Dire Dawa University Institute of Technology

School of Electrical and Computer Engineering

Computer Engineering Chair

Semester Project Report

Project Title: AXUM Web Based Movie Streaming

Group Members Name	**ID**
1. Kidus Mikael	TEMID9002
2. Biruk Gero	TEMID1409
3. Abiy Abebe	TEMID1378
4. Ayelign Dawit	TEMID1325
5. Mulugeta Sahile	TEMID1374
6. Munir Jemal	TEMID4087

Adviser: Mr. Haftu M.

March 8, 2023 GC

Dire Dawa, Ethiopia

Declaration

We, the undersigned, declare that this project is our own original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the project have been fully acknowledged.

____________________________	____________________________
Student Name	Signature
____________________________	____________________________
Student Name	Signature
____________________________	____________________________
Student Name	Signature
____________________________	____________________________
Student Name	Signature
____________________________	____________________________
Student Name	Signature
____________________________	____________________________
Student Name	Signature

This Project has been submitted for examination with my approval as a university advisor.

____________________________	____________________________
Project advisor	Signature

Acknowledgement

Firstly, we would like to express our sincere gratitude to our advisor Mr. Haftu for the continuous support, for his patience, motivation, and immense knowledge. And we would like to give special thanks to the Department of Electrical and Computer Engineering of Dire Dawa University Institute of Technology for giving us opportunities and support to do the research

Abstract

In this digital era, the entertainment industry is rapidly growing. Movies are no longer rented or sold; instead, we can watch as much as we want, anytime, anywhere, on any internet-connected screen. This technology is very important because it has made entertainment more entertaining and easier. Putting this in consideration, we built AXUM, a movie streaming web app that is simple and easy to use. AXUM was developed using a combination of programming languages, including Django(python), HTML, CSS, JavaScript and SQLite. This documentation provides a comprehensive guide to AXUM, a movie streaming web application. It covers all aspects of the platform, including the user interface, profile management, movie selection, and personalization options. The document outlines technical requirements and supported devices, as well as provides information on the security measures and privacy policies in place to protect user data. Additionally, it includes troubleshooting and support information to ensure a seamless movie streaming experience for users. In conclusion this document serves as a valuable resource for users to fully understand and utilize the capabilities of AXUM. The use of programming languages like Django, HTML, CSS, and SQLite makes AXUM a robust and reliable platform for streaming movies. By providing an easy-to-use interface and a vast selection of movies, AXUM is designed to provide an unparalleled movie streaming experience for users.

Key Words: Django, user interface, entertainment industry

Abbreviations

HTML - Hypertext Markup Language.

CSS - Cascading Style Sheet

SQL - Structured Query Language.

UML - Unified Modeling Language

VOD - video on demand

Table of Contents

List of Figures

List of Tables

Chapter One

Introduction

1.1 Background

Online Streaming is becoming more and more popular every day. It makes access to digital content, whether it be videos or audio, much faster and easier. While streaming content, the user is not required to download that content but can watch it online. What the user requires is a good Internet connection and a good service provider. This is where AXUM comes in. AXUM is a web application that provides access to unlimited content for free. Video on demand (VOD) is a programming system which allows users to select and watch/listen to video or audio content such as movies and TV shows whenever they choose, rather than at a scheduled broadcast time the method that prevailed with over-the air programming during the 20th century. Television VOD systems can stream content through either a set-top box, a computer or other device, allowing viewing in real time on a device such as a computer, digital video recorder or portable media player for viewing at any time [1].

1.2 Statement of problem

Traditional method entertainment is very difficult and time consuming. Users used to buy and rent CDs, DVDs. Go through a lot of process to get the CDs they want even wait for others to finish watching the movie they want to watch that is time consuming and tiring.

Movie enthusiasts today face a common problem - the difficulty in finding and accessing high-quality movies online. Streaming services are often fragmented and offer a limited selection, making it hard to find the perfect movie to watch.

The online movie streaming industry in Ethiopia has been slow to develop, leaving movie enthusiasts facing a common problem: finding the right platform to enjoy the best Amharic movies. With limited options available, it can be difficult for viewers to access high-quality movies in their preferred language. Axum aims to solve this problem by offering a comprehensive library of movies and TV shows, with a focus on Amharic content. By doing so, Axum provides an exceptional user experience for Ethiopian movie enthusiasts, addressing the need for a reliable platform to stream their favorite movies.

1.3 Objectives

1.3.1 General objective

The general objective of the project is to develop web-based movie streaming application.

1.3.2 Specific objectives

The specific objectives of our project include:

- ✓ Identifying problem of the existing system.
- ✓ Gathering required information for proposed system.
- ✓ Analyzing the gathered information.
- ✓ Designing the proposed system.
- ✓ Implementing the system.
- ✓ Testing the system.

1.4 Significance of the project

AXUM has much significance in terms of reducing time wasted by users to find and buy and rent the movie they want to watch, save the resources used to produce CDs and DVDs since we don't need them and improve users experience through digital entertainment.

1.5 Scope of the project

The area covered in our project include the following:

- ✓ The project provides a user registration system to enable users to enjoy movies.
- ✓ The project provides a system that allows users to create their own profile and receive a personalized experience.
- ✓ The project provides a profile specifically for children, which includes movies that are rated for kids. Parents can rest assured that the content their children access is appropriate.
- ✓ The project provides an administrator system that enables the admin to take actions on website or system users.

1.6 Limitation of the project

Our project is limited to sum factors specially when it comes to payment system, we didn't deploy any payment system since we found many difficult circumstances.

Chapter Two

Literature Review

2.1 introduction

The literature review is an essential aspect of the research process for any software development project, including the development of Axum, a movie streaming web application. In this section, we will provide a comprehensive review of the relevant literature on topics related to movie streaming web applications and their development. Specifically, we will focus on scholarly sources such as academic journals, conference proceedings, and other published materials that are relevant to the development of Axum. This review will help to establish the current state of knowledge on the topic and provide insights into best practices, key challenges, and emerging trends in the field of movie streaming web application development. Additionally, this review will help to identify gaps in the existing research and inform the development of research questions or hypotheses for our project. Overall, the literature review will serve as a crucial foundation for the development of Axum and contribute to our understanding of the broader context in which our project is situated.

2.2 Different related works

For years, Netflix has been known for its innovation and early-entry into different segments of the entertainment and media industry. Netflix began as the first DVD-by-mail service, but soon became the first in other categories as well including a DVD subscription plan and then a streaming service. Over 50 percent of adults living in the United States currently have a Netflix subscription. Netflix has been a front runner in the streaming industry for years, but their success has attracted new competitors to join the streaming industry. Disney+ is a new entrant that has taken the streaming industry by storm [1].

The movie streaming web app industry has experienced significant growth in recent years, with the rise of on-demand video services such as Netflix, Amazon Prime Video, and Disney+. This growth has been driven by advances in technology, changing consumer habits, and a growing demand for convenient and accessible entertainment options [2].

Movie streaming services have experienced explosive growth in recent years, with the number of subscribers to these services increasing dramatically. This growth has been driven by several factors, including the increasing availability of high-speed internet, the growth of smart TVs and connected devices, and the increasing demand for on-demand entertainment options. Research has shown that the convenience and accessibility of movie streaming services has made them increasingly popular with consumers, and that this trend is expected to continue in the coming years [2].

Digital Entertainment Group (DEG) reported that subscription-based video streaming has become the second largest revenue source for the home entertainment industry by delivering more than $5 billion to studios in 2015 (DEG 2016). Subscription-based video streaming platforms, such as Netflix and Hulu, allow subscribers to watch videos on demand on computers and mobile devices by paying a monthly subscription fee. Various contents are available on these platforms, including TV shows, recent blockbusters, classic films, small-budget independent films, and original programs produced by the platforms themselves. Video streaming services enjoyed a sustainable growth rate of 25% to 33% from 2013 to 2015 (DEG 2016; Variety 2016). Unlike the burgeoning streaming media, the largest revenue source, the sales of physical discs, contributed $6.1 billion in 2015 but declined by over 10% every year from 2013 to 2015 [3].

2.3 Summary

The literature review for the development of Axum, a movie streaming web application, provides a comprehensive overview of the relevant literature on topics related to movie streaming web application development. The document discusses the growth of movie streaming web apps and the shift in the home entertainment industry towards subscription-based video streaming services. It highlights the early entry of Netflix into the industry and the success of its streaming service, which has attracted new competitors like Disney+. The document also mentions the factors driving the growth of movie streaming services, including advances in technology and changing consumer habits. Additionally, it reports that subscription-based video streaming has become the second largest revenue source for the home entertainment industry and has surpassed sales of physical discs.

Chapter Three

Methodology

3.1 Introduction

In this chapter we will make a detail analysis of the project starting from project methodology, system design, requirements analysis, and there will be diagrams that will be used to depict the overall system functionalities. These include the use case diagram, sequence diagram and others. There will also be description of the functional and nonfunctional requirements of our system later in this chapter. Now let us start by project methodology.

3.2 Software System Requirements and Specification

This part of the document describes the features and behavior of our project. It describes the functional and nonfunctional requirement of the project

3.2.1 Functional Requirements

Our system provide service for customer and administrator. Functional requirements for customer and administrator are described as follow.

3.2.1.1 Functional Requirement for customers

- ✓ The system must allow customers to register just using their email address and create a password.
- ✓ The system must allow registered customers to log in to their account by entering their email and password they created.
- ✓ The system must allow customers to choose their profile if they have one and create their own profile if they don’t have one yet
- ✓ The system must allow customers to choose from all the available movies
- ✓ The system must allow customers to read each movies description in order to help them know what they are choosing
- ✓ The system must allow customers to watch the movie they have chosen

3.2.1.2 Functional Requirement for administrator

- ✓ The system must allow admin to log into their account using their username and password.
- ✓ The system must allow admin with permissions to create any users, users profile and movies.

- ✓ The system must allow admin with permissions to read any users, users profile and movies.
- ✓ The system must allow admin with permissions to update any users, users profile and movies.
- ✓ The system must allow admin with permissions to delete any users, users profile and movies.

3.2.2 Non-Functional Requirements

The following lists of nonfunctional requirements are expected from the system:

- ✓ Scalability: The system must be compatible with any environment.
- ✓ Security: should allow login to only authorized users.
- ✓ Reusability: Ability of an item that allows it to be used repeatedly unlike a disposable item.
- ✓ Accuracy: proposed system will be better due to reduction of error. All operation can be done correctly and it ensures that whatever information is coming from the data base is accurate.
- ✓ Availability: All data in the system will be available all the time.
- ✓ User friendly Interface: Users can easily input their information and watch the movie they want easily
- ✓ Error handling exception: The system must recover immediately when a user enters incorrect data

3.3 data collection methods

This stage is about gathering information and identifying important details about necessary facts to construct the required database system. There are about 6 fact finding techniques among them we used two techniques.

- ➢ Observing.
- ➢ Research
    - o Library research
    - o Internet research

3.4 System Development Tools

3.4.1 Hardware Tools

- Personal Computer

3.4.2 Software tools

- **HTML**: - stands for hypertext markup language. The standard markup language for documents designed to be displayed in a web browser [4].
- **Tailwind CSS**: - CSS stands for cascading style sheets CSS describes how HTML elements are to be displayed on screen, paper, or in other media. Tailwind CSS is an open-source CSS framework [5].
- **JavaScript**: -Java script is a lightweight interpreted programming language that is used to program the behavior of web pages [6].
- **Django**: - Django is a high-level web framework written in Python. It follows the Model-View-Controller (MVC) architectural pattern [7], which separates the application into three components:
    i. ***Models***: Models represent the data and the database schema of the application. They interact with the database to create, read, update, and delete (CRUD) data.
    ii. ***Views***: Views handle the logic of the application. They receive requests from clients, interact with the models to retrieve and manipulate data, and generate responses to send back to the clients.
    iii. ***Templates***: Templates define the presentation of the data. They are responsible for generating the HTML, CSS, and JavaScript code that is sent to the client's browser.
- **SQLite**: - SQLite is a lightweight, open-source relational database management system that is designed to be embedded into applications. It provides a simple, self-contained way to store and manage structured data within an application, without requiring a separate server or installation [8].
- **Visual Studio Code**: -Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications.

3.5 System and design methodology

This section consists of the modeling of the proposed system using object-oriented methodology by applying unified modeling language. All the activities performed by the actors are analyzed by using different modeling diagrams.

3.5.1 Object-oriented design

There are many ways to look at a problem to be solved using a software-based solution. One widely used approach to problem solving takes an object-oriented viewpoint. The problem domain is characterized as a set of objects that have specific attributes and behaviors. The objects are manipulated with a collection of functions called methods, operations, or services and communicate with one another through a messaging protocol. Objects are categorized into classes and subclasses. The definition of objects encompasses a description of attributes, behaviors, operations, and messages [9].

3.5.2 Unified Modelling Language.

UML, short for unified modeling language is a standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems as well as for business modeling and other non-software systems. The UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects [10]. UML defines nine types of diagrams those are class, object, use case, sequence, collaboration, state chart, activity, component, and deployment diagram.

3.5.3 Use case diagram

A use case defines a goal-oriented set of interactions between external users and the system under consideration or development and draw as a horizontal ellipse.

- ✓ **An actor:** is a person, organization, or external system that plays a role in one or more interactions with the system and draw as stickman figure.
- ✓ **System boundary**: indicates the scope of the system project. Anything within the box represent functionalities in side in scope.

In order to create relevant use cases for our system, the following actors for the system have been identified:

- ✓ Admin: - A person who has special privilege to manage users, profiles and movies of the system.
- ✓ User: - A Person who tries to watch movies

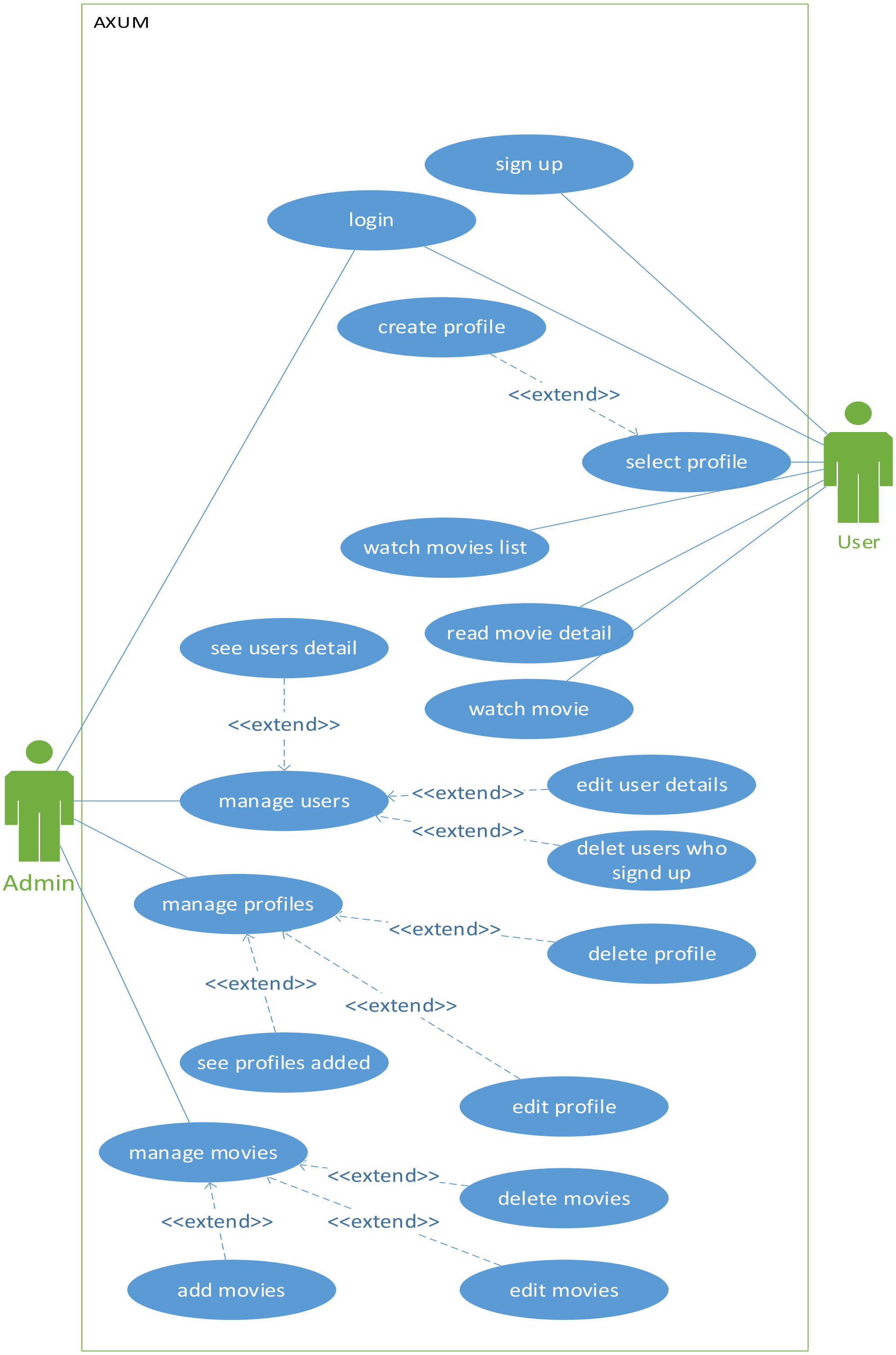


Figure 3.1 Use case diagram

3.5.4 Use case description

Table 3.1 Use case description for signup

Use case name	Signup	
Actor	Users	
Description	New users register using email and password.	
Pre-condition	User’s email has to be unused email address in the system. The password must be strong	
Basic course of action	Actor action	System response
	Step1. Browse the site Step 3. Click on the signup page. Step5. Enter the required input	Step2. The system displays home page. Step4. The system displays signup page. Step6. The system checks the validity of the input. Step7. The system displays appropriate profile page
Alternative course of action	If the information provided is not valid ✓ The system displays error message ✓ The system asks to re input the information again. If users’ email is already used ✓ The system displays already registered message. If the password is too common ✓ The system displays password is too common message.	
Post Condition	Users will be directed profile page.	

Table 3.2 Use case description for user login

Use case name	Login	
Actor	Users	
Description	The system authenticates registered users of the system.	
Pre-condition	Users must be registered.	
Basic course of action	Actor action	System response
	Step1. Browse the site Step 3. Click on the login page. Step5. Enter the required input	Step2. The system displays home page. Step4. The system displays login page. Step6. The system checks the validity of the input. Step7. The system displays appropriate profile page users and
Alternative course of action	If the information provided is not valid ✓ The system displays error message ✓ The system asks to re input the information again.	
Post Condition	Users will be directed to profile page.	

Table 3.3 Use case description for admin login

Use case name	Login
Actor	Admin
Description	The system authenticates registered admin of the system.

Pre-condition	admin must be registered.	
Basic course of action	Actor action	System response
	Step1. Browse the admin site Step5. Enter the required input	Step2. The system displays admin login page. Step6. The system checks the validity of the input. Step7. The system displays appropriate admin home page
Alternative course of action	If the information provided is not valid ✓ The system displays error message ✓ The system asks to re input the information again.	
Post Condition	admin will be directed to admin home page.	

Table 3.4 Use case description for select profile

Use case name	Select profile	
Actor	Users	
Description	Authenticated users select profile in order to watch the right movie for them	
Pre-condition	Users must be logged in	
	Actor action	System response

Basic course of action	Step1. As soon as user’s login they are directed to the profiles page Step 3. Click on the profile they want	Step2. The system displays available profiles if any Step4. The system displays appropriate movie list page
Alternative course of action	If the there are no profiles available the user will not be able to select a profile, they must first create a profile.	
Post Condition	Users will be directed movie list page.	

Table 3.5 Use case description for create profile

Use case name	Create profile	
Actor	Users	
Extends	Select profile	
Description	Authenticated users create profile in order to watch the right movie for them	
Pre-condition	Users and admin must be logged in	
Basic course of action	Actor action	System response
	Step1. Click on create profile page Step 3. Enter the required input	Step2. The system displays create profile page Step4. The system checks the validity of the input. Step5. The system displays appropriate select profile page
Alternative course of action	If the information provided is not valid ✓ The system displays error message ✓ The system asks to re input the information again.	

Post Condition	Users will be directed profile page.

Table 3.6 Use case description for watch movie list

Use case name	Watch movie list	
Actor	Users	
Description	Authenticated users can see the variety list of movies available in the system	
Pre-condition	Users must login and select a profile	
Basic course of action	Actor action	System response
	Step1. Click on their profile Step 3. Watch the list of movies available	Step2. The system displays appropriate watch movie list page
Alternative course of action	None	
Post Condition	Users will watch list of movies	

Table 3.7 Use case description for read movie details

Use case name	Read movie details
Actor	Users
Description	Authenticated users can read detail about the movie they want to watch
Pre-condition	Users must be logged in and select a profile

Basic course of action	Actor action	System response
	Step1. Click on the movie they want to read detail Step 3. Read the appropriate details of the movie	Step2. The system displays appropriate movie detail page
Alternative course of action	None	
Post Condition	Users will read the detail of the movie	

Table 3.8 Use case description for watch movie

Use case name	Watch movie	
Actor	Users	
Description	Authenticated users can watch and enjoy the movie they want	
Pre-condition	Users must be logged in and select a profile	
Basic course of action	Actor action	System response
	Step1. After reading the movie details click on play Step 3. Watch the movie	Step2. The system plays appropriate movie
Alternative course of action	None	
Post Condition	Users will enjoy the movie	

Table 3.9 Use case description for manage users

Use case name	Manage users	
Actor	Admin	
Description	Admin can see users’ detail, delete users and edit users’ information	
Pre-condition	Admin must be login and select uses button	
Basic course of action	Actor action	System response
	Step1. Click on users’ button Step 3. Select the appropriate manage option (see users’ detail, delete users and edit users’)	Step2. The system plays appropriate list of users
Alternative course of action	If there are no users that are signed in the system will display empty users list	
Post Condition	Select the appropriate manage option (see users’ detail, delete users and edit users’)	

Table 3.10 Use case description for see user’s details

Use case name	see user’s details	
Actor	Admin	
Extends	manage users	
Description	Admin can see details of users who have signed in to the system	
Pre-condition	admin must be logged in	
	Actor action	System response

Basic course of action	Step1. Click on the user we want to see details of	Step2. The system displays appropriate user’s information
Alternative course of action	If there are no users that are signed in the system will display empty users list	
Post Condition	observe user’s details successfully	

Table 3.11 Use case description for edit user’s details

Use case name	Edit user details	
Actor	Admin	
Extends	manage user	
Description	Admin can edit details of users who have signed in to the system	
Pre-condition	admin must be logged in	
Basic course of action	Actor action	System response
	Step1. Click on the user we want to see details of Step3. From the displayed details choose one edit and click on save button	Step2. The system displays appropriate user’s information Step4. Display success message
Alternative course of action	If there are no users that are signed in the system will display empty users list	
Post Condition	Edit user successfully	

Table 3.12 Use case description for delete user

Use case name	delete user	
Actor	Admin	
Extends	manage users	
Description	Admin can delete details of users who have signed in to the system	
Pre-condition	admin must be logged in	
Basic course of action	Actor action	System response
	Step1. Click on the user we want to delete Step3. Click on delete button	Step2. The system selects the user Step4. Display success message
Alternative course of action	If there are no users that are signed in the system will display empty users list	
Post Condition	Delete user successfully	

Table 3.13 Use case description for manage profile

Use case name	Manage profile	
Actor	Admin	
Description	Admin can see profiles, delete profile, create profile and edit profile	
Pre-condition	Admin must be logged in and select profile button	
	Actor action	System response

Basic course of action	Step1. Click on profile button Step 3. Select the appropriate manage option (see profiles, delete profile, create profile and edit profile)	Step2. The system plays appropriate list of profiles
Alternative course of action	If there are no profiles admin will not be able to manage profiles	
Post Condition	System displays all profiles with the appropriate manage option	

Table 3.14 Use case description for see profiles

Use case name	see profiles	
Actor	Admin	
Extends	manage profiles	
Description	Admin can see list of profiles created and their details	
Pre-condition	admin must be logged in	
Basic course of action	Actor action	System response
	Step1. Click on the profile we want to see details of	Step2. The system displays appropriate profile information
Alternative course of action	None	
Post Condition	observe profile details successfully	

Table 3.15 Use case description for edit profile

Use case name	Edit profile

Actor	Admin	
Extends	manage profile	
Description	Admin can edit details of profile who have been created by the users	
Pre-condition	admin must be logged in	
Basic course of action	Actor action	System response
	Step1. Click on the user we want to see edit Step3. From the displayed details choose one edit and click on save button	Step2. The system displays appropriate profile information Step4. Display success message
Alternative course of action	None	
Post Condition	Edit profile successfully	

Table 3.16 Use case description for delete profile

Use case name	delete profile	
Actor	Admin	
Extends	manage profile	
Description	Admin can delete details of users who have signed in to the system	
Pre-condition	admin must be logged in	
	Actor action	System response

Basic course of action	Step1. Click on the profile we want to delete Step3. Click on delete button	Step2. The system selects the profile Step4. Display success message
Alternative course of action	None	
Post Condition	Delete user successfully	

Table 3.17 Use case description for manage movie

Use case name	Manage movie	
Actor	Admin	
Description	Admin can add movie, delete movie and edit movie details	
Pre-condition	Admin must be login	
Basic course of action	Actor action	System response
	Step1. Click on movies button Step 3. Select the appropriate manage option (add movie, delete movie and edit movie details)	Step2. The system plays appropriate list of movies
Alternative course of action	None	
Post Condition	Select the appropriate manage option (add movie, delete movie and edit movie details)	

Table 3.18 Use case description for add movie

Use case name	Add movie	
Actor	Admin	
Extends	manage movie	
Description	Admin add a new movie with its appropriate details	
Pre-condition	admin must be logged in	
Basic course of action	Actor action	System response
	Step1. Click add movie button Step3 Enter the required information	Step2. The system displays add movie page Step4. System checks validity of the information Step4. System displays success message
Alternative course of action	If the information provided is not valid or have not provided all information ✓ The system displays error message ✓ The system asks to re input the information again.	
Post Condition	Movie added successfully	

Table 3.19 Use case description for edit movie

Use case name	Edit movie
Actor	Admin
Extends	manage movie
Description	Admin can edit details of movie

Pre-condition	admin must be logged in	
Basic course of action	Actor action	System response
	Step1. Click on the movie we want to edit Step3. From the displayed details choose one edit and click on save button	Step2. The system displays appropriate movie information Step4. Display success message
Alternative course of action	None	
Post Condition	Edit movie successfully	

Table 3.20 Use case description for delete movie

Use case name	delete movie	
Actor	Admin	
Extends	manage movie	
Description	Admin can delete movie	
Pre-condition	admin must be logged in	
	Actor action	System response

Basic course of action	Step1. Click on the movie we want to delete Step3. Click on delete button	Step2. The system selects the movie Step4. Display success message
Alternative course of action	None	
Post Condition	Delete movie successfully	

3.5.5 Class diagram

The class diagram shows the building blocks of any object-orientated system. Class diagrams are most useful in illustrating relationships between classes and interfaces. This class diagram that shows the relationships between the different models in the app. It can be useful for understanding the structure of the app's data model and for documenting the app's architecture. We used the Django-extensions package which is a third-party package that provides various useful commands and utilities for Django development.

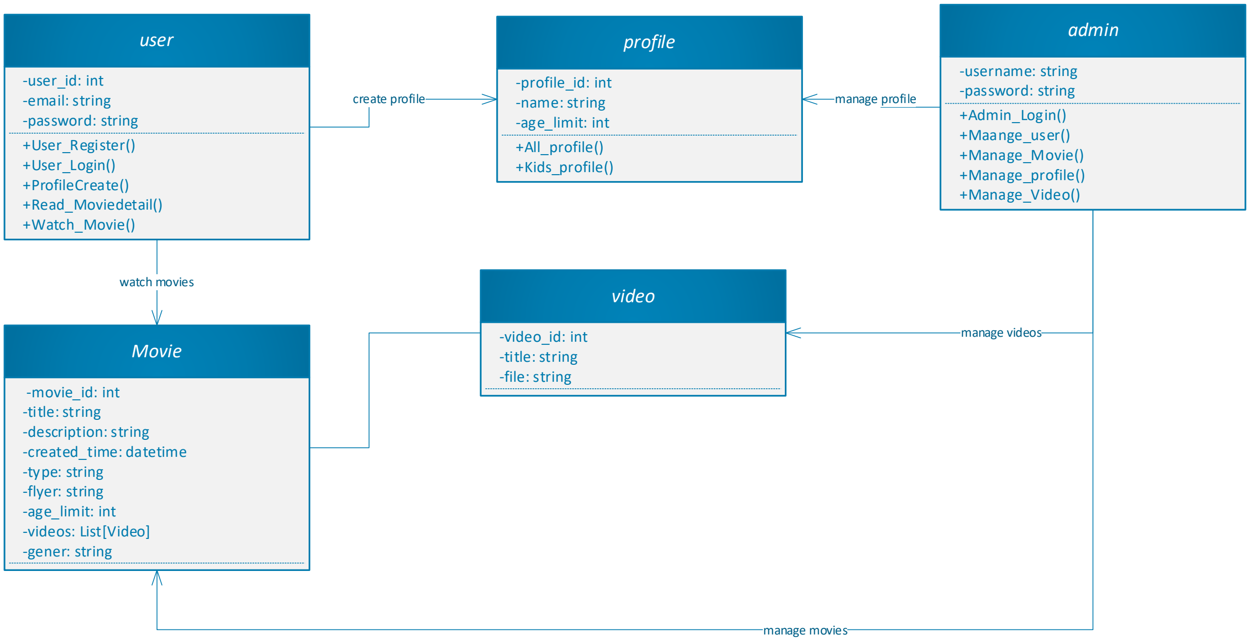


Figure 3.2 class diagram

3.5.6 Activity diagram

Activity diagram illustrates the dynamic nature of a system by modeling the flow of control from activity to activity.

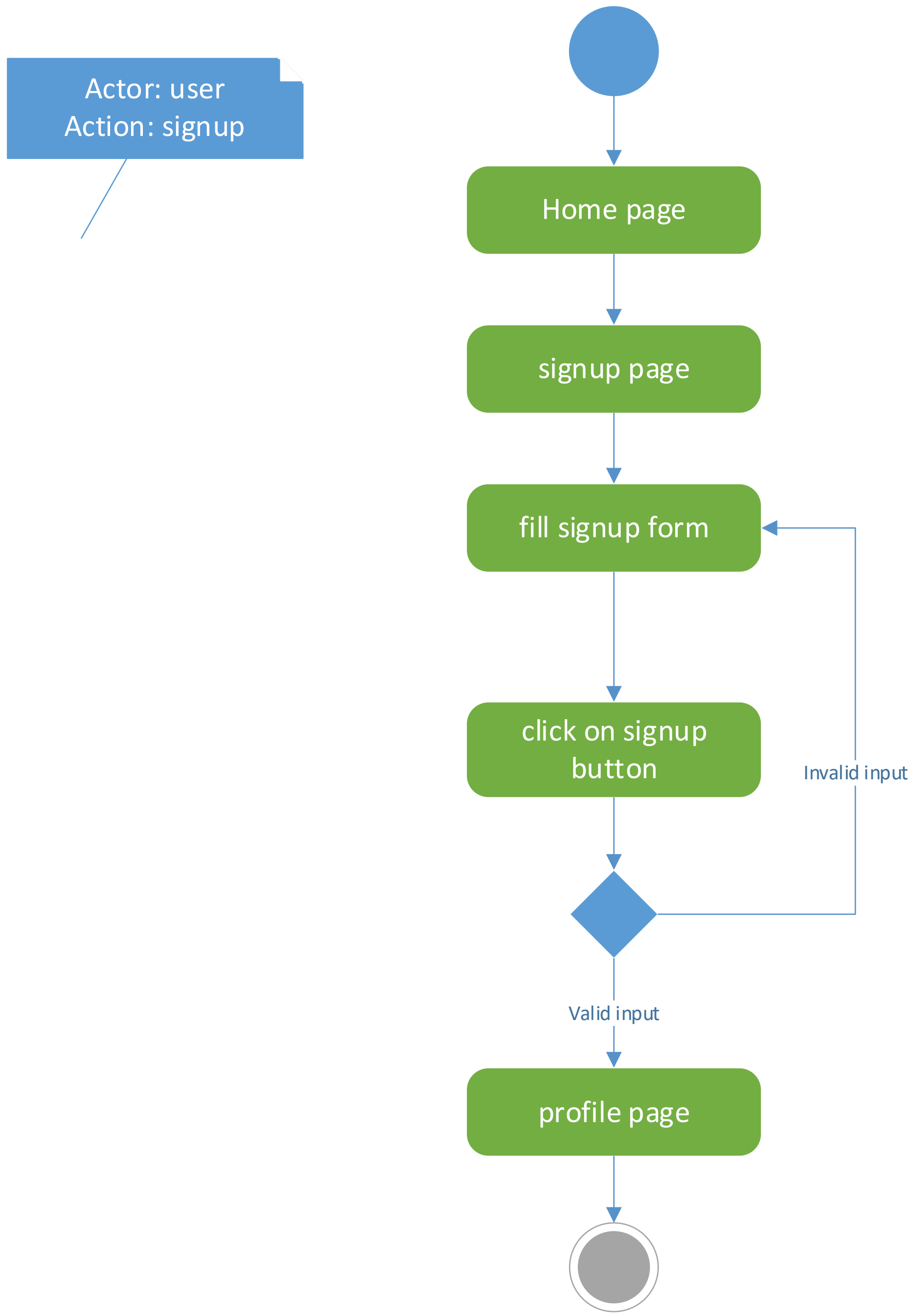


Figure 3.3 Activity diagram for signup

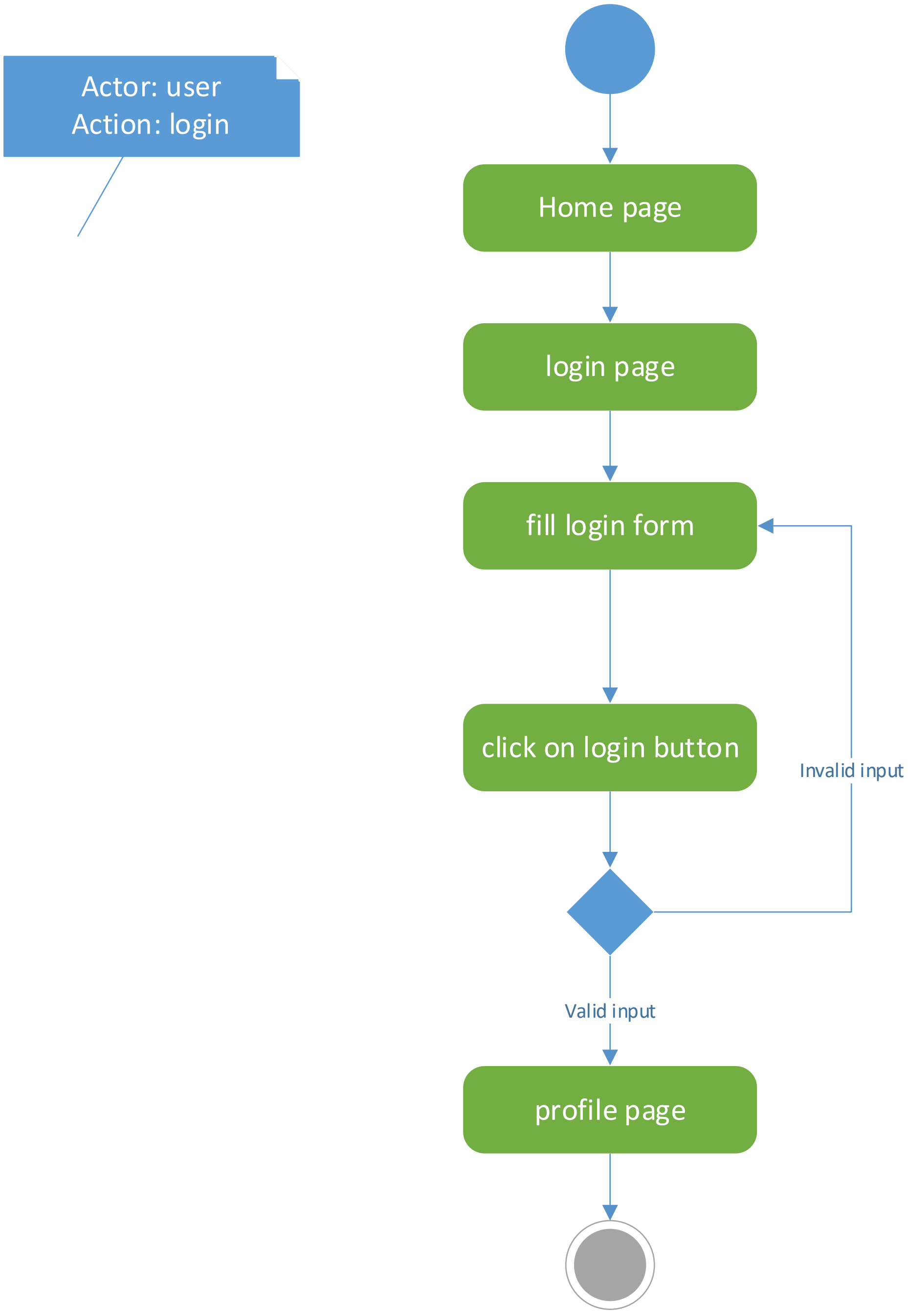


Figure 3.4 Activity diagram for user login

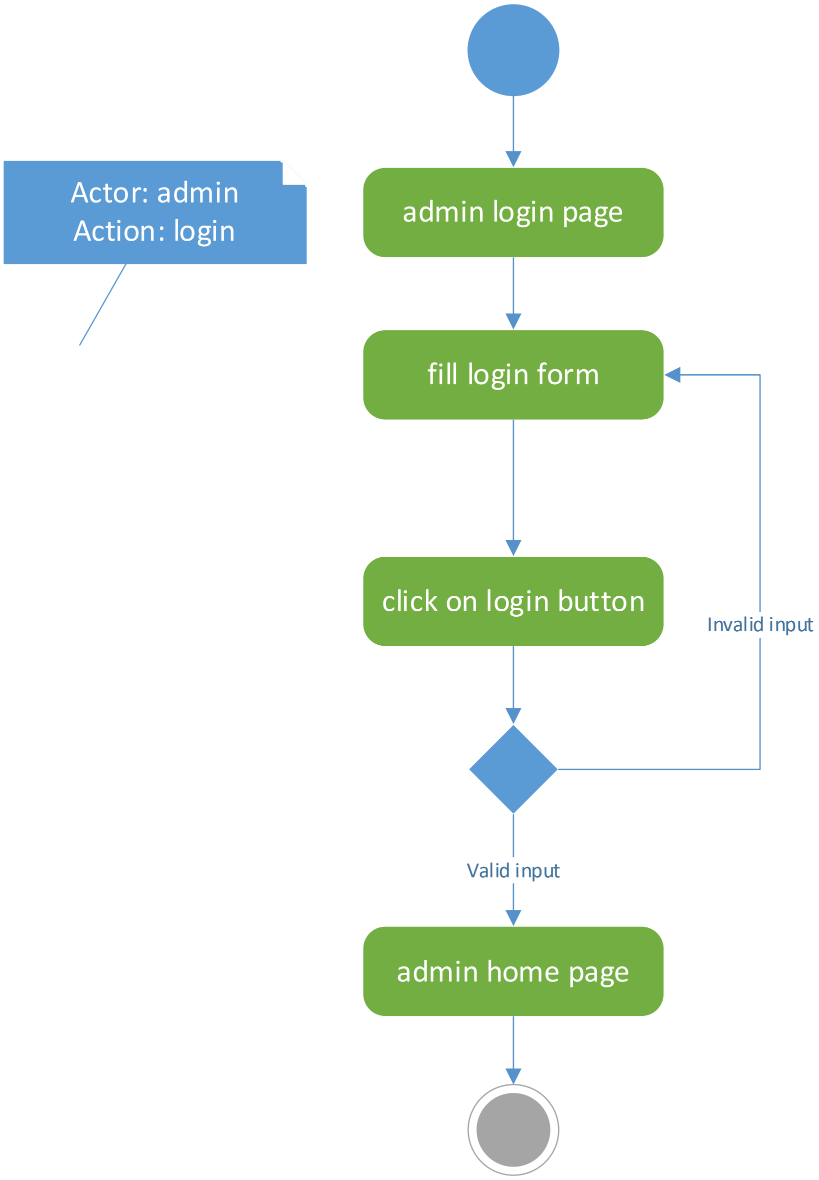


Figure 3.5 Activity diagram for admin login

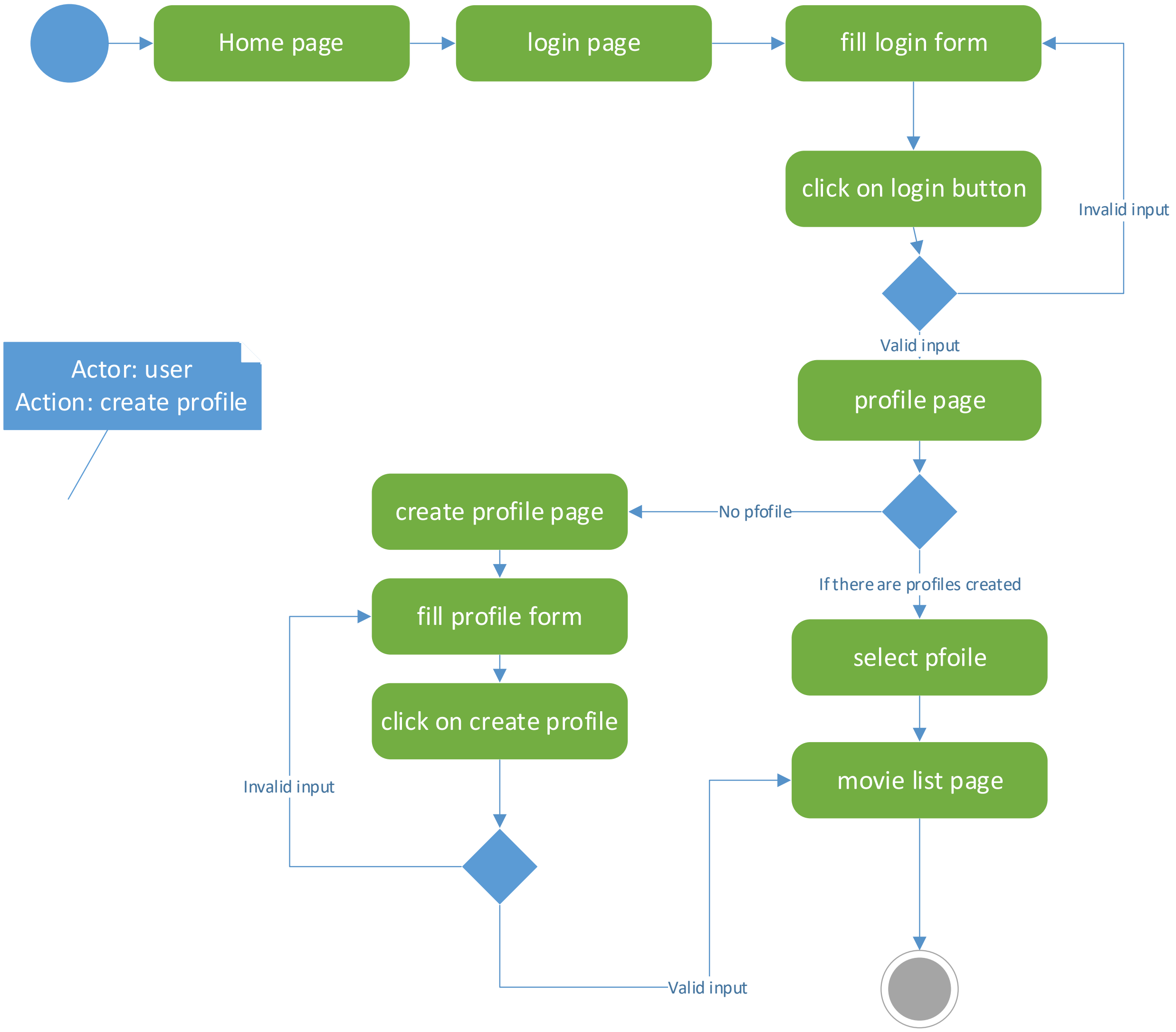


Figure 3.6 Activity diagram for create profile

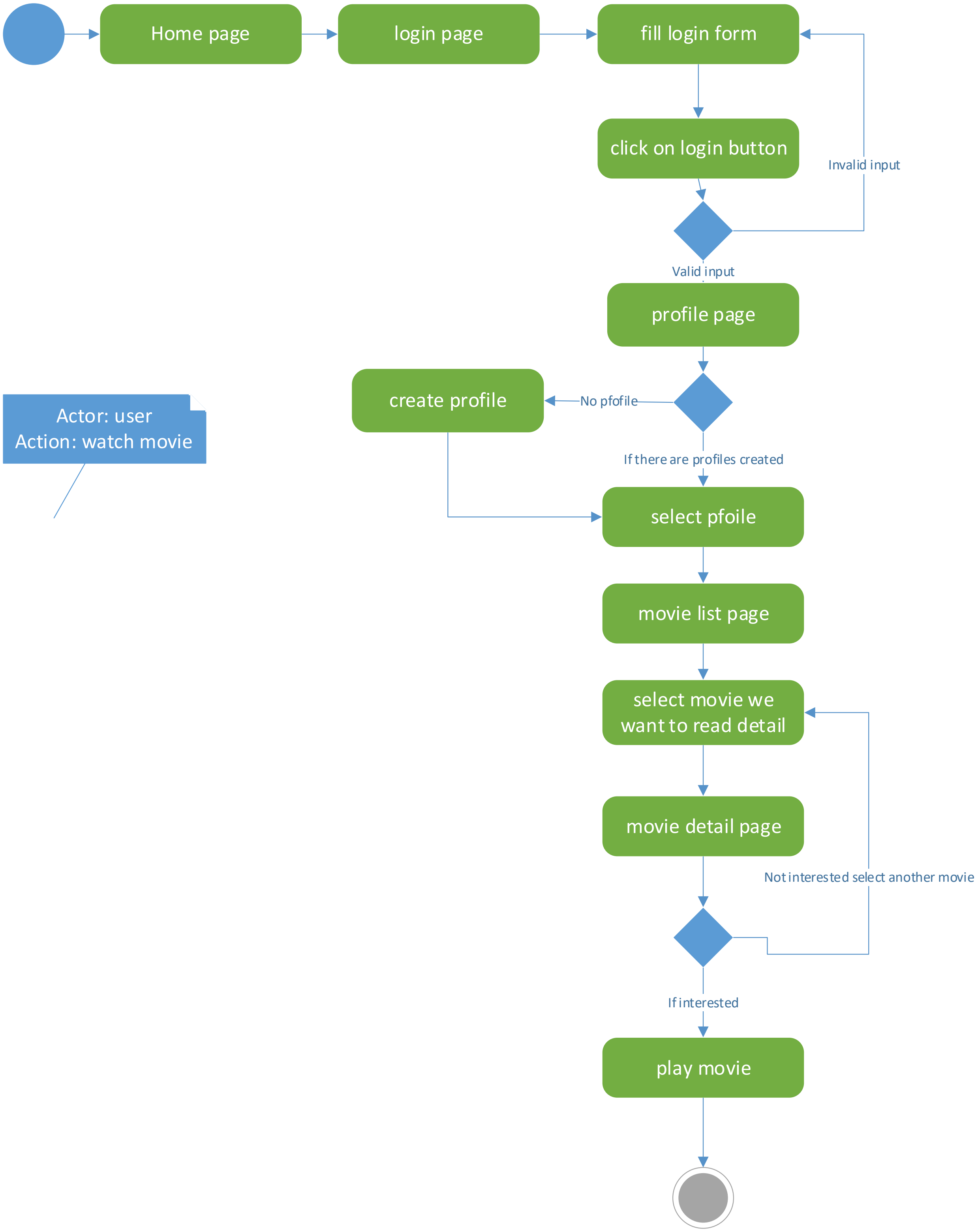


Figure 3.7 Activity diagram for watch movie

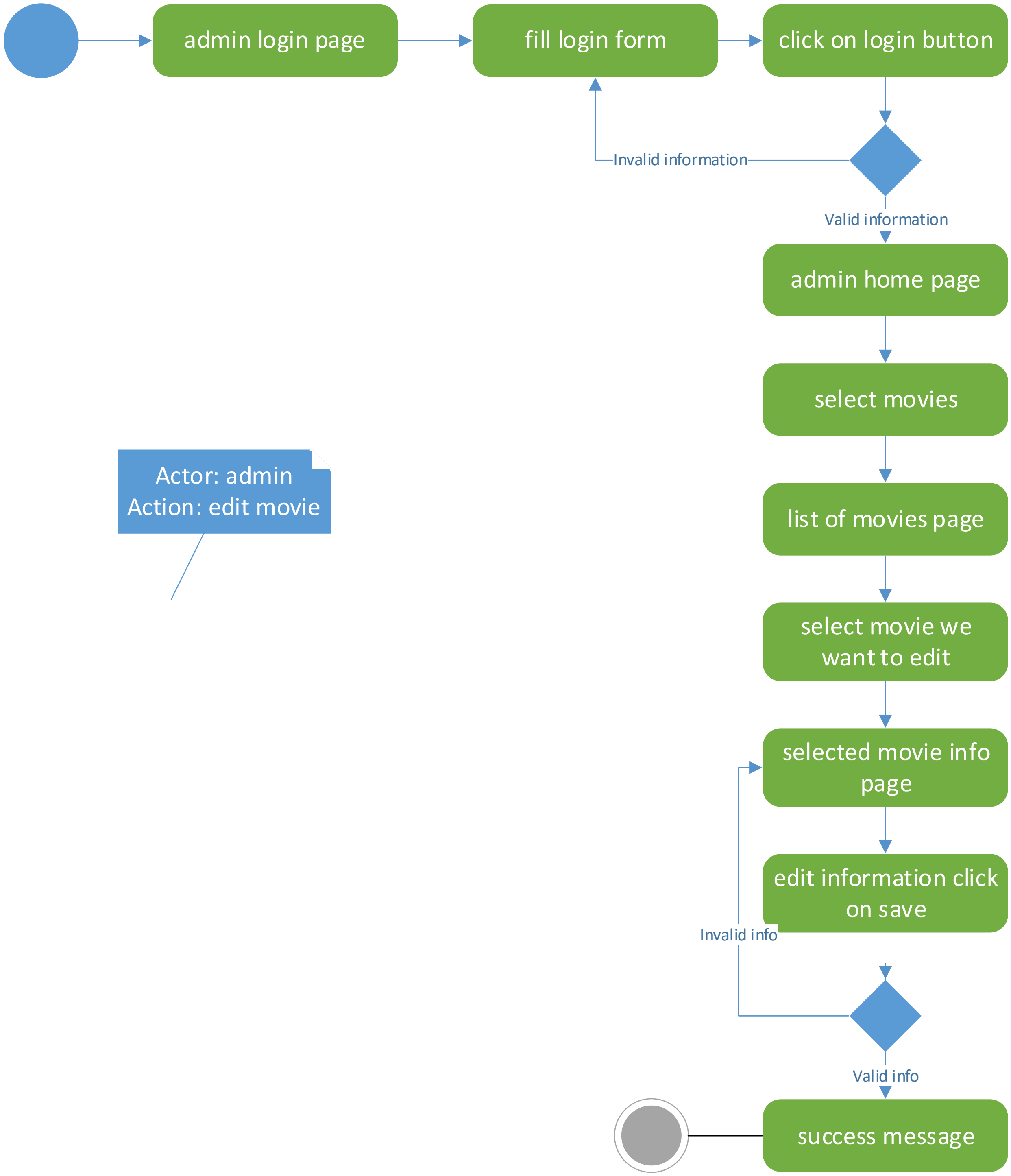


Figure 3.8 Activity diagram for edit movie

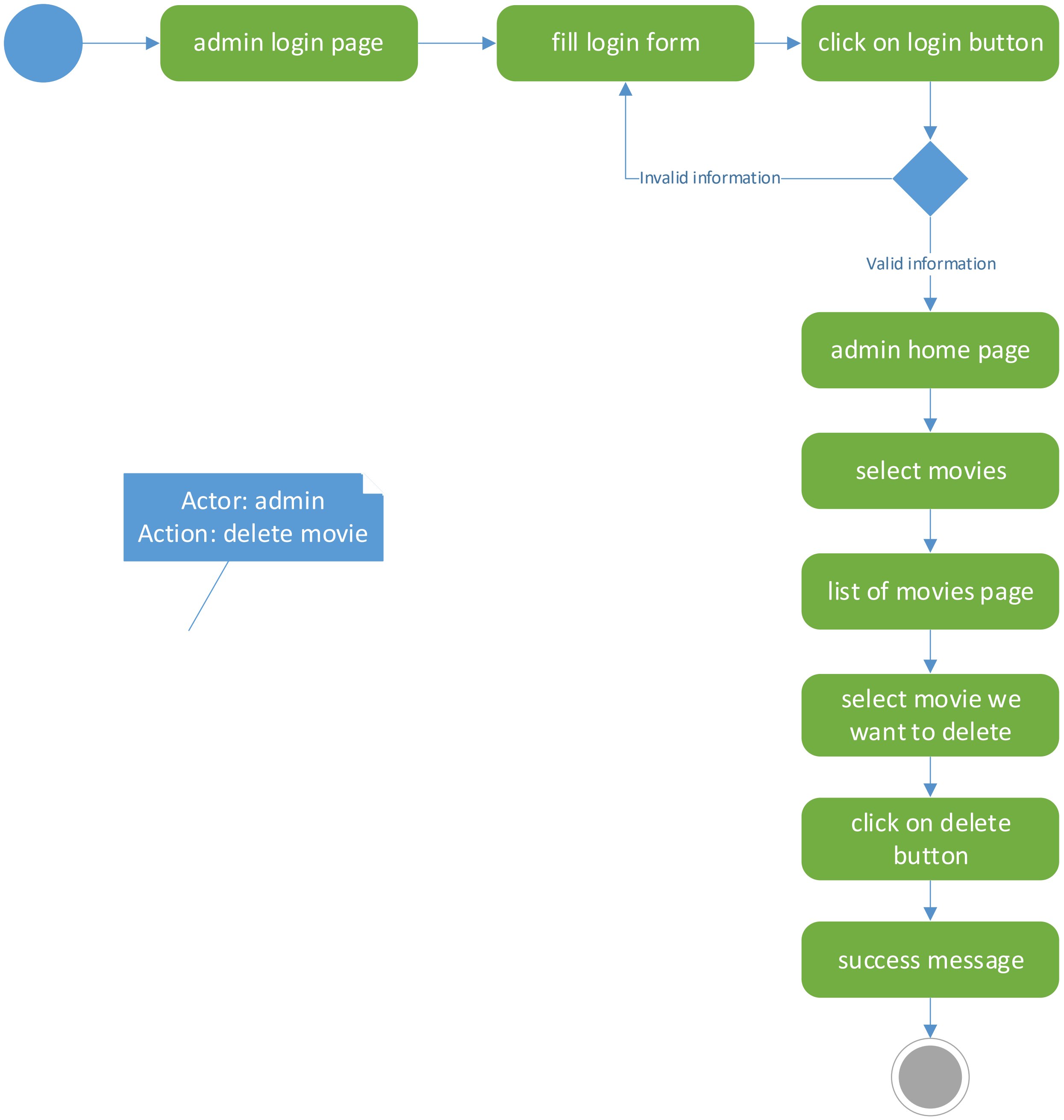


Figure 3.9 Activity diagram for delete movie

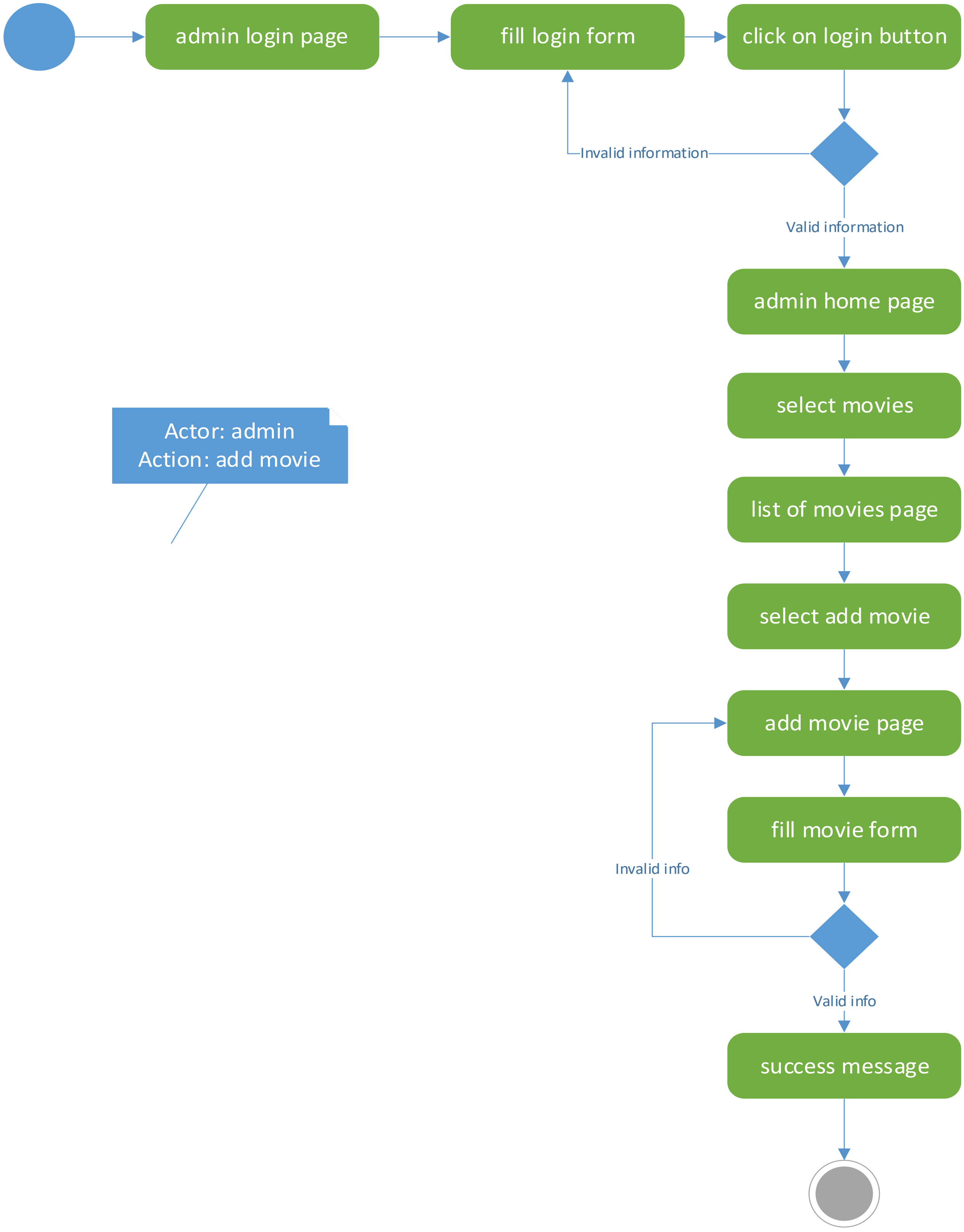


Figure 3.10 Activity diagram for add movie

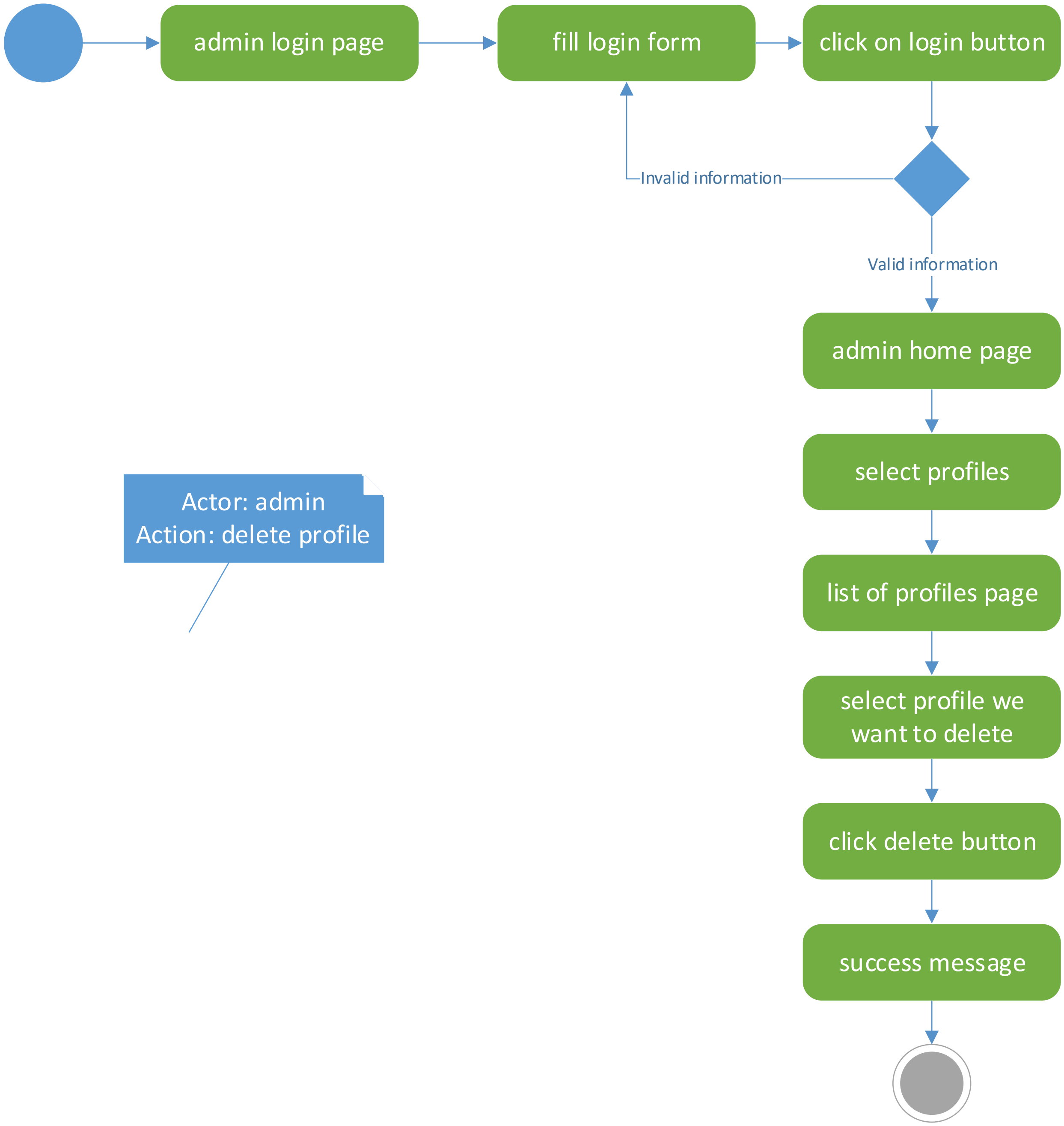


Figure 3.11 Activity diagram for delete profile

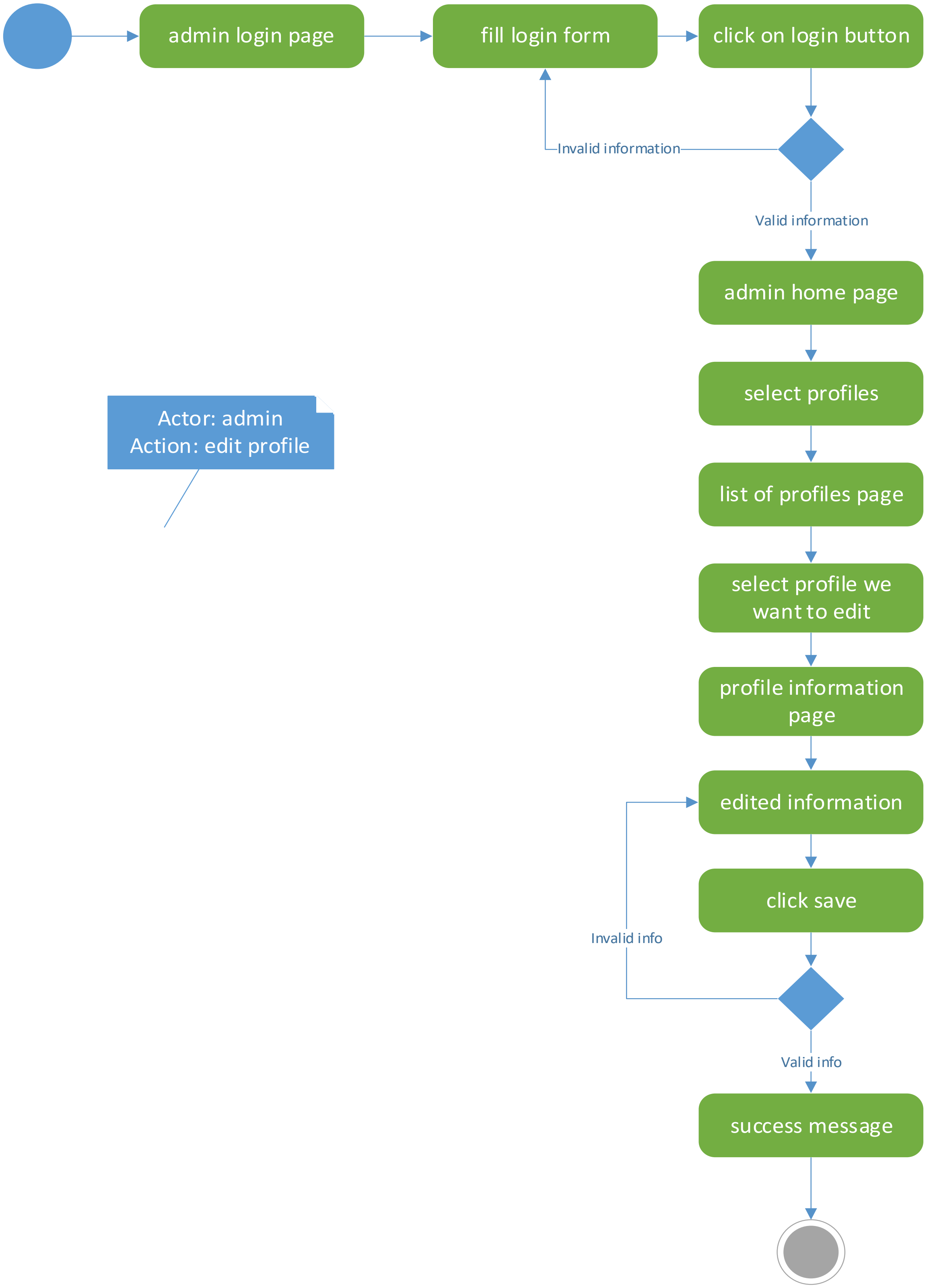


Figure 3.12 Activity diagram for edit profile

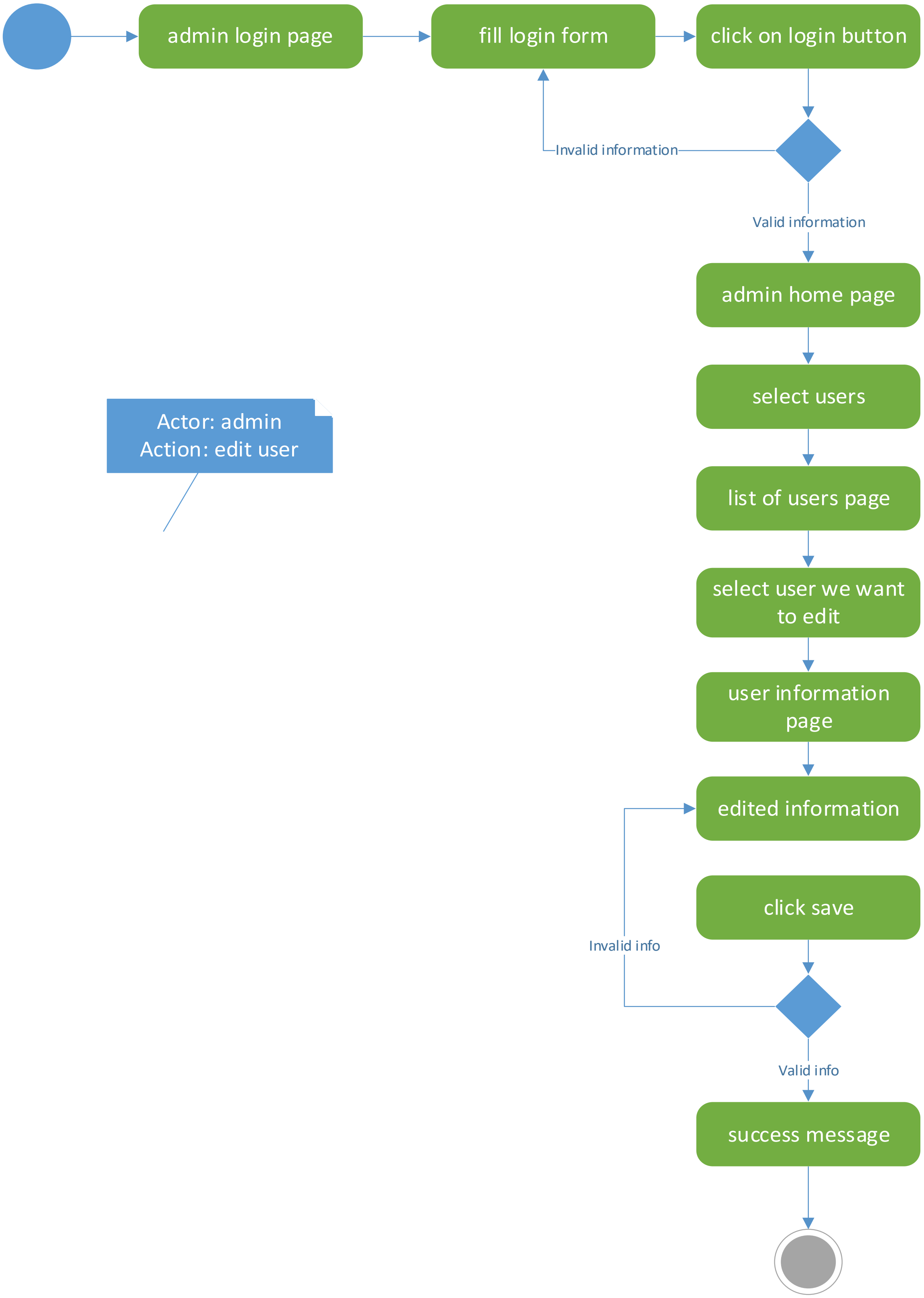

admin login page
fill login form
click on login button
Invalid information
Valid information
admin home page
select users
list of users page
select user we want to edit
user information page
edited information
click save
Invalid info
Valid info
success message
Actor: admin
Action: edit user

Figure 3.13 Activity diagram for edit user

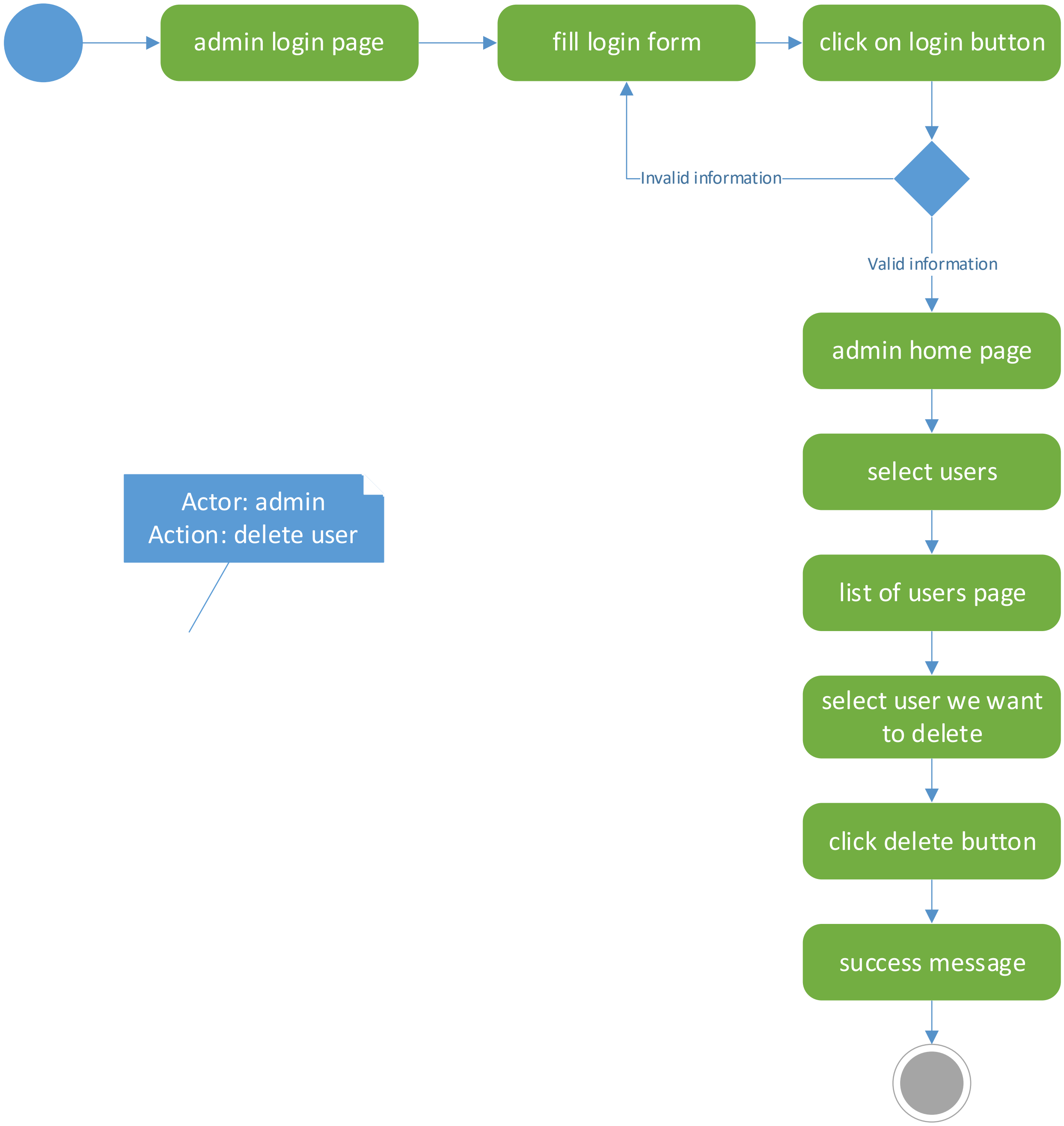


Figure 3.14 Activity diagram for delete user

3.6 Implementation and testing system

During this stage we turned the design specification into working computer code and then the code is tested until most of the errors have been detected and corrected.

3.6.1 Final Testing of the system

Testing by Requirements: -The requirements that are tested by the group member during the implementation is correctness, performance, accuracy, security of the system.

- ✓ Testing the correctness: - correctness determines how users can interact with the software and how the software should behave when it is used correctly. Users can easily interact with the application since it has easily understandable interface and the application responds correctly.
- ✓ Testing the Accuracy: The system gives only valid result, if no data is found with the specified criteria the system should not give invalid response. Since, our application fulfills these characteristics it is accurate.
- ✓ Testing the security: - To ensure the security of our application user must login to the system with email and password. The system allows only authorized users to login.

3.6.2 Error Handling

The system checks user inputs to the system to handle error. When the user tries to insert invalid data, this system handles and consumes the error or it does not allow to insert invalid input.

3.6.2.1 Unit testing

In unit testing we try to explore or test each module individually. When we test the module and the input is not correct the system retrieves an error message Integration and Systems Testing

- ✓ Integration testing: in integration testing we integrate and test all components of the system modules that a program comprises for testing purpose.
- ✓ System testing: System test ensures that the entire integrated software system meets requirements. It tests a configuration to insure known and predictable results.

Chapter Four

Result And Discussion

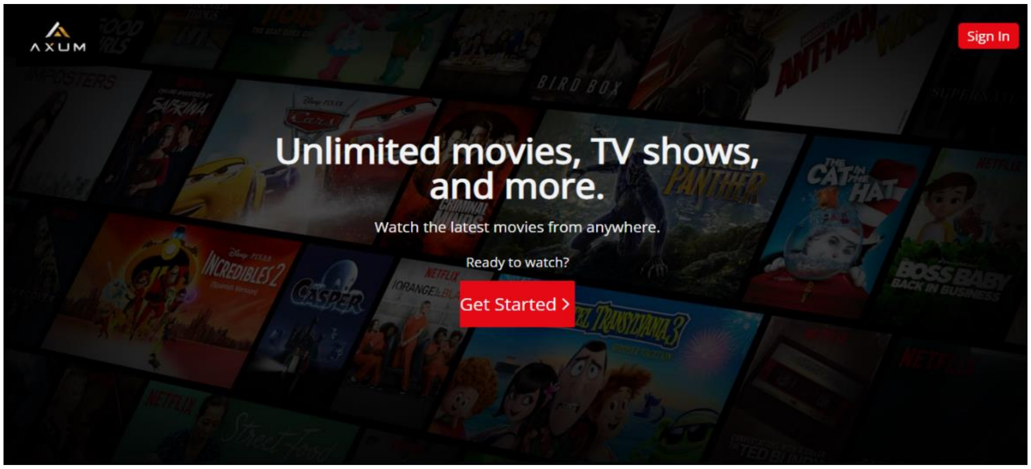
4.1 Introduction

The purpose of this section is to provide a clear understanding of the project outcomes and to draw conclusions based on the data collected. Therefore, the results and discussion section are a crucial part of any document, as it provides the reader with a comprehensive understanding of the research findings and their implications. Our project creates a graphical user interface that makes it easier for users and administrators to use. The system has a home page which visitors interface first. The home page describes the purpose of the system, advertise the system.

4.2 Sample user interface result

Description

The homepage is the initial webpage that users encounter when attempting to access AXUM. It serves as an introduction to the purpose and functionality of the AXUM, and is designed to promote and advertise our project to potential users and stakeholders. This page, also referred to as index.html in our source code, is carefully crafted to provide users with an overview of the features and benefits of AXUM, and to encourage them to explore the website further. Through a combination of engaging content and eye-catching visuals, we aim to capture the attention of our audience and convey the unique value proposition of AXUM. By presenting a clear and compelling message on our homepage.



Unlimited movies, TV shows, and more.

Watch the latest movies from anywhere.

Ready to watch?

Figure 3.15 Home page image

Description

The login page is where users who have already signed up for the system can log in using the email address and password they provided during the sign-up process. This page is accessed by clicking the "Get Started" or "Sign In" button on the home page, and serves as a secure gateway to the system's features and functionality. Once a user has entered their login credentials, the system will authenticate their identity and grant them access to their account. The login page is designed to be intuitive and user-friendly, with clear instructions and prompts to guide users through the login process. Overall, the login page plays a critical role in providing users with a secure and seamless experience when accessing the system, and serves as a vital component in the overall user experience. This page, also referred to as login.html in our source code

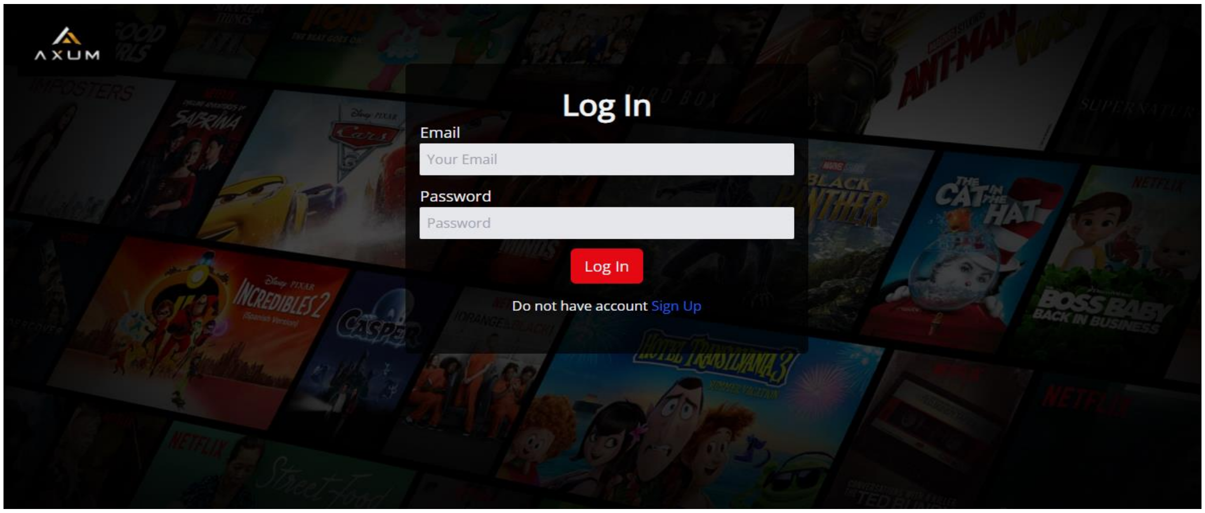


Figure 3.16 log in page image

Description

The sign-up page is where users can register for the system by providing their email address and password. This page is accessed by clicking the "Sign Up" button on the login page, and serves as the first step in creating an account for the system. The sign-up page is carefully designed to collect all necessary information from the user, email address and chosen password. Once the user has filled out the required fields and submitted their information, the system will validate their data and create their account or register their information to the data base. This page, also referred to as signup.html in our source code, is an essential part of the user experience, as it provides users with a streamlined and secure way to join the system. By making the sign-up process as simple and user-friendly as possible.

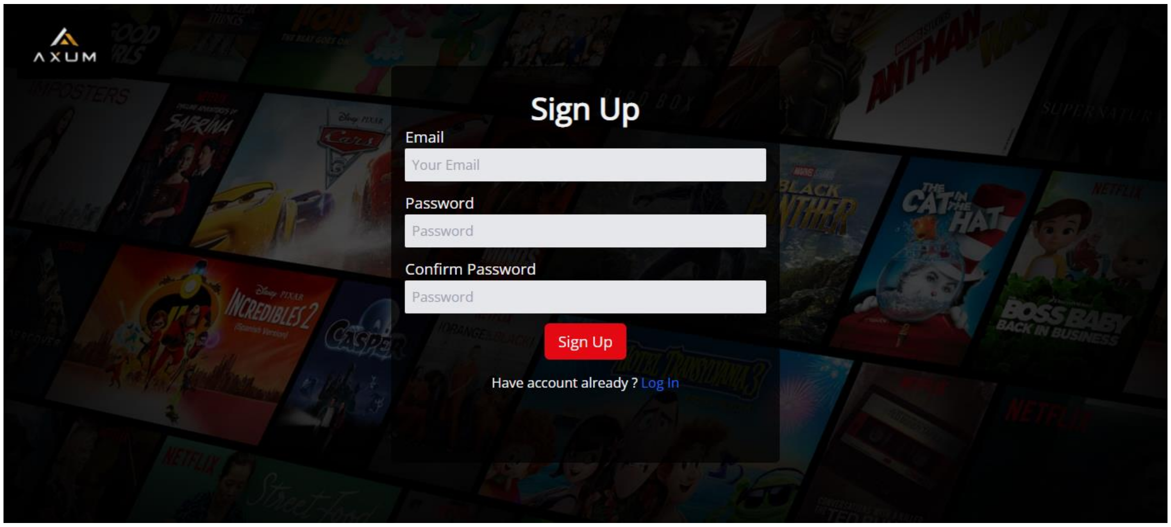


Figure 3.17 sign up page image

Description

The profile page is the first page that users see after logging into the system. It is the central hub where users can access their own personalized profiles, which allow them to enjoy a customized and tailored experience based on their preferences and interests. This page is designed to be user-friendly and intuitive, with clear navigation options and easy-to-use tools for managing and creating profiles. For example, parents may be able to select a kid's profile to access movies and TV shows that are appropriate for children, while other users can choose from a variety of options to personalize their experience. The profile page, also known as profileList.html in our source code, is a critical component of the user experience. Overall, the profile page plays a crucial role in creating a seamless and enjoyable experience for users, and is an essential part of our system's overall design and functionality.

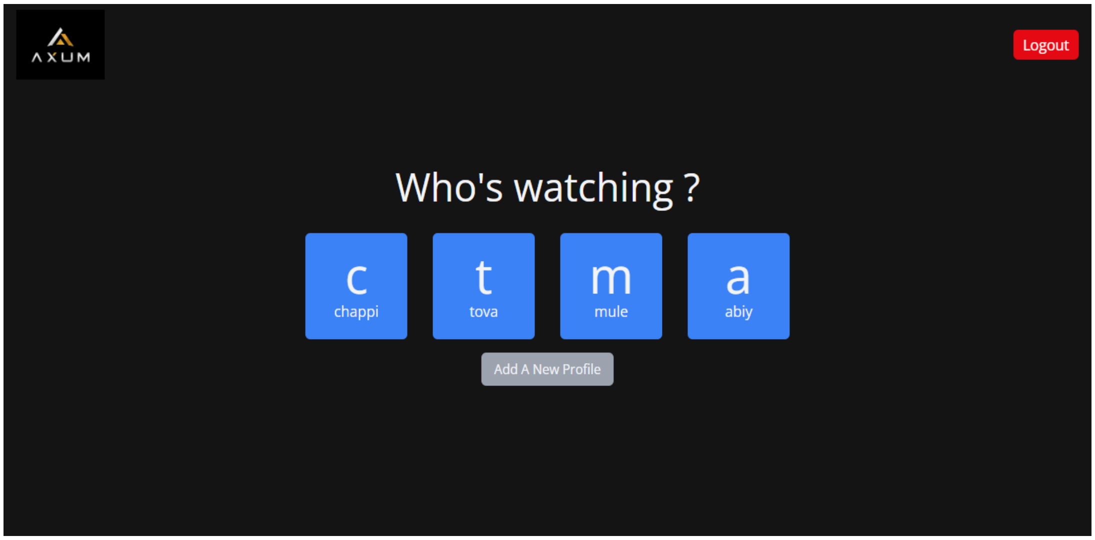
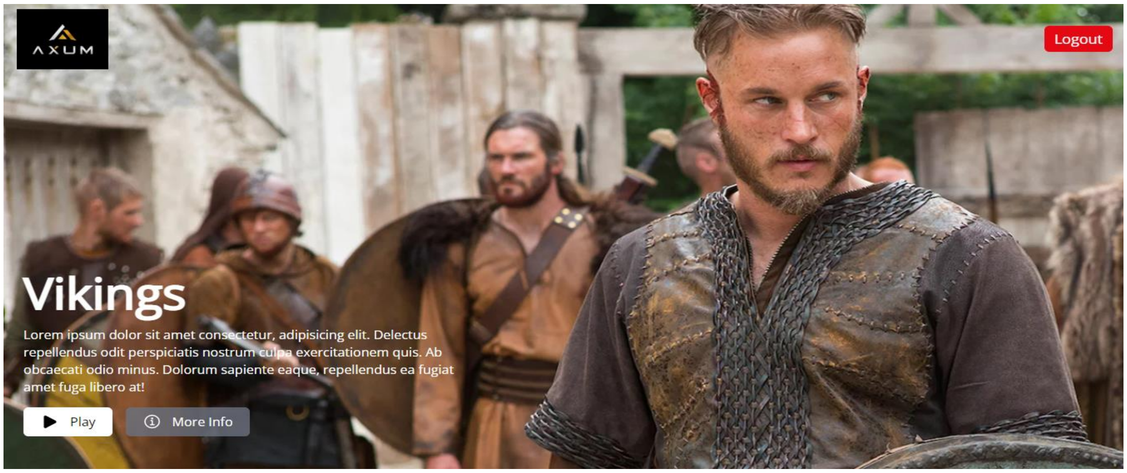


Figure 3.18 profile page image

Description

The movie list page is the page that users access after selecting their profile. It is the page where users can view a list of movies that they can choose from to watch. This page is carefully designed to provide users with a clear and comprehensive list of movies that are available for streaming or download. Users can browse through the list and select the movie they want to watch, or use search and filter tools to find specific movies based on their preferences. The movie list page, also known as movieList.html in our source code, is a vital component of the user experience, as it provides users with a convenient and easy-to-use way to discover new movies and enjoy their favorite films.



Lorem ipsum dolor sit amet consectetur, adipisicing elit. Delectus repellendus odit perspiciatis nostrum culpa exercitationem quis. Ab obcaecati odio minus. Dolorum sapiente eaque, repellendus ea fugiat amet fuga libero at!

Figure 3.19 movie list (1) page image

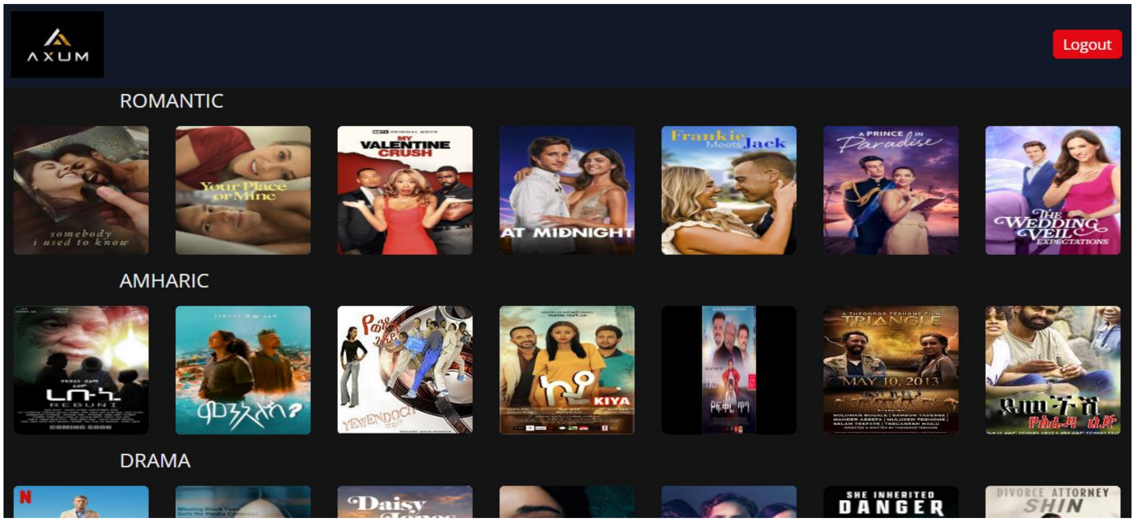
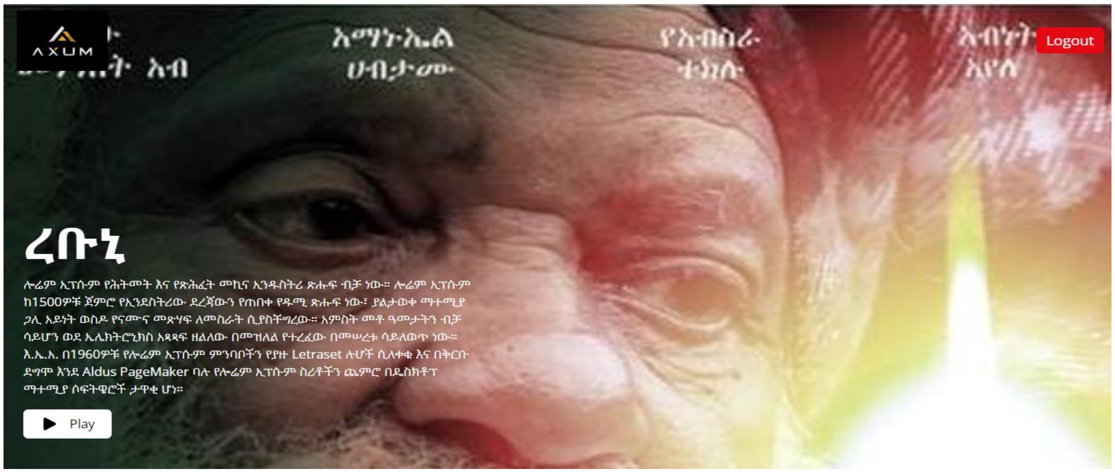


Figure 3.20 movie list (2) page image

Description

The movie detail page is the page where users can view detailed information about a specific movie before deciding whether or not to watch it. This page typically includes a comprehensive description of the movie. The movie detail page is designed to be visually engaging and informative, with a user-friendly layout that makes it easy to browse and explore different movie options. Users can typically access the movie detail page by clicking on a specific movie from the movie list page. The movie detail page, also known as movieDetail.html in our source code, is a critical component of the user experience, as it provides users with the information, they need to make informed decisions about which movies to watch. By presenting users with rich and engaging movie descriptions and other relevant information, we aim to create a seamless and enjoyable user experience that keeps users coming back for more.



ረቡኒ

ሎሬም ኢፕሱም የሕትመት እና የጽሕፈት መኪና ኢንዱስትሪ ጽሑፍ ብቻ ነው። ሎሬም ኢፕሱም ከ1500ዎቹ ጀምሮ የኢንደስትሪው ደረጃውን የጠበቀ የዱሚ ጽሑፍ ነው፣ ያልታወቀ ማተሚያ ጋሊ አይነት ወስዶ የናሙና መጽሃፍ ለመስራት ሲያስችግረው። አምስት መቶ ዓመታትን ብቻ ሳይሆን ወደ ኤሌክትሮኒክስ አጻጻፍ ዘልለው በመዝለል የተረፈው በመሠረቱ ሳይለወጥ ነው። እ.ኤ.አ. በ1960ዎቹ የሎሬም ኢፕሱም ምንባቦችን የያዙ Letraset ሉሆች ሲለቀቁ እና በቅርቡ ደግሞ እንደ Aldus PageMaker ባሉ የሎሬም ኢፕሱም ስሪቶችን ጨምሮ በዴስክቶፕ ማተሚያ ሶፍትዌሮች ታዋቂ ሆነ።

Figure 3.21 movie detail page image

Description

The watch movie page, also known as showMovie.html in our source code, is where users can begin streaming and watching the movie they have selected. This page typically includes a media player that allows users to start, pause, play, fast-forward, and rewind the movie as needed. The watch movie page is designed to be user-friendly and intuitive, with clear and easy-to-use controls that allow users to customize their viewing experience based on their preferences. By providing a seamless and engaging movie-watching experience, we aim to create a system that keeps users coming back for more. The watch movie page is a critical part of this experience, as it allows users to fully immerse themselves in the movies they have selected and enjoy a high-quality viewing experience that meets their needs and preferences.

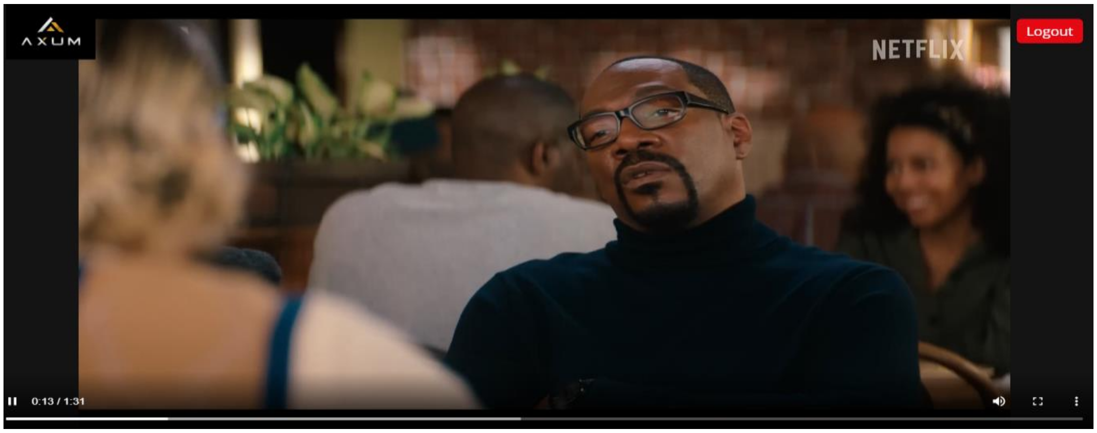


Figure 3.22 watch movie page image

Chapter Five

Conclusion and Recommendations

5.1 Conclusion

Online movie streaming is an expanding field that offers vast opportunities for users to find their desired movies conveniently. In conclusion, our movie streaming web app AXUM, represents a major achievement in the development of user-friendly and intuitive online movie streaming. With a vast library of movies spanning different genres and categories, users can easily find and watch their favorite movies from any device. Additionally, AXUM provides a convenient and user-friendly platform for users to browse and stream movies online.

To build AXUM, our team used cutting-edge web development technologies, including HTML, Tailwind CSS, and JavaScript for the frontend, and Django and SQLite for the backend and database, respectively. We also took great care to thoroughly document the development process, including the methods used to build the system, such as UML class diagram, use case diagram, and activity diagram, which provide a clear and detailed overview of how the system works. In addition to the documentation of the development process, we extensively tested the system using unit testing and implemented robust error handling techniques to ensure that the system remains stable and reliable, even under challenging conditions. Our profiles also provide options for user's kids to enjoy their own rated movies, adding to the overall value proposition for movie lovers looking for a convenient and enjoyable streaming experience.

Overall, our team's commitment to providing a high-quality and reliable product that can meet the needs of users and administrators alike, along with our use of advanced features, sleek design, and cutting-edge technology, makes AXUM a comprehensive and robust movie streaming web app with the potential to revolutionize the way people interact with streaming media online.

5.2 Recommendation

Based on the scope of the AXUM project, the development team has successfully created a user-friendly and comprehensive movie streaming web app. Looking towards the future, the team envisions the system being available to a wider audience and potentially expanding its functionality. As a recommendation to those who may work on future iterations of the system, it would be beneficial to add a payment system for subscription to the service. This was not included in the current version due to changes in requirements, but would provide a convenient option for users who want to access premium content. Additionally, it is important to note that the system requires a stable internet connection in order to function effectively. We strongly advise implementing the system with this in mind to ensure a seamless streaming experience for users. Overall, AXUM has the potential to be a valuable and successful platform for movie streaming and we recommend exploring further development opportunities to enhance its functionality and reach.

References

[1] K. Mehta, A STUDY ON THE USAGE AND AWARENESS OF NETFLIX AMONG THE, vol. Volume 7, 2020.

[2] "“Dr. Krupa Mehta, Pritesh Kothiya et al. "a study on the usage and awareness of netflix among the youth" JETIR (2020):2349-5162".

[3] "Yu, Yinan, et al. The causal effect of video streaming on DVD Sales: Evidence from a natural experiment. Working Paper, The University of Hong Kong, 2017.".

[4] "Alawar, Mariam W., and Samy S. Abu-Naser. "CSS-Tutor: An intelligent tutoring system for CSS and HTML." (2017).".

[5] "Klimm, Marvin Christian. Design Systems for Micro Frontends-An Investigation into the Development of Framework-Agnostic Design Systems using Svelte and Tailwind CSS. Diss. Hochschulbibliothek der Technischen Hochschule Köln, 2021.".

[6] "Jensen, Simon Holm, Anders Møller, and Peter Thiemann. "Type analysis for javascript." SAS. Vol. 9. 2009.".

[7] "Holzner, Steven. Django: Visual QuickPro Guide. Peachpit Press, 2009.".

[8] "Kreibich, Jay. Using SQLite. " O'Reilly Media, Inc.", 2010.".

[9] "Pressman, Roger S. Software engineering: a practitioner's approach. Palgrave macmillan, 2001.".

[10] "Schach, Stephen R. Object-oriented and classical software engineering. Vol. 6. New York: McGraw-Hill, 2007.".