplete.

2.1 Assumptions

This Bingescape design makes several assumptions about the software and hardware requirements as are in the SRS. All the environmental operating requirements of both the user interface and the database can be found in the Bingescape requirements. Both the database and the user application make the following assumptions about the operating environment. The system can be described by the operating requirements associated with this document and in the SRS. The system application in execution will have the necessary resources availed as required. This entails sufficient memory and permanent storage space and the adequate CPU for the application. The application makes the following assumptions about its operating environment.

2.2 Constraints

The Bingescape shall be a web-based system. This system shall be developed using HTML, CSS, JavaScript, ReactJs, Firebase.

2.3 Design methodology

The waterfall model will be used as the best language for this system type. This is because the waterfall model is suitable visualize, clarify, construct and document features of the system.

The design will take the following approach:

1. **Requirement Gathering and analysis**
2. **System Design** – The requirement specifications from first phase are studied in this phase and the system design is prepared.
3. **Implementation**
4. **Integration and Testing**
5. **Deployment of system**
6. **Maintenance**

2.4 System environment:

The requirements for the system architecture of the Let's Agrii are system:

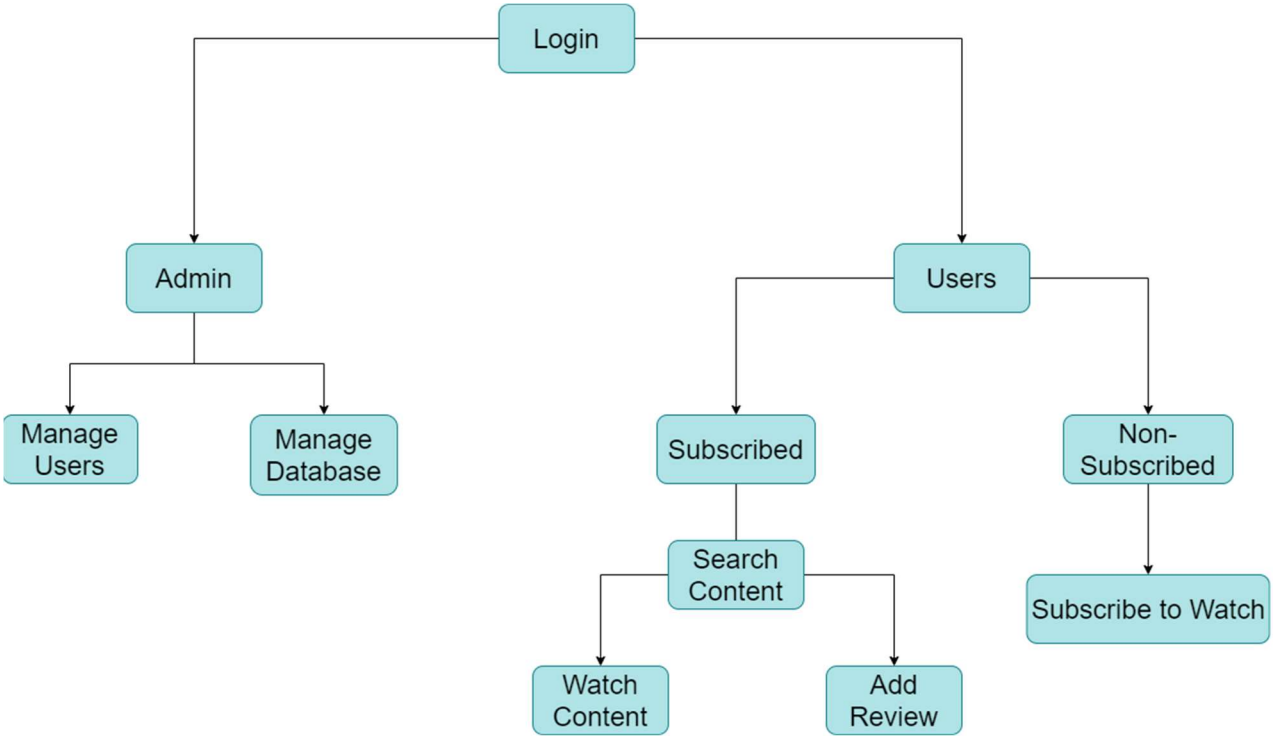
- Scalability: The architecture is scaled horizontally, across multiple servers, and across multiple regions.
- Security: The architecture of the project uses as minimal a amount of code as possible. Most of the back-end pieces are hidden away for security purposes.

3. Architecture

This section describes many of the issues that are needed to be able to addressed or resolved before embarking on a complete design solution. This document is based on the version v1.0 as in SRS document. There is no need for reference in case any part is not understood or felt incomplete.

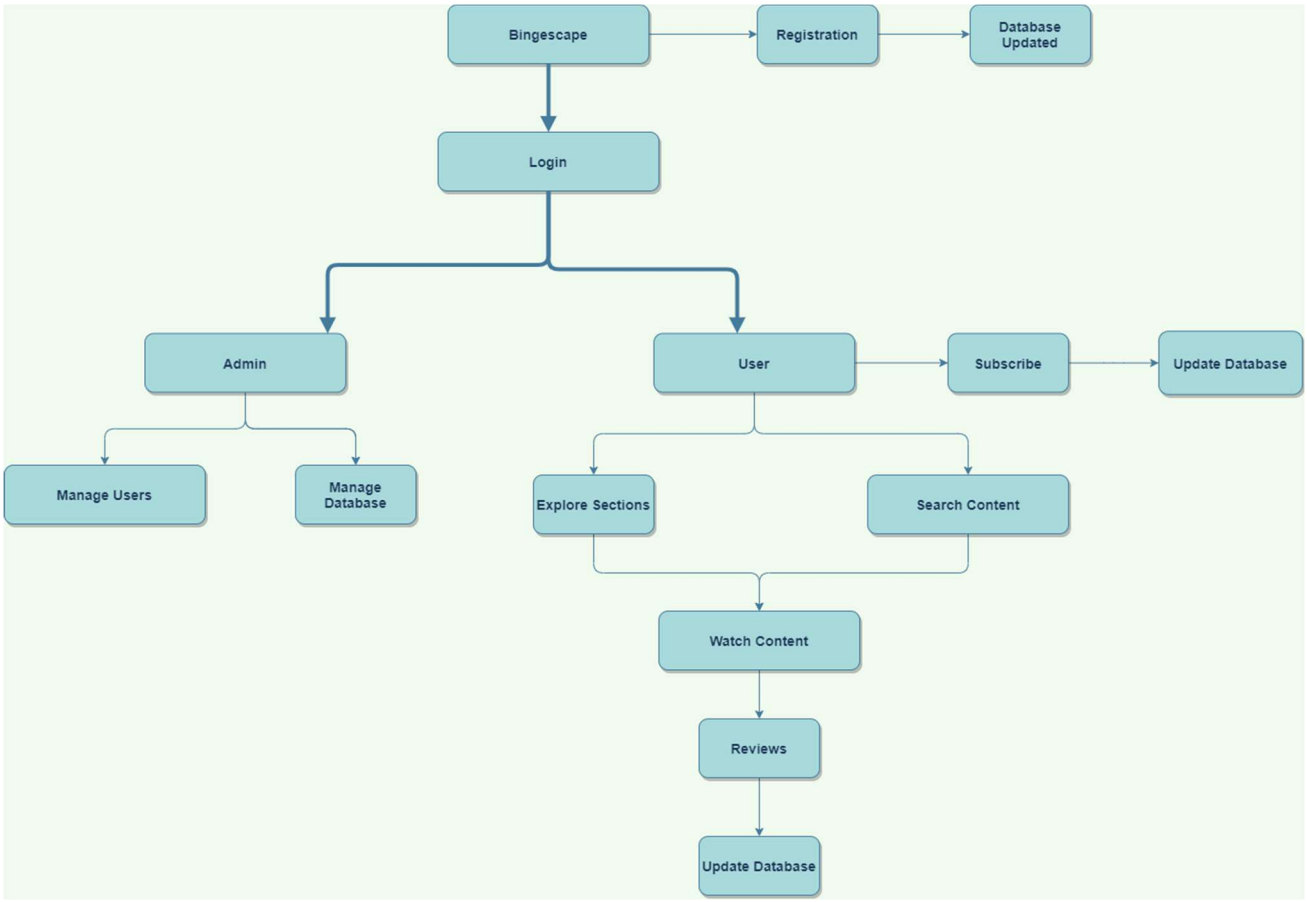
3.1 System design

Systems design for Bingescape involves a step-by-step approach to the design of the whole project which may have a bottom-up or top-down approach. However, the process is systematic wherein takes into account all the related variables of the system created(from the architecture to the required hardware and software, right down to the data and how it travels and transforms throughout its travel through the system). Systems design then overlaps with systems analysis, systems engineering, and systems architecture. In this phase, the complex activity of system development is divided into several smaller sub-activities, which coordinate with each other to achieve the main objective of system development.



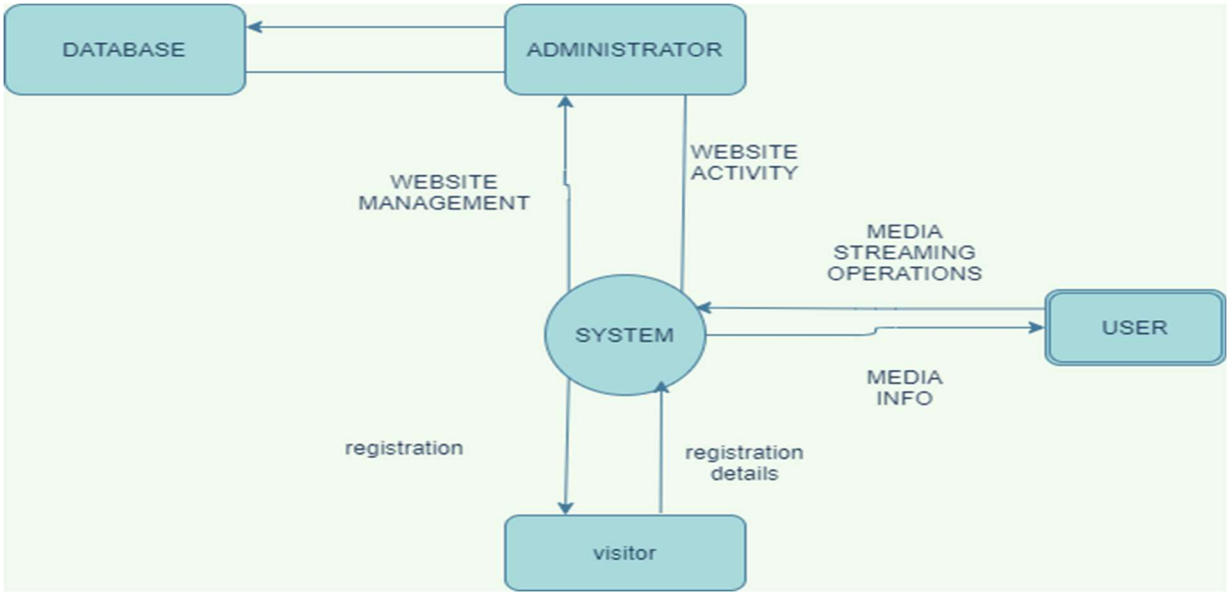
3.2 Functional decomposition tree

The main functions of the system are decomposed into smaller sub functions or sub-modules and further. The System shall take place following structure of organization after implementation. The decomposition is stable and functions should be made highly cohesive.



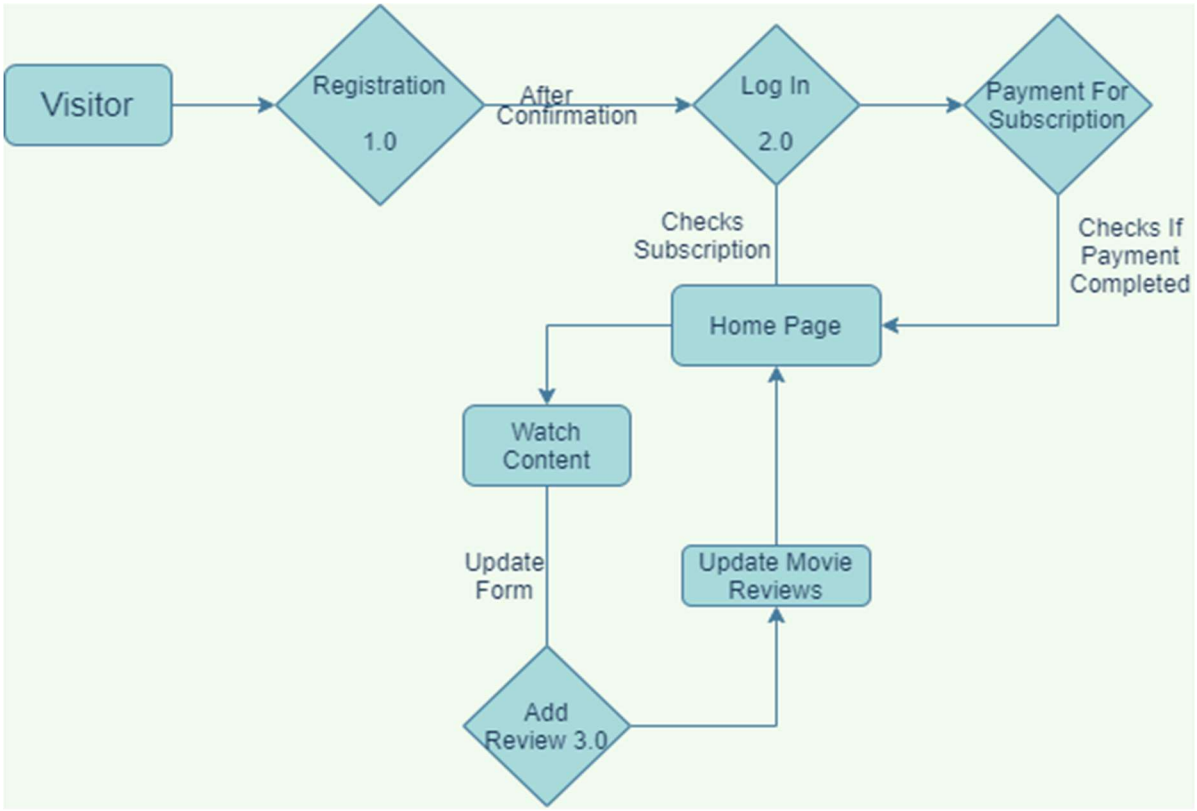
3.3 Context diagram

Context diagram describes the main actors interacting with the system.



3.4 Data flow diagrams

User Data Flow Diagram



3.5 Data dictionary

Table 1: Admin

Field	Type	NULL	Default
uname	varchar(30)	NO	None
Password	varchar(30)	NO	None
Name	varchar(30)	NO	None

Table 2: Net Banking

Field	Type	NULL	Default
uname	varchar(30)	NO	None
password	varchar(30)	NO	None
bank	varchar(30)	NO	None

Table 3: User

Field	Type	NULL	Default
pid	varchar(11)	NO	None
name	varchar(20)	NO	None
email	varchar(50)	NO	None
Mob	Varchar(10)	NO	None

Table 4: Movies:

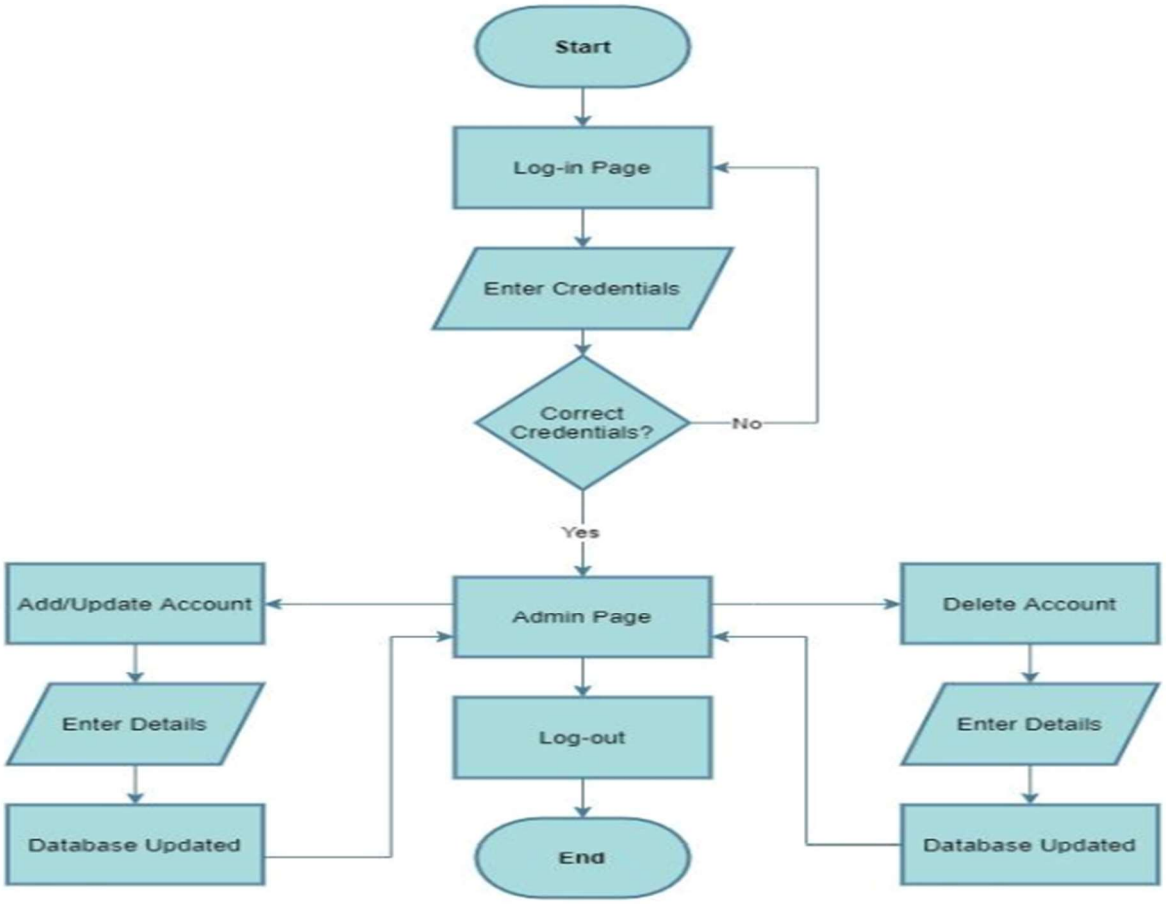
Field	Type	NULL	Default
mid	varchar(11)	NO	None
name	varchar(20)	NO	None
category	varchar(10)	NO	None
year	varchar(4)	NO	None

Table 5: Subscription

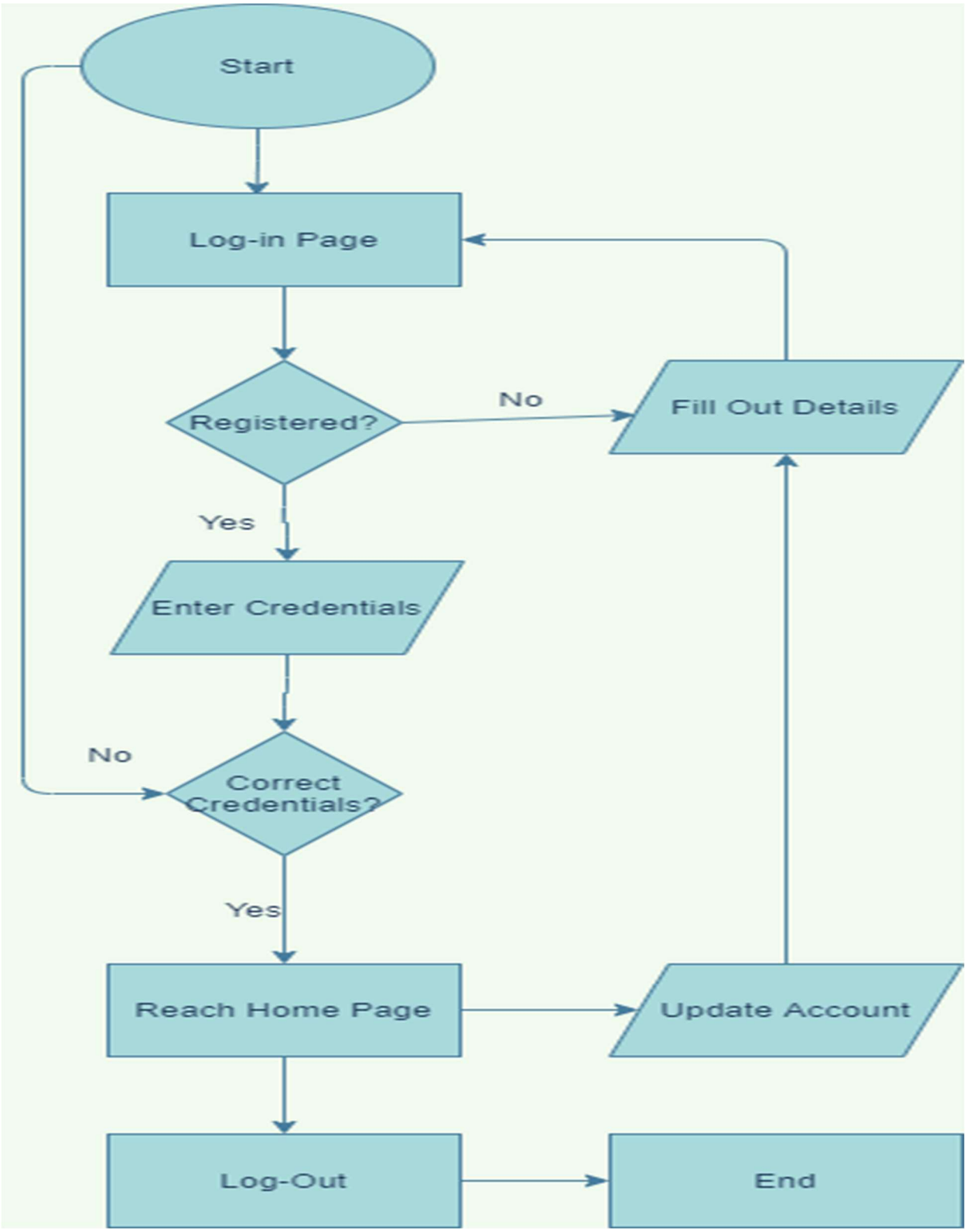
Field	Type	NULL	Default
SId	varchar(11)	NO	None
Sname	varchar(20)	NO	None
Duration	varchar(7)	NO	None
Price	varchar(5)	NO	None

4. Component design
4.1 Activity diagram

Admin Activity Diagram



User Activity Diagram



5. User Interface Design

The User Interface for Bingescape is designed according to standard UI design principles.

- **The simplicity principle:** Users can navigate through the website with relative ease. The UI is minimalistic and easy on the eye.
- **The visibility principle:** Every function is clearly visible to the user and can be easily accessed.
- **The structure principle:** Elements on the page are organized such that related elements are grouped together, and unrelated elements or/and a group of elements are kept separate, at a distance from each other.
- **The feedback principle:** The design ensures and validates users' actions and selections.
- **The reuse principle:** Certain names and elements have been reused to increase the ease of use and reduce ambiguity.

The user interface will consist following main screens, a login screen which will consist of a user dialogue text box, and three labels for data input. The login screen will be used to authenticate the user to the system.

The sample user interface forms and screens that the user will interact with include:

- **The "Landing"** page will be the first page the user sees. Users can either log in to the web application or head to the **"Home/Dashboard"** page if already signed in. Users can create a note without logging in as well.
- **The "Login/Register"** page is the webpage where existing users can log in and new users can register themselves and make an account.
- **The "Home/Dashboard"** page is the link the user is brought to after logging in.
  From here, the user can access all functionalities of the website: finding the perfect shows/movies to watch, search command, and also to get answers to their queries through the FAQ page or contact us through our About Us section.
- **The "User Profile"** page shows all the info about the users and the shows/movies they watch through our application and also to effectively manage their subscription. This page is the effective dashboard for the user where they have access to all other functionalities.
- **The "Payment Page"** This page is the section for payment and processing of subscription confirmation. It is linked to a third-party payment vendor which the user has no concern about.