ONLINE SURVEY

Submitted By

SUDIP DATTA

Student Id: 221-35-880

Supervised By

Tapushe Rabaya Toma

Assistant Processor

This project report has been submitted in fulfilment of the requirements for the degree of **Bachelor of Science in Software Engineering**

APPROVAL

This project titled on "**Online Survey**", submitted by **Sudip Datta (ID: 221-35-880)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Chairman

Dr. S. M. Hasan Mahmud
Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1

A.H.M Shahariar Parvez
Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

Tapushe Rabaya Toma
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 3

Khalid Been md. Badruzzaman Biplob
Lecturer (Senior Scale)
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

External Examiner

Dr. Md Sazzadur Rahman
Professor
Institute of Information technology
Jahangirnagar University, Bangladesh

DAFFODIL INTERNATIONAL UNIVERSITY

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	:	Sudip Datta
Date of Birth	:	09 July 2001
Title	:	Online Survey
Academic Session	:	2022-2025

I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☑	OPEN ACCESS	I agree that my project to be published as online open access (Full Text)

I acknowledge that Daffodil International University reserves the following rights:

1. The Project is the Property of Daffodil International University.
2. The Library of Daffodil International University has the right to make copies of the Project for the purpose of research only.
3. The Library of Daffodil International University has the right to make copies of the Project for academic exchange.

Certified by:

Sudip
(Student's Signature)

(Supervisor's Signature)

Student ID: 221-35-880
Date: 24 December 2025

Name of Supervisor: Tapushe Rabaya Toma
Date: 24 December 2025

NOTE: * If the Project is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

PROJECT DECLARATION LETTER (OPTIONAL)

Librarian,
Daffodil International University,
Daffodil Smart City,
Ashulia.Dhaka,Bangladesh

Dear Sir,

CLASSIFICATION OF Project AS RESTRICTED

Please be informed that the following project is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name	Sudip Datta
Project Title	Online Survey

Reasons (i)

(ii)

(iii)

Thank you.

Yours faithfully,

(Supervisor's Signature)

Date: 24 December 2025

Stamp:

Note: This letter should be written by the supervisor and addressed to the Librarian, *Daffodil International University* with its copy attached to the thesis.

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this project and in my opinion, this project is adequate in terms of scope and quality for the award of the degree of Bachelor of Science.

(Supervisor's Signature)

Full Name : **Tapushe Rabaya Toma**

Position : Assistant Professor

Date : 24 December 2025

STUDENT'S DECLARATION

I hereby declare that the work in this project is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Daffodil International University or any other institution.

Sudip

(Student's Signature)

Full Name : **Sudip Datta**
ID Number : 221-35-880
Date : 24 October 2025

ONLINE SURVEY

SUDIP DATTA

Project submitted in fulfillment of the requirements
for the award of the degree of

Bachelor of Science/Master of Science

Department of Software Engineering

DAFFODIL INTERNATIONAL UNIVERSITY

OCTOBER 2025

ACKNOWLEDGEMENTS

At the start of this I thank God for letting me have the strengths, patience and opportunities to complete this project.

I Acknowledgements I would like to thank my supervisor Tapushe Rabaya Toma , Assistant Professor, Department of Software Engineering for his advice, valuable assistance and feedback to let me be participate in this work and fulfill this project. His forward moves and tips are very motivating for me.

I am indebted to the all faculty members of Department of Software Engineering for the support and advice they provided.

Last but not the least, heartfelt thanks to my dear parents for their unwavering love, unconditional support and prayers that is a base of my coming success.

DEDICATION

I therefore declare that I have done this project under the oversight of "**Tapushe Rabaya Toma", "Assistant Professor, Department of Software Engineering"**, Department of Software Engineering, Daffodil International University. Also declare that neither entire record nor any portion of this record has been submitted somewhere else for my degree.

ABSTRACT

Major companies in Bangladesh struggle to capture market trends due to insufficient consumer data, inefficient payment systems, and costly traditional survey methods. This gap hinders business growth and innovation. *TaskBuddy* addresses this by providing a web-based platform where users earn rewards by completing surveys, while companies gain real-time consumer insights. The platform uses HTML/CSS, PHP, and MySQL, ensuring secure payments (via Bkash/Nagad), real-time dashboards

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

UI	User Interface
DB	Database
API	Application Programming Interface
UML	Unified Modelling Language
SQL	Structured Query Language
HTTPS	Hypertext Transfer Protocol Secure
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JS	Java Script

CHAPTER 1 INTRODUCTION

1.1 Background

Surveys are a crucial means to collect human opinion and information, however traditional approaches are tedious, slow and need much manual work. In an age of online platforms, less and less people are willing to invest time in sharing their opinions (Dillman, 2007). TaskBuddy is the result and aims to make this easy for end-users and administrators.

The platform allows users to participate in surveys, reply and earn rewards while administrators the option to effectively setup and administer surveys on web. The point is to make survey collection more orderly, accurate and convenient for everybody.

1.1.1 Context and Relevance

The fast evolving digital landscape of Bangladesh has generated a need for quality consumer insights across sectors. Businesses need efficient, low-cost and reliable tools for gathering market feedback. Conventional survey methods are time consuming and resource-intensive, as well as highly restrictive. TaskBuddy aims to fill that gap by providing a digital platform for companies to post surveys and users to complete them for compensation.

1.1.2 Problem Identification

Majority of Bangladeshi companies still employ outdated and manual surveying process, thus low response rates and obtaining inaccurate insights. Global Survey platforms The current global survey websites used in Bangladesh do not have local payment options (bKash/Nagad) and very minimal payout for the participants. There is also no reliable Bangladeshi website that ensures payment security for survey sites. This leaves a large mismatch between what companies need for data and what users can earn.

1.1.3 Purpose and Justification

TaskBuddy Project Purpose: Development of secure, scalable, users friendly online survey platform that enables companies to generate surveys and analyze the results, and surveyees to earn real money. Local payments are being integrated, which ensures practical feasibility as well. This platform yields efficient data for organizations and equips surveyes with income systems.

1.1.4 Scope

For the purposes of this project you will be required to design and build a web-based, functional MVP (Minimum Viable Product). Features such as survey creation, taking part in surveys, saving answers, user authentication, role-based permissions system also. with a withdrawal interface and admin management. (Kaplan, The challenges and opportunities of social media. Business Horizons) The likes of mobile apps, AI powered analytics and more advanced automation will have to wait for phase two.

1.2 Project Planning and Initiation

Feasibility Study (Step-by-Step)

Phase 1 Preliminary Analysis & Project Scope Definition:

The current market practices and limitations of the survey were discussed. The remit of TaskBuddy was to fill the gap in low-cost local survey platforms with incentives for the respondents to participate

Phase 2 Market Feasibility Analysis (or Market Research):

There are over 45 million active internet users in Bangladesh, many of whom are students and young professionals looking for micro-earning opportunities. Digital trucking survey tools are becoming more and more necessary for businesses. Market research indicates a high demand for localized, gift-based site

Phase 3 Technical Feasibility Analysis:

The system will be programmed with HTML CSS, JS, PHP and MySQL. These tools take care of the full functional stack from creating surveys to making payments.

Phase 4 Financial Feasibility Analysis:

The work is financially viable as it uses open-source tools. You may earn commission depending on survey you complete, premium plans for corporations and data service packages.

1.3 Target User Profile and Tentative Elicitation Process

1.3.1 Target User

TaskBuddy serves three user groups:

• Survey Participants
• Company Clients
System Administrators

1.3.2 User profile

Table 1.3.2.1: User Profile for Participant

User Class	**Participant**
Type of user	Survey participants who complete surveys
	for monetary rewards.
Age range	18 – 50 years
Frequency of use	Daily or weekly
Mandatory	Optional; users join voluntarily.
Computer experience	Basic to intermediate computer knowledge.
Education	Secondary school, college, university students.
goal	Earn money by completing surveys.
Language skills	Basic English or Bangla reading skills.
Number of users	Large; potentially thousands nationwide.
Training	No formal training required
Others system use	Mobile banking apps (bKash/Nagad), social media.

Table 1.3.2.2: User Profile for Company

User Class	**Company**
Type of user	User who give survey to admin and also can create survey with admin
Age range	22 – 55 years
Frequency of use	Monthly or per project campaign.
Mandatory	Optional; companies register as needed.
Computer experience	Intermediate to advanced.
Education	Bachelor degree or higher in business, marketing or research.
goal	Collect real-time consumer insights.

Language skills	Professional English and Bangla.
Number of users	Such a organization
Training	Minimal training
Others system use	email tools.
Way of working	Office or remote use by authorized company staff.

Table 1.3.2.3: User Profile for Admin

User Class	**Admin**
Type of user	Technical system administrators and support team.
Age range	22 – 45 years
Frequency of use	Daily system monitoring and maintenance.
Mandatory	Yes, essential role for system operation.
Computer experience	Advanced technical skills (web systems, DBMS, security).
Education	Bachelor degree in CSE/Software Engineering or related field.
goal	Ensure smooth platform operation and handle user issues.
Language skills	Fluent Bangla and English.
Number of users	Small admin team (2–5 people).
Training	System architecture and backend training required.
Others system use	Database tools, monitoring dashboards, logs, server panels.
Way of working	Technical maintenance with structured workflow.

1.3.3 Elicitation Process

The following elicitation techniques were used:

1 Interviews with companies/panel for their needs
2 Online user survey participant's expected rewards
3 Competitor analysis (SurveyMonkey, Swagbucks)
4 Prototype testing feedback

1.3.3.1 Stakeholder Interviews

We also carried interviews with the developers, frequent participants in the surveys and moderators of the platform. These interviews revealed the pain points with traditional survey collection, including manual data entry, low response rates, slow processing of results and managing incentives to study participants. The feedback received from these stakeholders assisted in prioritizing the core functionalities required by the TaskBuddy system (e.g., convener award elimination, survey editor and responsive management).

1.3.3.2 Surveys

The instrument was administered to individuals with a high likelihood of participating in online surveys and organizations using surveys for data collection.

User Assumptions The user centred surveys were useful in determining that users expect easy access to rewards, transparent rewards, simple question answering and mobile platforms.

In the organization-oriented polls, their demands were: organized storage of responses, quick analytics and the possibility to reach an appropriate target audience.

All this data I pasted in the preceding paragraphs are part of what drives TaskBuddy.

1.3.3.3 Observation

Observation sessions were run on how organizations are currently gathering feedback with handwritten forms, Google forms, and spreadsheets. These sessions revealed workflow problems such as elimination of data to the trash, manual classification, challenges tracking participants and in place reward mechanism. Watching these practices in the wild, polished up the nitty-gritty of core TaskBuddy workflow: survey creation, participant engagement, response management, and reward fulfillment.

1.3.3.4 Competitor Analysis

The most frequently provided features in existing online survey tools such as Google Forms, SurveyMonkey and Toluna were identified.

This inspection helped to reveal their weaknesses and the place for TaskBuddy to address in it.

Some of the missing or weak features are: Reward-based
participation system

Transparent withdrawal and wallet features

Built-in support ticket system

Single Dashboard for Survey Creators and Takers

Localized payment and withdrawal options

1.3.3.5 Use Case Scenario

Case scenarios were identified from surveyed practice and user activity. These distributions are the windows on users' daily use of the platform.

They include:

A user look into available surveys, fill them and earn rewards in their wallet

TaraGuns A user who asks to cash out upon reaching the required balance.

Admin - adding new surveys, managing questions and processing withdrawal requests.

A user opening a support ticket and the admin responding in your dashboard.

Survey makers looking at submitted answers and basic statistics about them

These use case cases greatly contributed to formulating a clear image of the system features.

1.3.3.6 Prototyping and Feedback

1. Templates for login, dashboard, take survey and admin were built. These prototypes were demonstrated to a sample of users and designers for feedback.
2. Their feedback helped improve:
3. Layout and user flow
4. Survey answering experience
5. Navigation clarity between pages
6. Wallet and withdrawal visibility
7. Display of survey information
8. This friendly user input informed refinements to the interface and made the system more intuitive for everyone.

1.4 Project Block Diagram

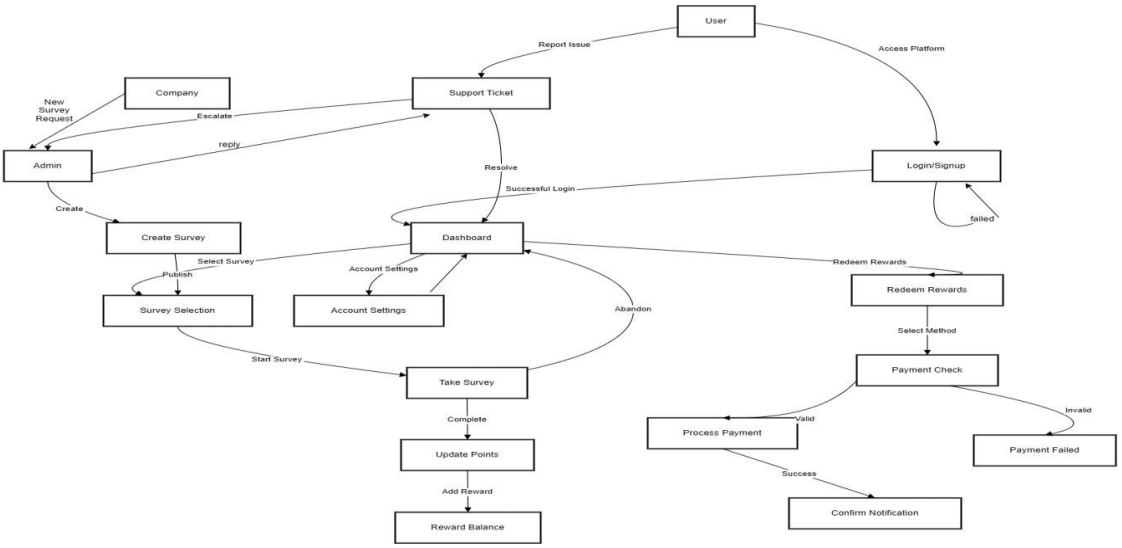


Figure 1: System Block Diagram

1.5 System Requirements

1.5.1 Hardware Requirements

- 4GB RAM minimum
- Stable internet
- Standard processor

1.5.2 Software Requirements

- PHP 8+
- MySQL
- Apache Server
- HTML, CSS, JS

1.5.3 Constraints and Dependencies

- Dependent on stable internet
- Payment API dependency
- Limited time and resources

1.6 Project Scheduling

The 16-week schedule includes:

- Requirement analysis
- Design
- Development
- Testing
- Deployment
- Documentation

1.7 Summary

This chapter introduces the background, purpose, feasibility, users, system requirements, and scheduling of TaskBuddy. It provides the foundation for further system analysis and design.

CHAPTER 2 DESIGN AND IMPLEMENTATION

2.1 Introduction

Major companies in Bangladesh struggle to capture market trends due to insufficient consumer data, inefficient payment systems, and costly traditional survey methods. This gap hinders business growth and innovation. *TaskBuddy* addresses this by providing a web-based platform where users earn rewards by completing surveys, while companies gain real-time consumer insights. The platform uses HTML/CSS, PHP, JS and MySQL, ensuring secure payments (via Bkash/Nagad), real-time dashboards

2.2 Functional Requirements

It need for declear the main functional work of Taskuddy

FR01	**Sign Up**
Description	Before using the Online Survey Users must be registered first
Stakeholder	participants(users) ,Admin

FR02	**Log In**
Description	Registered users must log in to take part in surveys and manage their accounts.
Stakeholder	Survey participants(users), Admin

FR03	**Create survey**
Description	Companies can conduct surveys to gather insights from consumers.

Stakeholder	Company (Panel Provider) ,Admin

FR04	**Participate Survey**
Description	Users can browse and select available surveys to complete, and earn rewards for participation.
Stakeholder	participants(users)

FR05	**Edit Survey**
Description	Admin can edit survey when necessary.
Stakeholder	Admin

FR06	**Delete Survey**
Description	Admin can delete survey when necessary
Stakeholder	Admin

FR07	**Delete User**
Description	Admin can delete user for suspicious activity.

Stakeholder	Admin

FR08	**Response Submission**
Description	User response store in admin side and admin can view that and save that.
Stakeholder	Admin

FR09	**Withdrawal Request Handling**
Description	Admin can handle participant withdraw request by approve or reject
Stakeholder	Admin

FR10	**Withdraw Money**
Description	User can withdraw their money by bikash, nagad.
Stakeholder	Participant

FR11	**View Withdraw History**
Description	User can view their withdraw history
Stakeholder	Participant

FR12	Support submission
Description	Submit support ticket for survey issues, payment delays, account problem and many more
Stakeholder	Participants

FR13	View Ticket
Description	Participant can view his ticket position and can make reply of admin response.
Stakeholder	Participants

FR14	Manage Support
Description	Admin can see user ticket and make reply
Stakeholder	Admin

FR15	Update User Profile
Description	Users can manage their profiles, update personal information, and track their survey participation and rewards.

Stakeholder	Participants

FR16	**Change Password**
Description	Users can change their password by giving current password and new password.
Stakeholder	Participants

FR17	**Admin Account Setup**
Description	Admin can create new admin for account management and many more.
Stakeholder	Admin

FR18	**Log Out**
Description	Users can log out of this system by using this option. The login will be saved if the user wants to save their account.
Stakeholder	Partipants, Admin

2.3 Non-Functional Requirements

Security Access Control

1. Role-Based Access Control (RBAC) for users/companies/admins
2. IP whitelisting for admin operations

Authentication

1. Password complexity (8+ chars, special characters)
2. Session timeout (15 min inactivity)

Data Confidentiality

1. Encryption for payments and personal data
2. GDPR-compliant anonymization

System Reliability

1. 99.9% uptime for core features

2.3.1 Performance <2s

1. survey loading time
2. Support 1,000
3. concurrent users

2.3.2 Reliability

99.9% uptime for core features

2.3.3 Usability

Mobile-responsive design

2.3.4 Compliance

1. A secure payment system
2. Bangladesh Data Protection Act

2.4 Object-oriented System design using UML

2.4.1 Use Case Diagram

Figure 2: Use case Diagram

2.4.2 Case Description

Table 2.4.2.1: Case Description-01: Sign Up

<table>
<tr><td>Use Case</td><td>Sign Up</td></tr>
<tr><td>Goal</td><td>Users can Sign Up in their system</td></tr>
<tr><td>Precondition</td><td>Users must visit website for sign up</td></tr>
<tr><td>Success End Condition</td><td>Successfully sign up</td></tr>
<tr><td>Failed End Condition</td><td>Submission Not Submitted</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Participants, Admin

System</td></tr>
<tr><td>Trigger</td><td>User will request a Sign Up form to fill up</td></tr>
<tr><td>Description / Main

Success

Scenario</td><td><table>
<tr><td>1.</td><td>Press “Sign Up” Button</td></tr>
<tr><td>2.</td><td>Provide Sign Up form</td></tr>
<tr><td>3.</td><td>Enter Information</td></tr>
<tr><td>4.</td><td>Press “Create Account” Button.</td></tr>
<tr><td>5.</td><td>Information saved</td></tr>
<tr><td>6.</td><td>Successfully sign up</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>The user Did not fill up the details!</td></tr>
<tr><td></td><td>4.1.a. Fill Up the Box.</td></tr>
<tr><td>5.1</td><td>The system did not respond</td></tr>
<tr><td></td><td>5.1.a. Try again</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>User must Fill up all information very carefully</td></tr>
</table>

Table 2.4.2.2: Case Description-02: Log In

<table>
<tr><td>Use Case</td><td>Log In</td></tr>
<tr><td>Goal</td><td>Users can log in to the system.</td></tr>
<tr><td>Precondition</td><td>Users must Sign Up Before</td></tr>
<tr><td>Success End Condition</td><td>Successfully log In</td></tr>
<tr><td>Failed End Condition</td><td>Details Not Valid</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Participant, Admin

System</td></tr>
<tr><td>Trigger</td><td>User will request a log In form to Fill Up</td></tr>
<tr><td>Description / Main

Success

Scenario</td><td>
<table>
<tr><td>1.</td><td>Press “log In” Button</td></tr>
<tr><td>2.</td><td>Provide log In form</td></tr>
<tr><td>3.</td><td>Enter Information</td></tr>
<tr><td>4.</td><td>Press “log In” Button.</td></tr>
<tr><td>5.</td><td>Information saved</td></tr>
<tr><td>6.</td><td>log In successfully</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows</td><td>
<table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>Information not valid</td></tr>
<tr><td></td><td>4.1.a Enter valid information</td></tr>
<tr><td>5.1</td><td>System Response</td></tr>
<tr><td></td><td>5.1.a.Try again</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>The user Will fill up all the details with valid information</td></tr>
</table>

Table 2.4.2.3: Case Description-03: Create survey

<table>
<tr><td>Use Case</td><td>Create Survey</td></tr>
<tr><td>Goal</td><td>Create survey for participant</td></tr>
<tr><td>Precondition</td><td>User must sign in as Admin</td></tr>
<tr><td>Success End Condition</td><td>Successfully created survey</td></tr>
<tr><td>Failed End Condition</td><td>Survey creation failded</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Admin, Company

System</td></tr>
<tr><td>Trigger</td><td>User will request a create survey form to fill up</td></tr>
<tr><td>Description / Main

Success

Scenario</td><td><table>
<tr><td>1.</td><td>Press “Create Survey” Button</td></tr>
<tr><td>2.</td><td>Provide create survey form</td></tr>
<tr><td>3.</td><td>Enter survey details</td></tr>
<tr><td>4.</td><td>Press “Create Survey” Button.</td></tr>
<tr><td>5</td><td>The system saves the details</td></tr>
<tr><td>6</td><td>Successfully Created Survey</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>The user did not fill up all details!</td></tr>
<tr><td></td><td>3.1.a. Provide all required filed carefully.</td></tr>
<tr><td>5.1</td><td>The system Doesn’t save the details.</td></tr>
<tr><td></td><td>6.1.a. Try Again</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The user Will fill up all the details in 40 minutes.</td></tr>
</table>

Table 2.4.2.4: Case Description-04: participate Survey

<table>
<tr><td>Use Case</td><td>Participate Survey</td></tr>
<tr><td>Goal</td><td>Users can earn money by complete survey</td></tr>
<tr><td>Precondition</td><td>Users must sign up, then login and also must completed ID verification.</td></tr>
<tr><td>Success End Condition</td><td>Successfully Complete survey</td></tr>
<tr><td>Failed End Condition</td><td>Not allow for that survey.</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Participant

System</td></tr>
<tr><td>Trigger</td><td>User will request for a survey</td></tr>
<tr><td>Description / Main

Success

Scenario</td><td><table>
<tr><td>1.</td><td>Click Start Task</td></tr>
<tr><td>2.</td><td>Provide survey details</td></tr>
<tr><td>3.</td><td>Enter survey response</td></tr>
<tr><td>4.</td><td>Press Submit Survey</td></tr>
<tr><td>5.</td><td>The system saves the details</td></tr>
<tr><td>6.</td><td>shows survey completed successfully</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>The user Did not fill up the details!</td></tr>
<tr><td></td><td>4.1.a. Fill Up the Box.</td></tr>
<tr><td>5.1</td><td>The system Doesn’t save the details.</td></tr>
<tr><td></td><td>6.1.a. Try Again</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The user Will fill up survey response very carefully</td></tr>
</table>

Table 2.4.2.5: Case Description-05: Edit Survey

<table>
<tr><td>Use Case</td><td>Edit Survey</td></tr>
<tr><td>Goal</td><td>Edit survey for changing company survey requirement</td></tr>
<tr><td>Precondition</td><td>User must sign in as Admin.</td></tr>
<tr><td>Success End Condition</td><td>Successfully edit survey</td></tr>
<tr><td>Failed End Condition</td><td>Edit survey submission is not responding</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Admin

System</td></tr>
<tr><td>Trigger</td><td>User will request a edit survey form to fill up</td></tr>
<tr><td>Description / Main

Success

Scenario</td><td>
<table>
<tr><td>1.</td><td>Press “Edit Survey” Button</td></tr>
<tr><td>2.</td><td>Provide edit survey form</td></tr>
<tr><td>3.</td><td>Enter editing details in form</td></tr>
<tr><td>4.</td><td>Press “Edit Survey” Button.</td></tr>
<tr><td>5.</td><td>Information saved</td></tr>
<tr><td>6.</td><td>The system saves the details and shows them

Successfully Edited survey</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows</td><td>
<table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>The user Did not fill up the details!</td></tr>
<tr><td></td><td>4.1.a. Fill Up all required field</td></tr>
<tr><td>5.1</td><td>The system did not respond</td></tr>
<tr><td></td><td>5.1.a. Show Error Message.</td></tr>
<tr><td>6.1</td><td>The system Doesn’t save the details.</td></tr>
<tr><td></td><td>6.1.a. Notification: “Details did not Save”</td></tr>
</table>
</td></tr>
</table>

Quality Requirements	Admin should edit survey very carefully with valid information

Table 2.4.2.6: Case Description 06: **Delete Survey**

Use Case	Delete survey
Goal	Admin can delete survey
Precondition	Admin must sign in for delete survey
Success End Condition	Successfully delete survey
Failed End Condition	Delete survey is not successfully complete
Primary Actors: Secondary Actors:	Admin System
Trigger	User will request a delete form for delete survey
Description / Main Success Scenario	1. Press “Manage Survey” Button 2. Provide manage survey dashboard 3. Press delete survey 4. Information saved by System 5. The system saves the details and shows them Successfully Edit Delete survey
Alternative Flows	1.1 System Error 1.1.a. Try Again!! 4.1 System is not responding 4.1.a. Reload
Quality Requirements	The admin delete survey when necessary.

Table 2.4.2.7: Case Description-07: Delete User

<table>
<tr><td>Use Case</td><td>Delete User</td></tr>
<tr><td>Goal</td><td>Manage user for user benefit and TaskBuddy reputation</td></tr>
<tr><td>Precondition</td><td>User must sign in as admin and need for manage participant management</td></tr>
<tr><td>Success End Condition</td><td>Successfully manage user like, delete user</td></tr>
<tr><td>Failed End Condition</td><td>Management is not successfully complete</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Admin

System</td></tr>
<tr><td>Trigger</td><td>User will request a User Management form to fill up</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>Press “User Management” Button</td></tr>
<tr><td>2.</td><td>Provide user management dashboard</td></tr>
<tr><td>3.</td><td>Manage user by delete user and see user activity</td></tr>
<tr><td>4.</td><td>Information saved by System</td></tr>
<tr><td>5.</td><td>The system saves the details and shows them

Successfully Edit User/Delete user</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>System is not responding</td></tr>
<tr><td></td><td>4.1.a. Reload</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The admin delete user for his suspicious activity</td></tr>
</table>

Table 2.4.2.8: Case Description-08: Response Submission

<table>
<tr><td>Use Case</td><td>Response Submission</td></tr>
<tr><td>Goal</td><td>Submit participant survey response for giving to the company.</td></tr>
<tr><td>Precondition</td><td>Survey response limit completed</td></tr>
<tr><td>Success End Condition</td><td>Successfully save response</td></tr>
<tr><td>Failed End Condition</td><td>Information is not save</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Admin

System</td></tr>
<tr><td>Trigger</td><td>User will request a response submission form for save.</td></tr>
<tr><td>Description / Main

Success

Scenario</td><td><table>
<tr><td>1.</td><td>Press “View Response” Button</td></tr>
<tr><td>2.</td><td>Show survey response</td></tr>
<tr><td>3.</td><td>Press Download survey response</td></tr>
<tr><td>3.</td><td>Successfully download</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>3.1</td><td>Response des not save by system</td></tr>
<tr><td></td><td>4.1.a. Reload</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>Admin download response when survey limit reach</td></tr>
</table>

Table 2.4.2.9: Case Description-09: Withdrawal Request Handling

<table>
<tr><td>Use Case</td><td>Withdrawal Request Handling</td></tr>
<tr><td>Goal</td><td>Withdrawal request handling for payment user and reject withdrawal request for prevent fraud also</td></tr>
<tr><td>Precondition</td><td>Users request withdraw request</td></tr>
<tr><td>Success End Condition</td><td>Successfully withdraw money</td></tr>
<tr><td>Failed End Condition</td><td>System is not response</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Admin

System</td></tr>
<tr><td>Trigger</td><td>User will request a withdrawal request form for withdraw handling</td></tr>
<tr><td>Description / Main

Success

Scenario</td><td>
<table>
<tr><td>1.</td><td>Press “Withdrawals” Button</td></tr>
<tr><td>2.</td><td>Provide Withdrawals form</td></tr>
<tr><td>3.</td><td>See user withdrawal request information</td></tr>
<tr><td>4.</td><td>Press Approve or Reject button</td></tr>
<tr><td>5.</td><td>Show successfully notify user for approve or reject</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows</td><td>
<table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>The system does not response</td></tr>
<tr><td></td><td>4.1.a. Reload</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>Admin first check his survey response and activity for approve or reject.</td></tr>
</table>

Table 2.4.2.10: Case Description-10: Withdraw Money

Use Case	Withdraw Money	
Goal	Users can withdraw his money	
Precondition	User must have balance and want to withdraw	
Success End Condition	Successfully withdraw money	
Failed End Condition	System Error	
Primary Actors: Secondary Actors:	Participant System	
Trigger	User will request a registration form to fill up	
Description / Main Success Scenario	1.	Press “withdraw money” Button
	2.	Provide withdraw money page
	3.	Write amount and Bikash/Nagad/Rocket number
	4.	Press Submit Withdrawal Request
	5.	Information save
	6.	Show withdraw request successfully sent
Alternative Flows	1.1	System Error
		1.1.a. Try Again!!
	4.1	Required for empty field
		4.1.a. write amount and number again
	5.1	System validate error
		5.1.a. Try again
Quality Requirements	The user must enter his number very carefully	

Table 2.4.2.11: Case Description-11: Withdraw History

Use Case	Withdraw History
Goal	Users can view his withdraw history
Precondition	User must have withdraw first and need login
Success End Condition	Successfully view withdraw history
Failed End Condition	System Error
Primary Actors: Secondary Actors:	Participant System
Trigger	User will request a registration form to fill up
Description / Main Success Scenario	1. Press “withdraw History” Button 2. Provide withdraw History page
Alternative Flows	1.1 System Error 1.1.a. Try Again!!
Quality Requirements	The user see carefully and check for approve or reject

Table 2.4.2.12: Case Description-12: Support Submission

Use Case	Support Submission
Goal	Users can get help by admin
Precondition	User must log in and need for support
Success End Condition	Successfully submit support ticket
Failed End Condition	Support Submission Failed
Primary Actors: Secondary Actors:	Participant System

<table>
<tr><td>Trigger</td><td>User will request a support submission form to fill up</td></tr>
<tr><td>Description / Main

Success

Scenario</td><td><table>
<tr><td>1.</td><td>Press “Open Ticket” Button</td></tr>
<tr><td>2.</td><td>Provide support form</td></tr>
<tr><td>3.</td><td>Enter problem subject and details</td></tr>
<tr><td>4.</td><td>Press “Submit Ticket” Button.</td></tr>
<tr><td>5.</td><td>Save details</td></tr>
<tr><td>6</td><td>Successfully submit ticket</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>The user Did not fill up the details!</td></tr>
<tr><td></td><td>4.1.a. Checked By the system & Notify by Fill up all details.</td></tr>
<tr><td>5.1</td><td>The system did not respond</td></tr>
<tr><td></td><td>5.1.a Try Again</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The user Will fill his problem with details</td></tr>
</table>

Table 2.4.2.13: Case Description-13: View Ticket

Use Case	View Ticket
Goal	Users can see his ticket history
Precondition	User must make ticket past for see his history
Success End Condition	Successfully see ticket history and make reply
Failed End Condition	Support Submission Failed
Primary Actors: Secondary Actors:	Participant System
Trigger	User will request a Ticket History form to fill up

<table>
<tr><td>Description / Main

Success

Scenario</td><td><table>
<tr><td>1.</td><td>Press “Ticket History” Button</td></tr>
<tr><td>2.</td><td>Provide ticket history form</td></tr>
<tr><td>3.</td><td>Press view details button</td></tr>
<tr><td>4.</td><td>Reply admin response</td></tr>
<tr><td>5.</td><td>Send reply</td></tr>
<tr><td>6</td><td>Successfully reply the ticket</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>3.1</td><td>The user Did not fill up the details!</td></tr>
<tr><td></td><td>4.1.a. Reload</td></tr>
<tr><td>5.1</td><td>Required write something</td></tr>
<tr><td></td><td>5.1. reply admin response</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The user view ticket carefully and reply admin response</td></tr>
</table>

Table 2.4.2.14: Case Description-14: Manage Support

Use Case	Manage Support
Goal	Reply user support ticket
Precondition	User must sign in as admin and need for reply support ticket.
Success End Condition	Successfully reply user ticket
Failed End Condition	Reply submission fail
Primary Actors: Secondary Actors:	Admin System
Trigger	User will request a Support Ticket form for reply

<table>
<tr><td>Description / Main

Success

Scenario</td><td><table>
<tr><td>1.</td><td>Press “Support Ticket ” Button</td></tr>
<tr><td>2.</td><td>Provide Support request list and reply</td></tr>
<tr><td>3.</td><td>Press someone ticket view and reply button</td></tr>
<tr><td>4.</td><td>Reply his support</td></tr>
<tr><td>5.</td><td>Press send reply button</td></tr>
<tr><td>6.</td><td>The system saves the details and shows them !!!

Successfully Send reply!!! Notify</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>3.1</td><td>System is not responding</td></tr>
<tr><td></td><td>4.1.a. Reload</td></tr>
<tr><td>5.1</td><td>The system show fill up the filed</td></tr>
<tr><td></td><td>5.1.a. Write support reply</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The user Will fill up all the details in 30 minutes.</td></tr>
</table>

Table 2.4.2.15: Case Description-15: Update User Profile

Use Case	Update User Profile
Goal	Users can update his user profile.
Precondition	Users must install login and also need for change
Success End Condition	Notification: !!!Successfully Update profile!!!
Failed End Condition	Notification: “Submission Not Submitted”
Primary Actors: Secondary Actors:	Customer System

<table>
<tr><td>Trigger</td><td>User will request a Update Profile form to fill up</td></tr>
<tr><td>Description / Main
Success
Scenario</td><td><table>
<tr><td>1.</td><td>Press “Update Profile” Button</td></tr>
<tr><td>2.</td><td>Provide Update Profile form</td></tr>
<tr><td>3.</td><td>Enter update Information</td></tr>
<tr><td>4.</td><td>Press “Update Profile” Button.</td></tr>
<tr><td>5.</td><td>Information saved</td></tr>
<tr><td>6.</td><td>The system saves the details and shows them !!!
Successfully Update Profile!!! Notify</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>The user Did not fill up any details!</td></tr>
<tr><td></td><td>4.1.a. Update any information</td></tr>
<tr><td>5.1</td><td>The system did not respond</td></tr>
<tr><td></td><td>5.1.a. Try Again</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The user Will fill with true information when update.</td></tr>
</table>

Table 2.4.2.16: Case Description-16: Change Password

Use Case	Change Password
Goal	Users can change their password for secure their account
Precondition	Users must be login in and need for change password.
Success End Condition	Successfully Change Password
Failed End Condition	Password is not Change for system error

<table>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Participant

System</td></tr>
<tr><td>Trigger</td><td>User will request a change password form to fill up</td></tr>
<tr><td>Description / Main

Success

Scenario</td><td><table>
<tr><td>1.</td><td>Press “Change password” Button</td></tr>
<tr><td>2.</td><td>Provide change password form</td></tr>
<tr><td>3.</td><td>Enter current password and new password for 2 times</td></tr>
<tr><td>4.</td><td>Press “Change Password” Button.</td></tr>
<tr><td>5.</td><td>Information saved</td></tr>
<tr><td>6.</td><td>The system saves the details and shows them

Successfully Change password</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>Current password invalid</td></tr>
<tr><td></td><td>4.1.a. Fill give current password carefully and other 2 box</td></tr>
<tr><td>5.1</td><td>The system did not respond</td></tr>
<tr><td></td><td>5.1.a. Try Again</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>User must enter a strong new password for security</td></tr>
</table>

Case Description-17: Add Admin

Use Case	Add Admin
Goal	Main Admin can add sub admin for user management and many more
Precondition	Users must login in and need to add admin.
Success End Condition	Notification: !!!Successfully Create Admin!!!

Failed End Condition	Notification: “Admin not created ”
Primary Actors: Secondary Actors:	Admin System
Trigger	User will request Add Admin form to fill up
Description / Main Success Scenario	1. Press “Add Admin” Button 2. Provide Add admin form 3. Enter admin username and password 4. Press “Add Admin” Button. 5. Information saved 6. The system saves the details and shows them !!! Successfully Add Admin!!! Notify
Alternative Flows	1.1 System Error 1.1.a. Try Again!! 4.1 The user Did not fill up the details! 4.1.a. Fill up Again with valid information 5.1 The system did not respond 5.1.a. Try Again
Quality Requirements	Main admin should add admin when necessary

Table 2.4.2.17: Case Description-18: Log Out

Use Case	Log Out
Goal	Users can Log Out from the Taskbuddy website
Precondition	Users must be a valid user and sign in before for sign out.
Success End Condition	Notification: !!!Successfully Sign Out!!!
Failed End Condition	Notification: “System error for sign Out”

<table>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Participant

System</td></tr>
<tr><td>Trigger</td><td>User will go and press for sign out button.</td></tr>
<tr><td>Description / Main

Success

Scenario</td><td><table><tr><td>1.</td><td>Press “Log Out” Button</td></tr><tr><td>2.</td><td>System remove the current season</td></tr><tr><td>3.</td><td>Show Successfully log out</td></tr></table></td></tr>
<tr><td>Alternative Flows</td><td><table><tr><td>1.1</td><td>System Error</td></tr><tr><td></td><td>1.1.a. Try Again!!</td></tr></table></td></tr>
<tr><td>Quality Requirements</td><td>The user log out when necessary.</td></tr>
</table>

2.4.3 Activity Diagram

For Sign Up

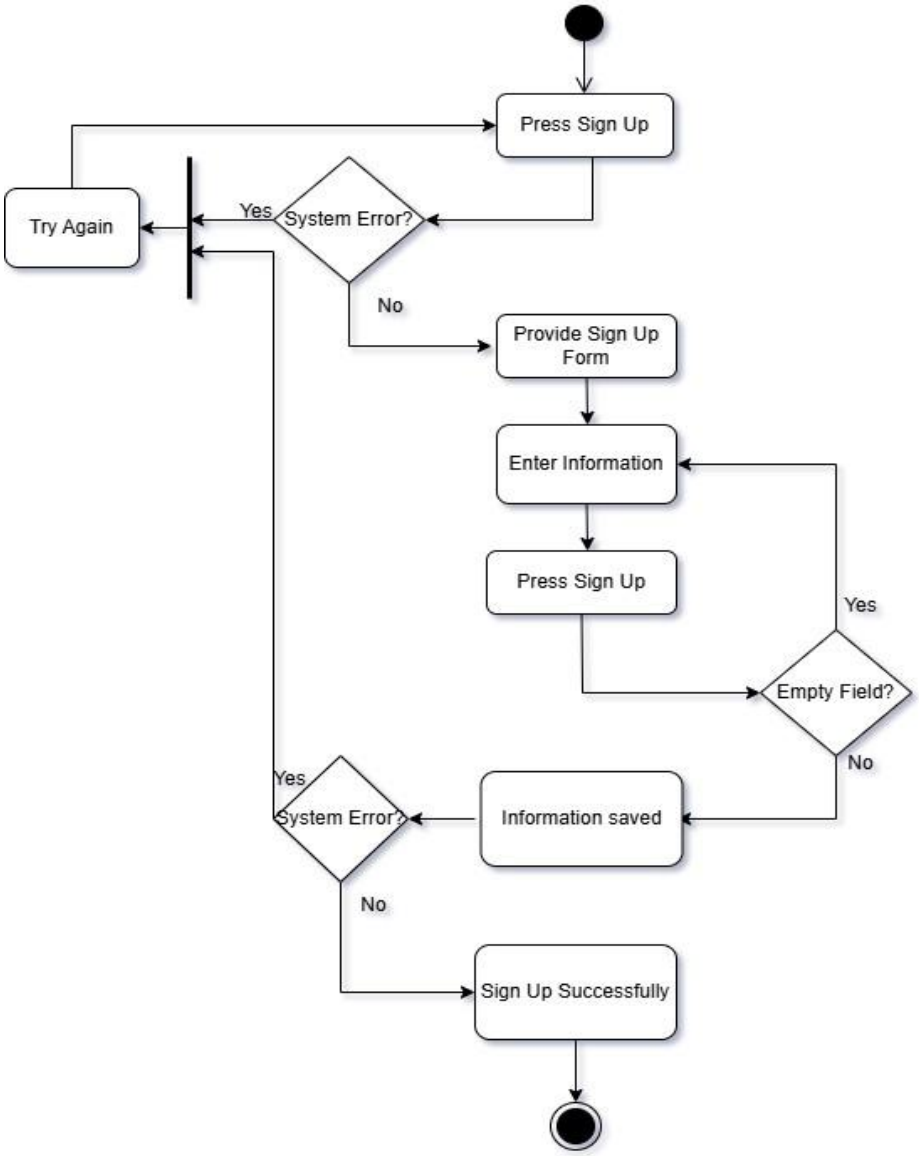


Figure 2.1 Activity diagram for Sign Up

For Log In

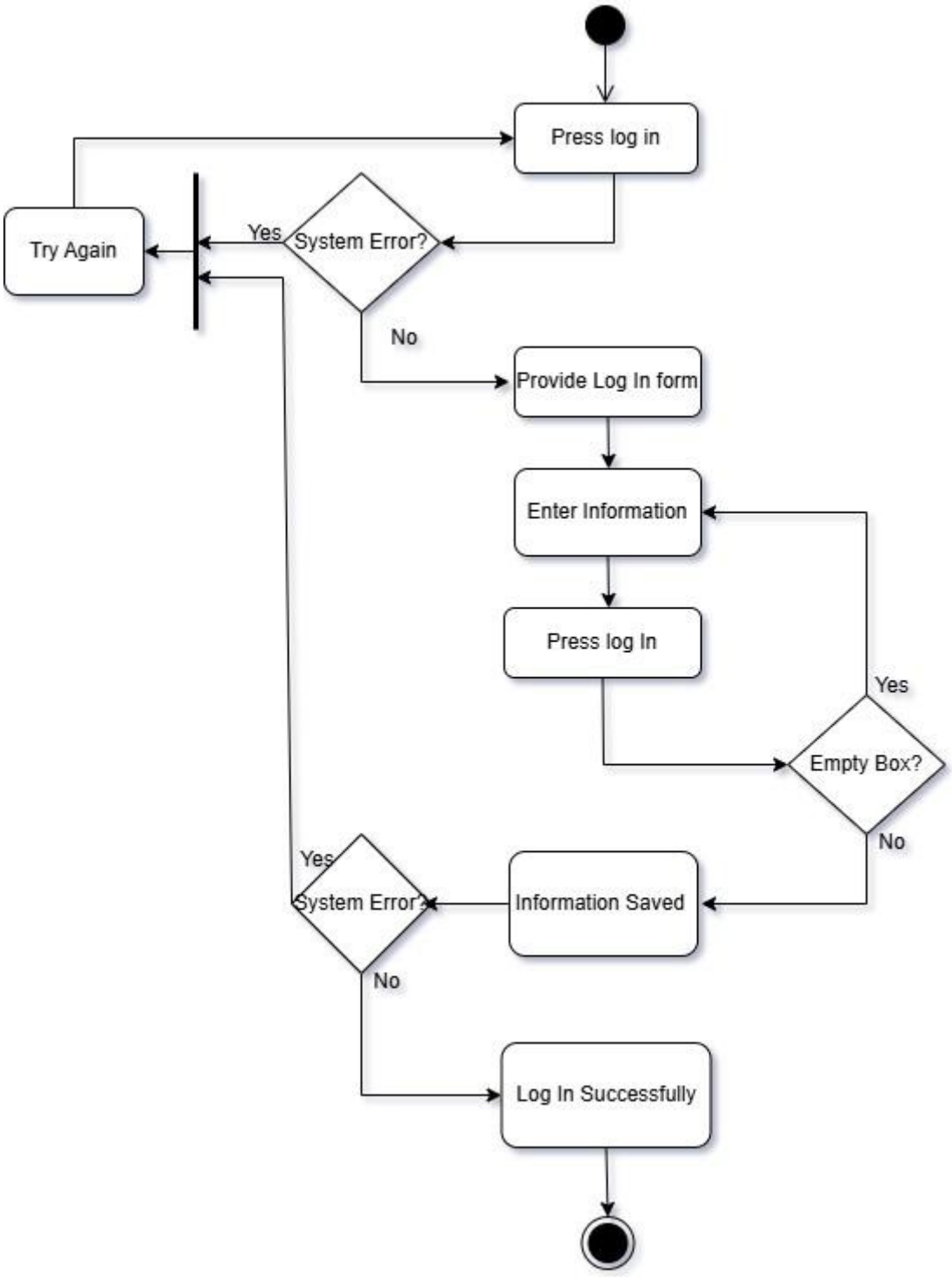


Figure 2.2 Activity diagram for log in

For Survey Creation

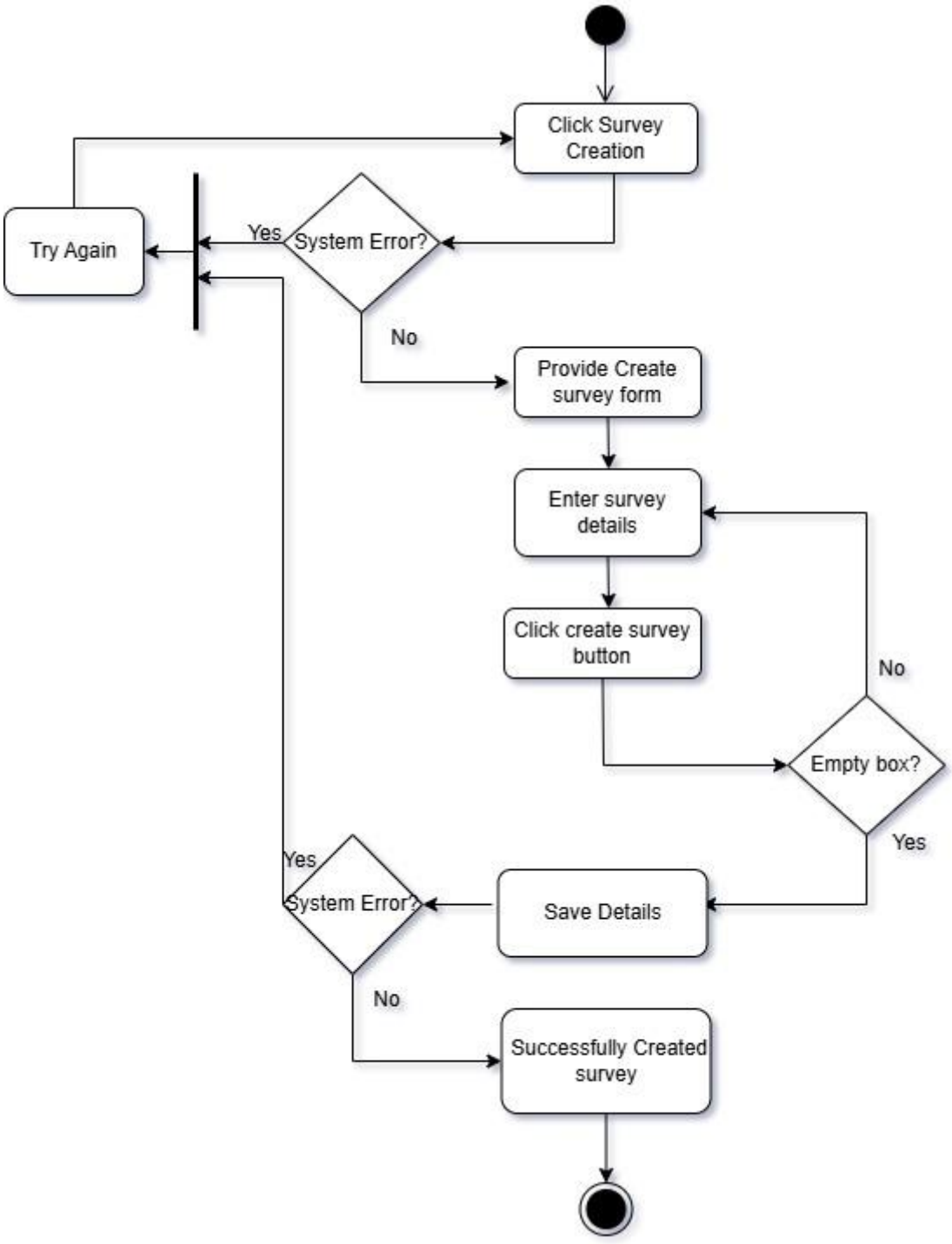


Figure 2.3 Activity diagram for Survey Creation

For Survey Participation

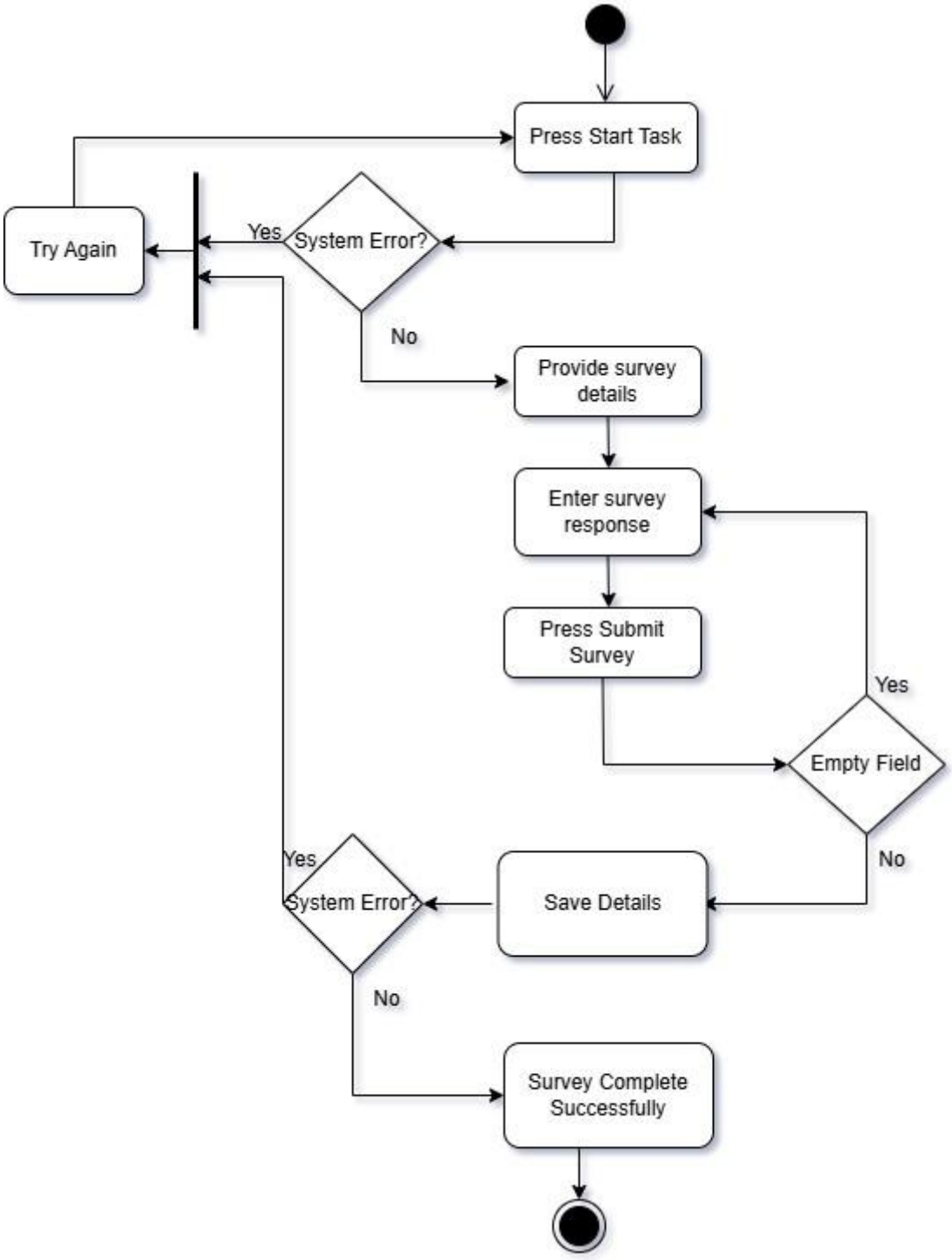


Figure 2.4 Activity diagram for Survey Participation

For Survey Editing

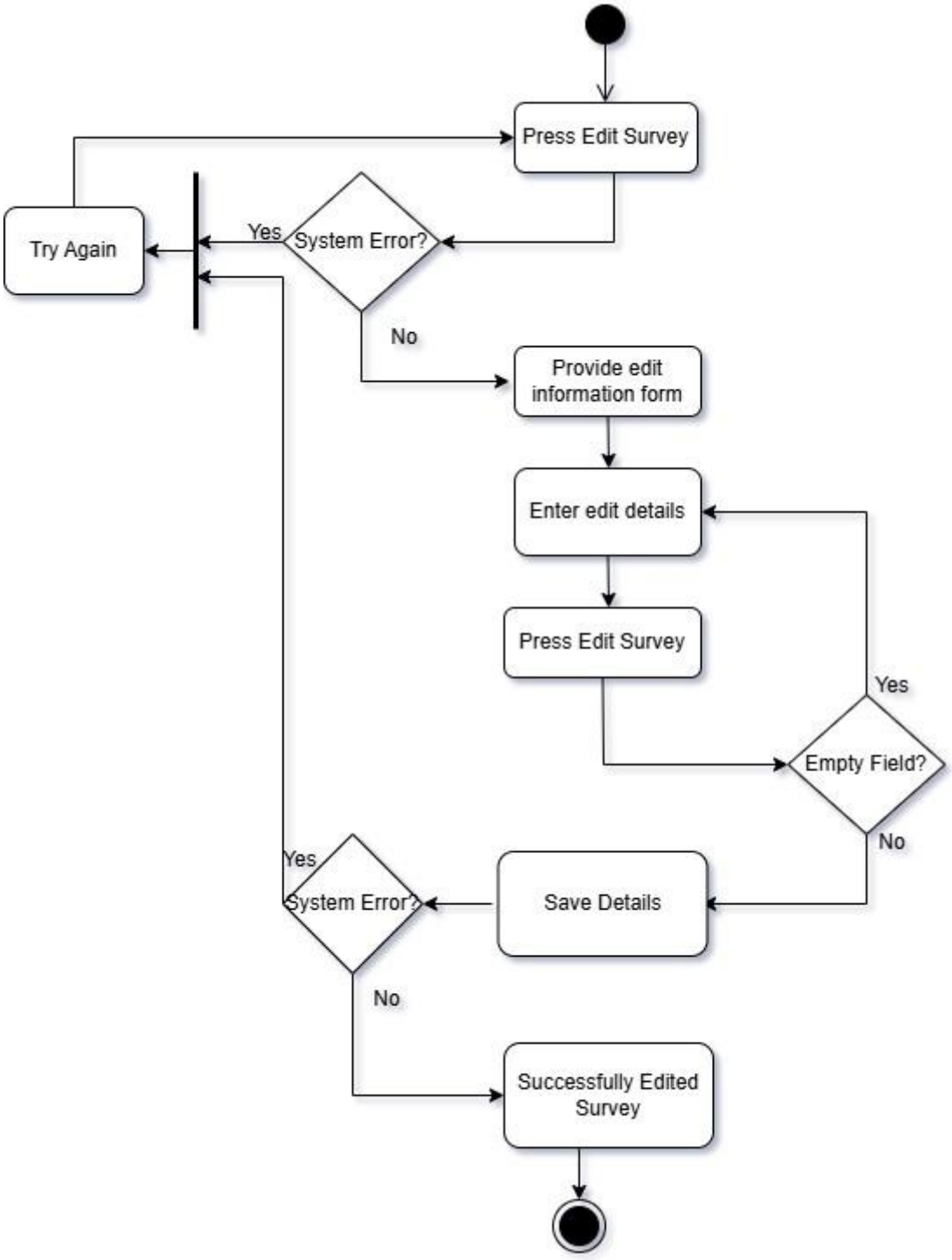


Figure 2.5 Activity diagram for Survey Editing

For User Management

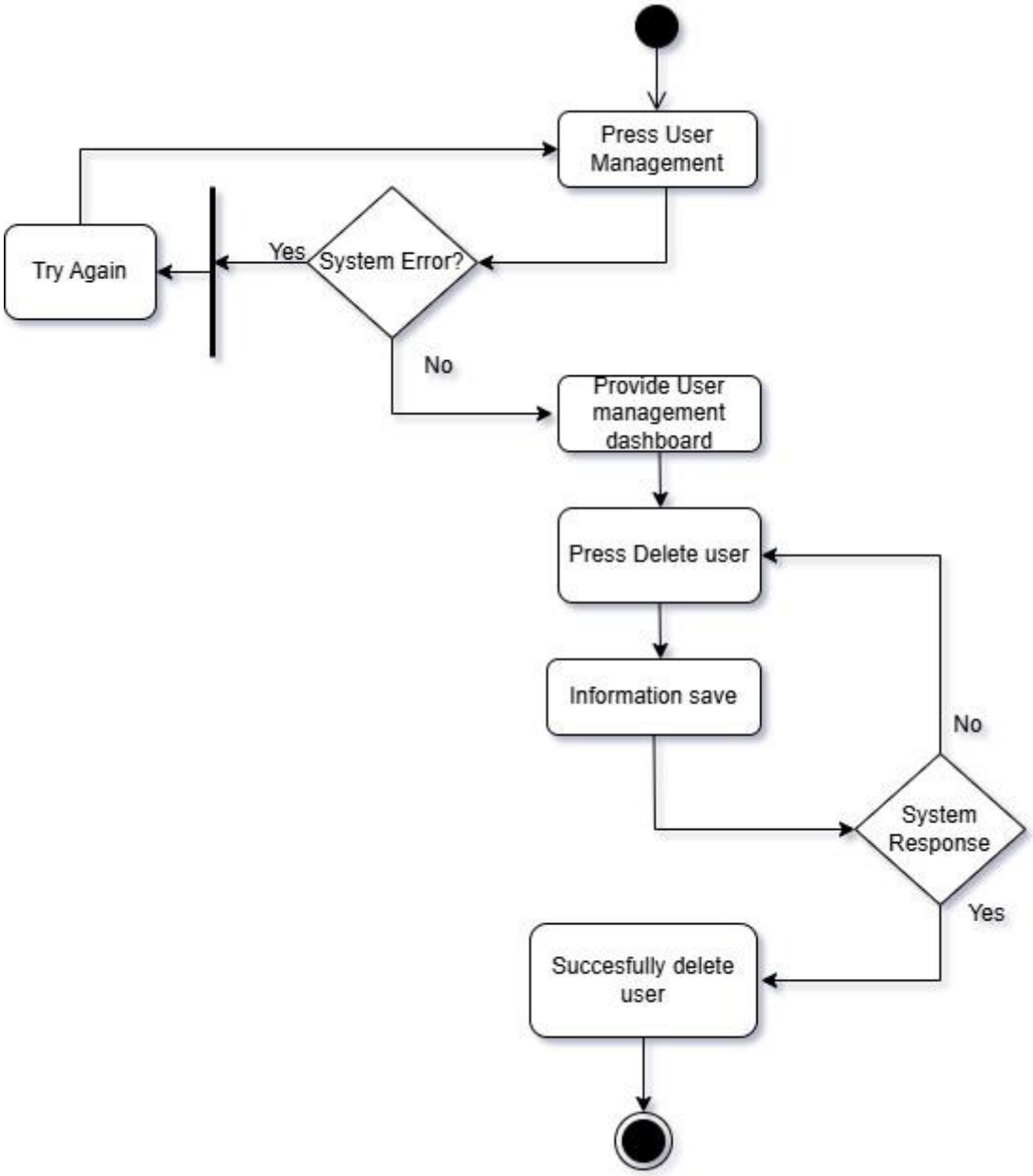


Figure 2.6 Activity diagram for User Management

For Response submission

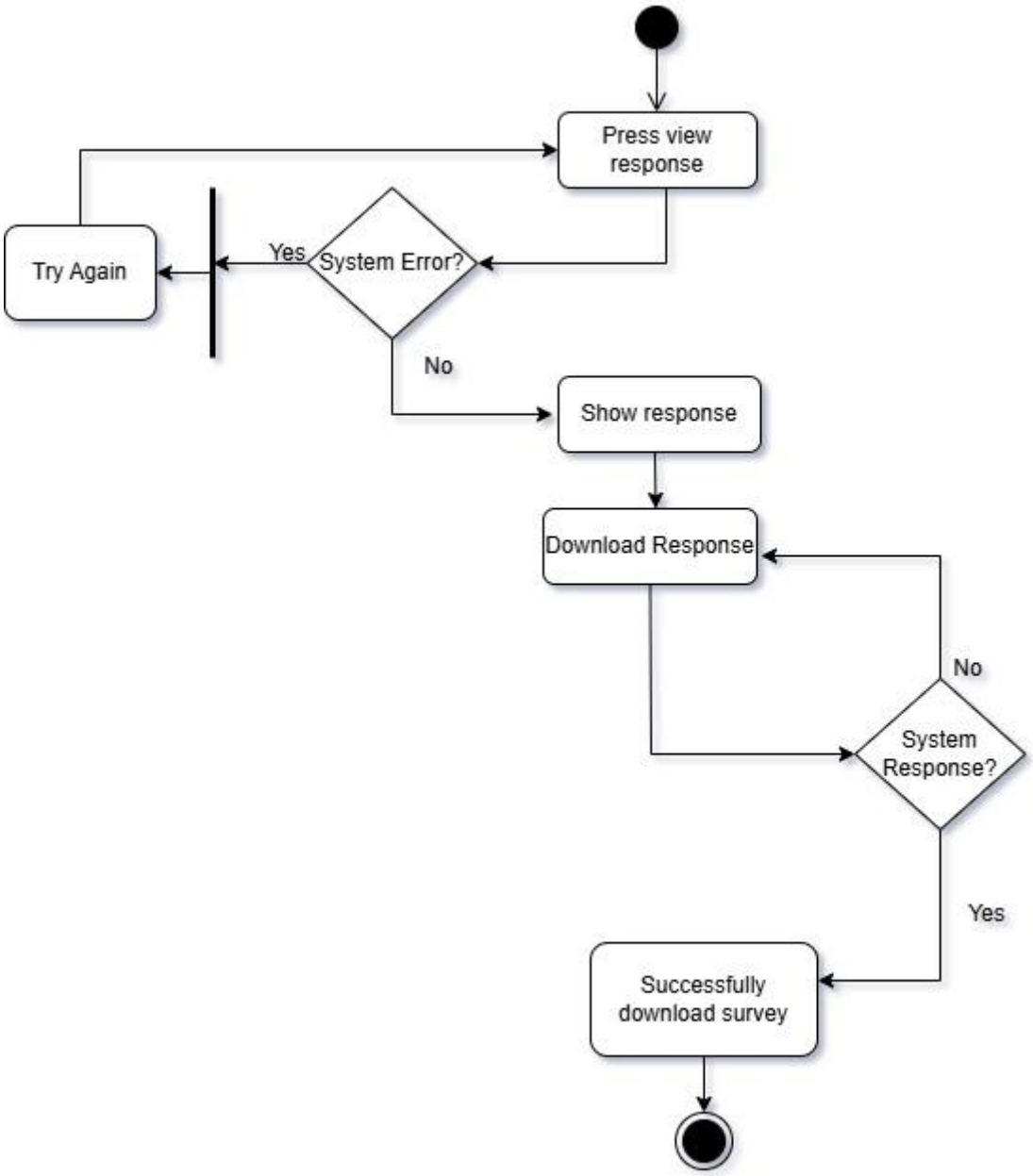


Figure 2.7 Activity diagram for Response Submission

For withdrawal request handling

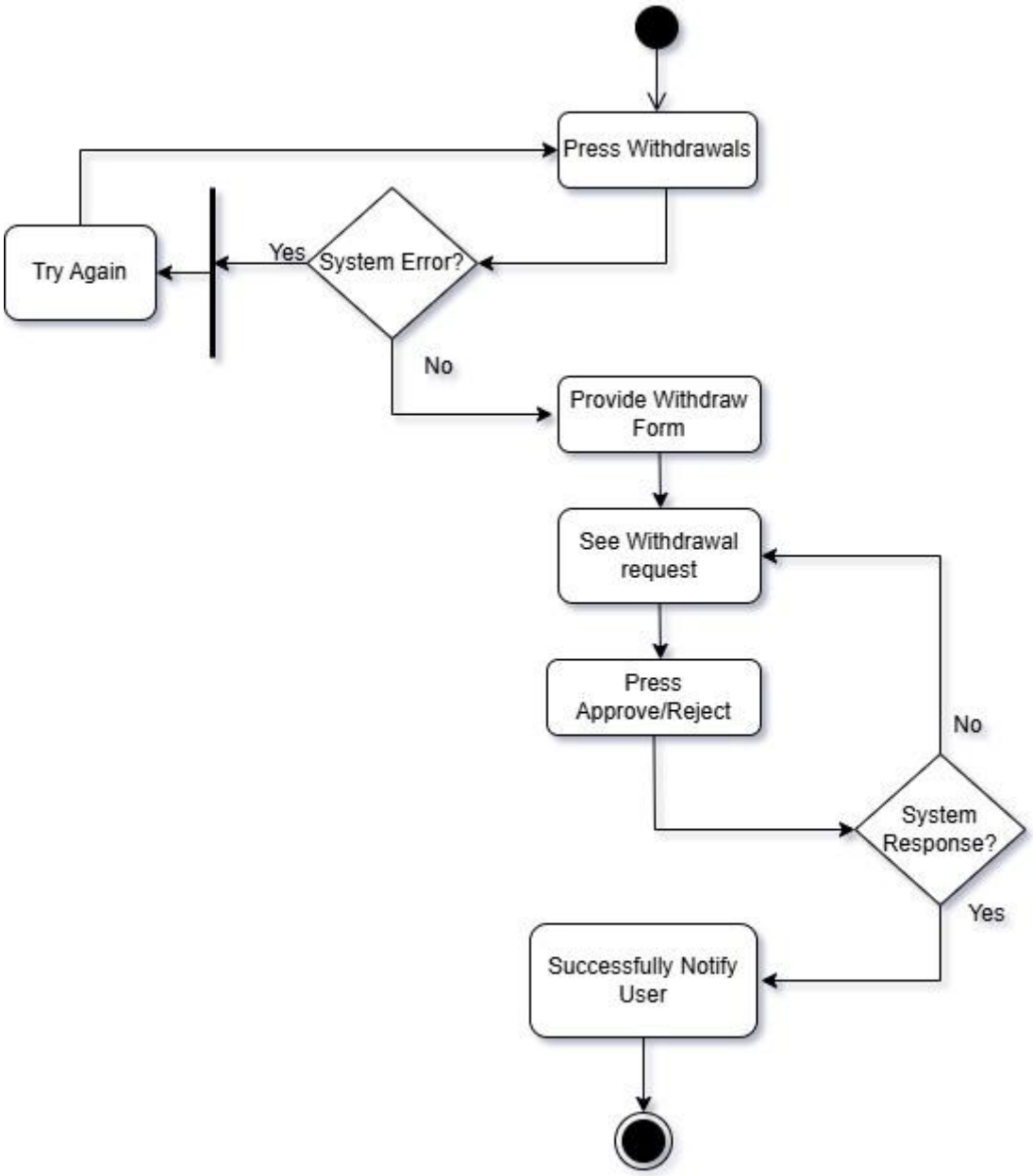


Figure 2.8 Activity diagram for Withdrawal Request Handling

For Withdraw Money

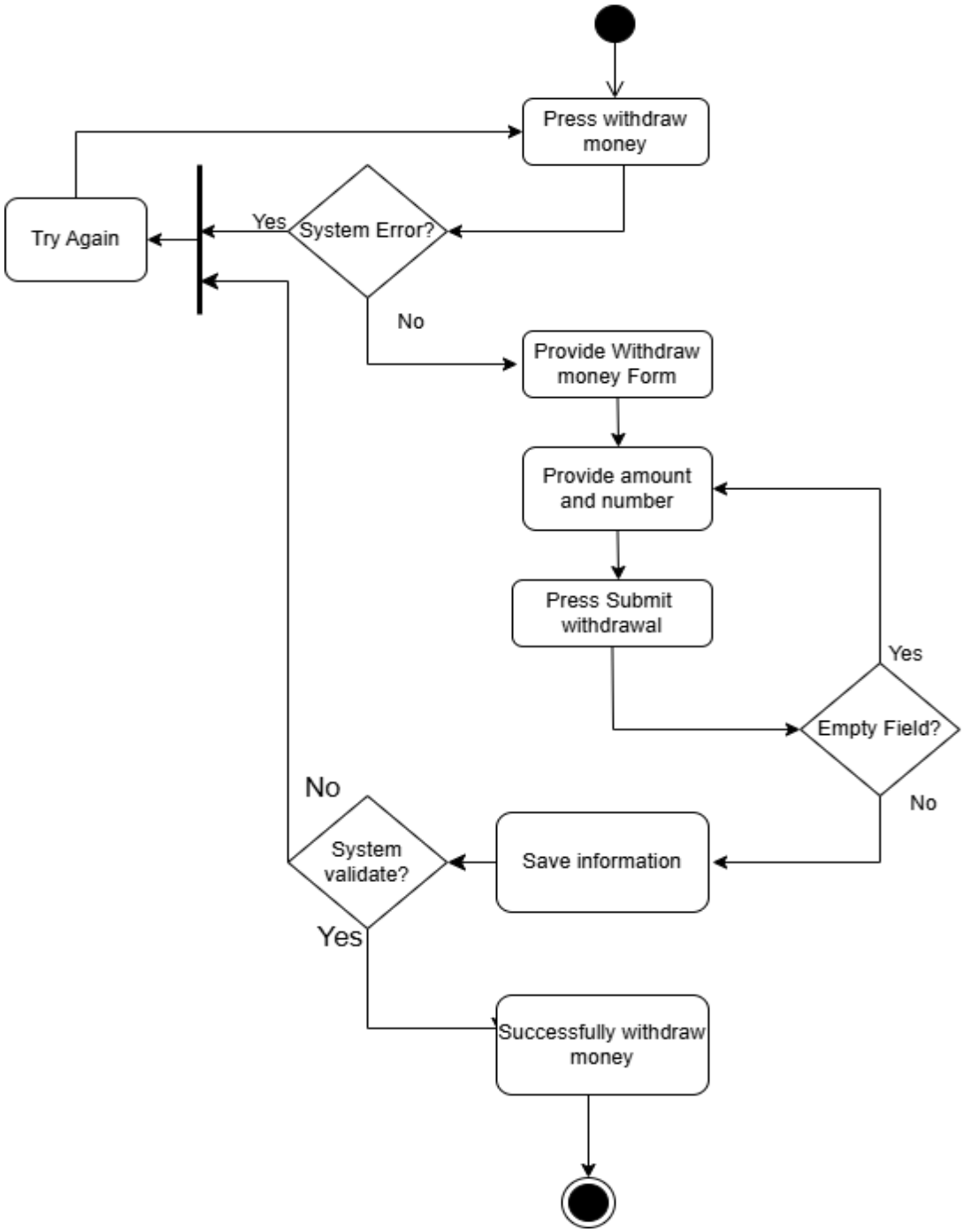


Figure 2.9 Activity diagram for Withdraw Money

For Support Submission

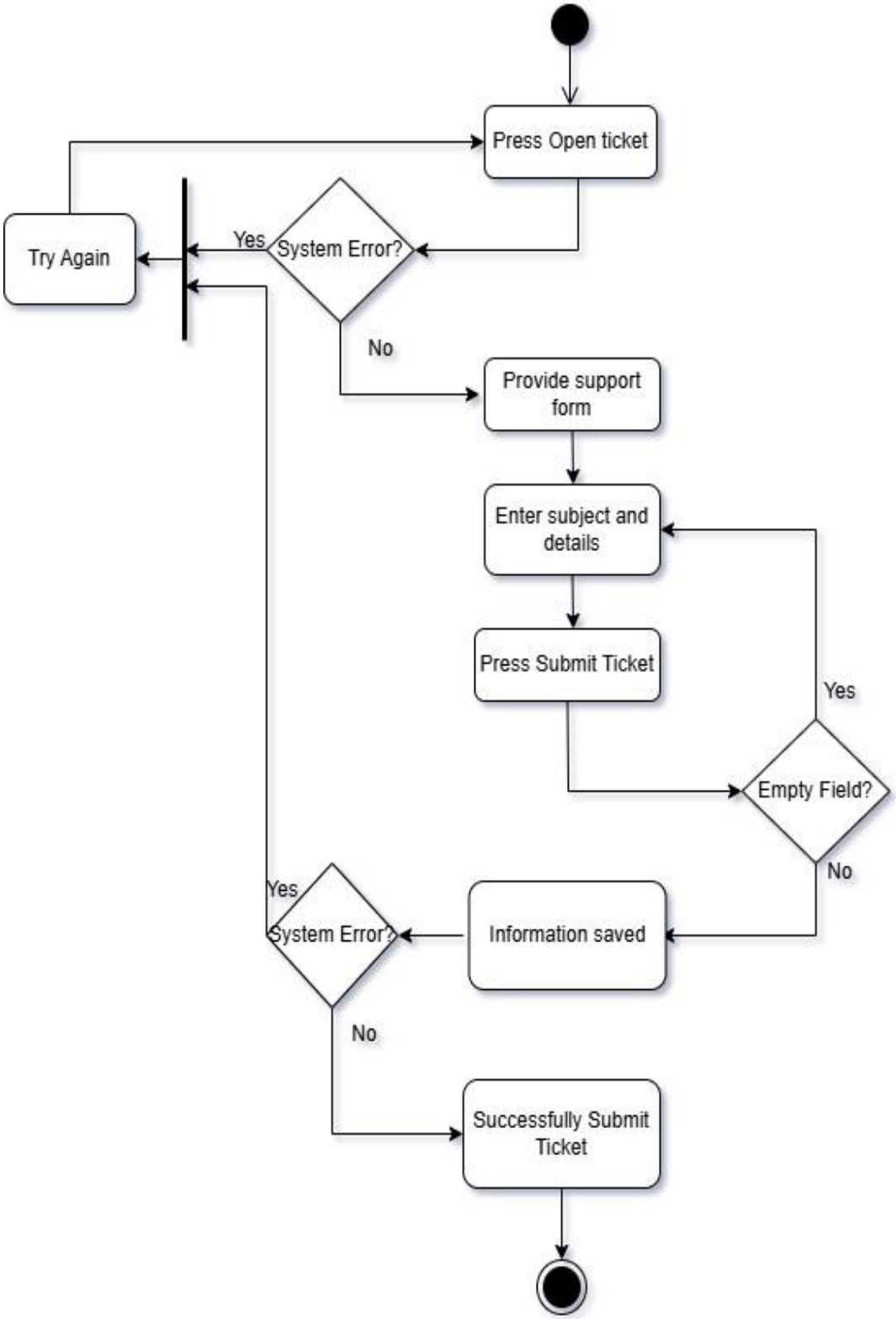


Figure 2.10 Activity diagram for Support submission

For Ticket History

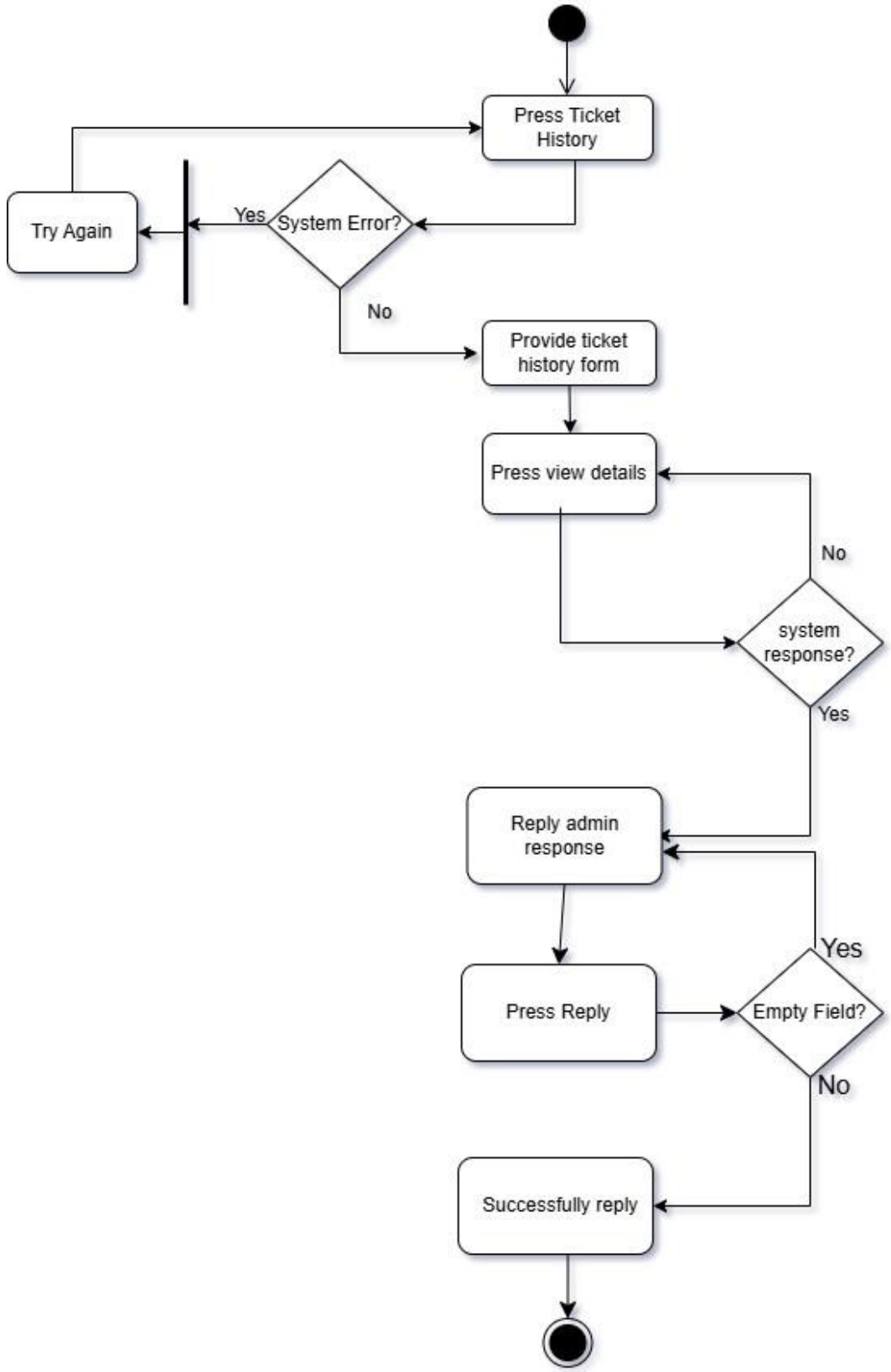


Figure 2.11 Activity diagram for Ticket History

For Support management

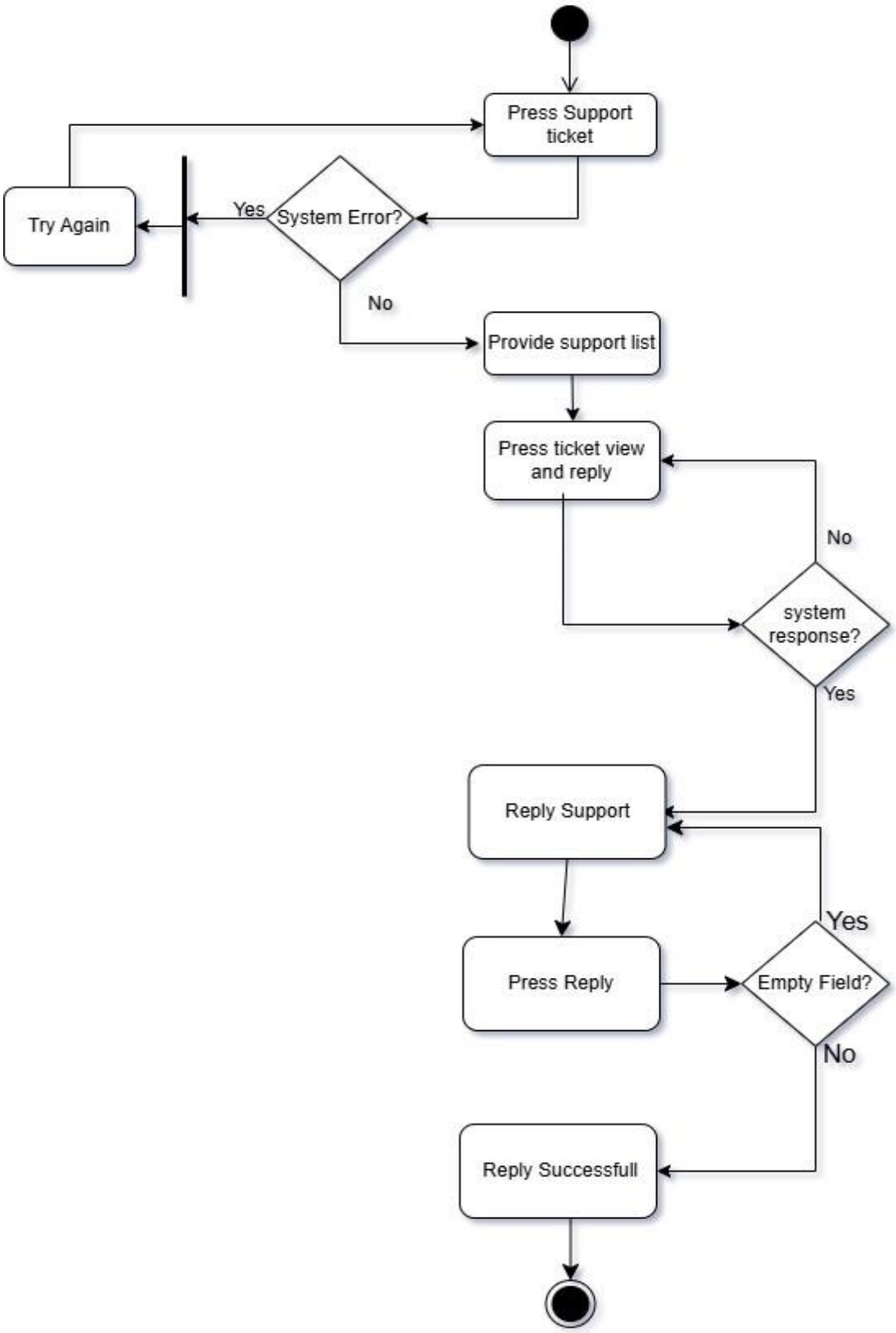


Figure 2.12 Activity diagram for Support management

For User Profile Management

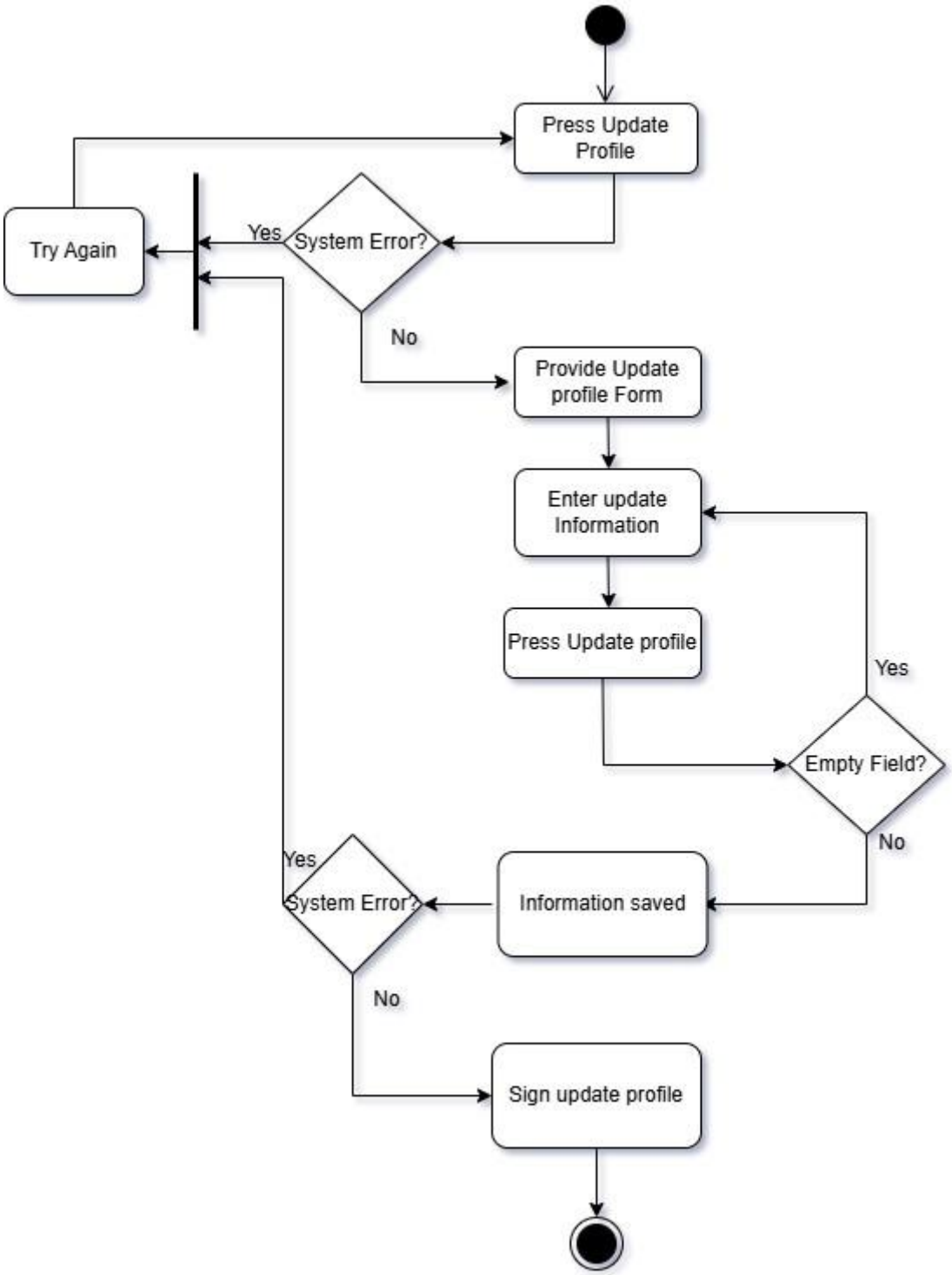


Figure 2.13 Activity diagram for user profile management

For Change password

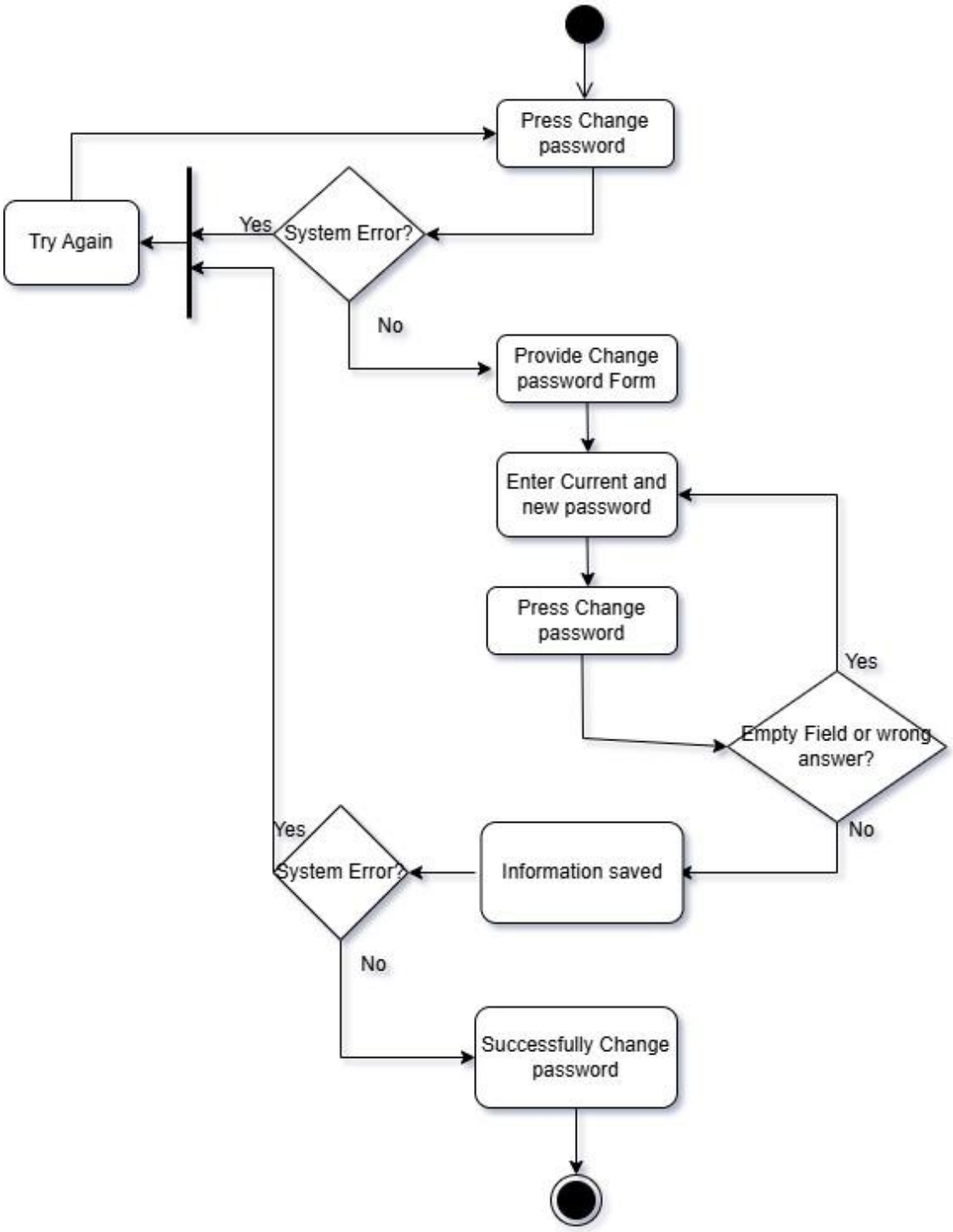


Figure 2.14 Activity diagram for Change password

For Add Admin

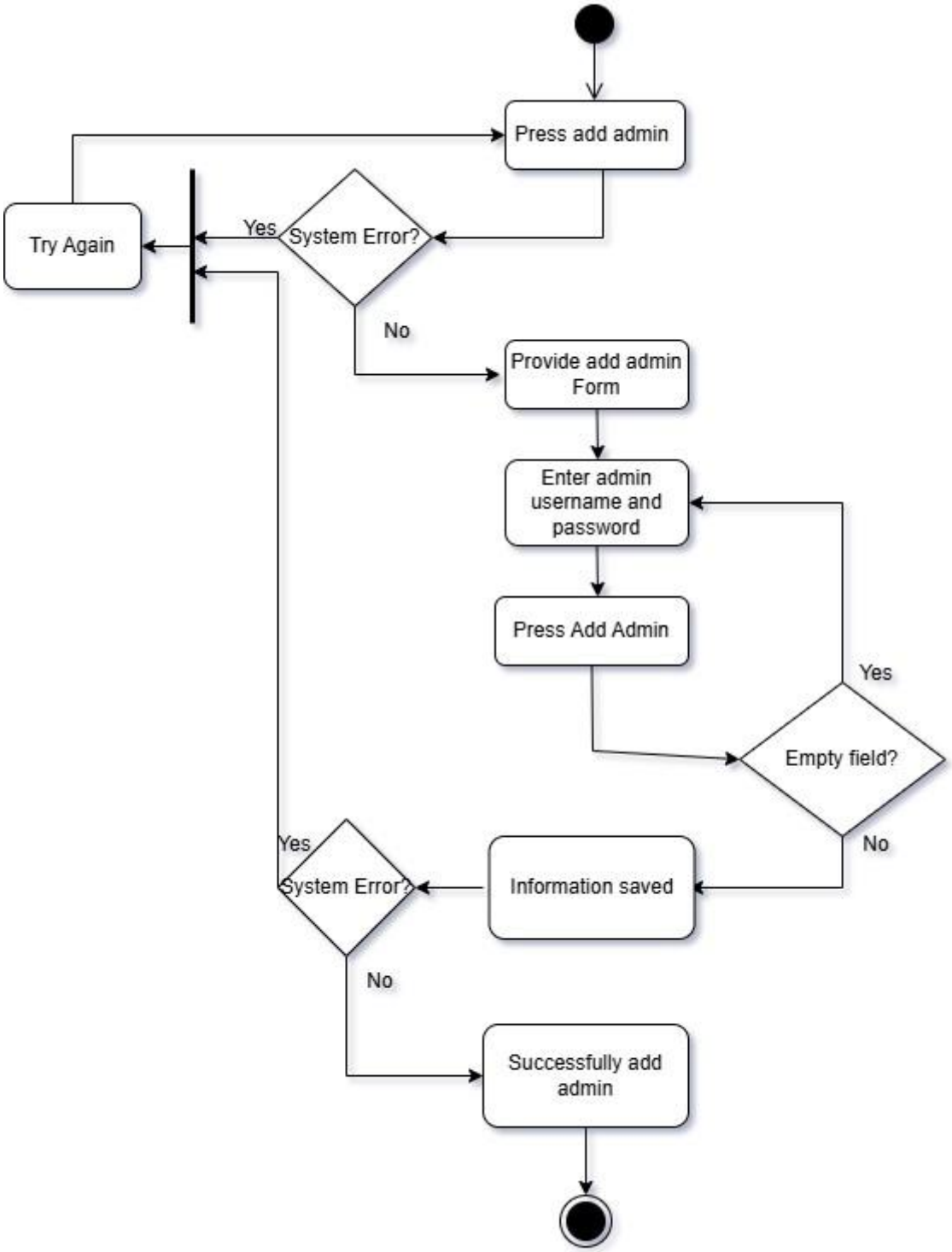


Figure 2.15 Activity diagram for Admin account setup

For Log Out

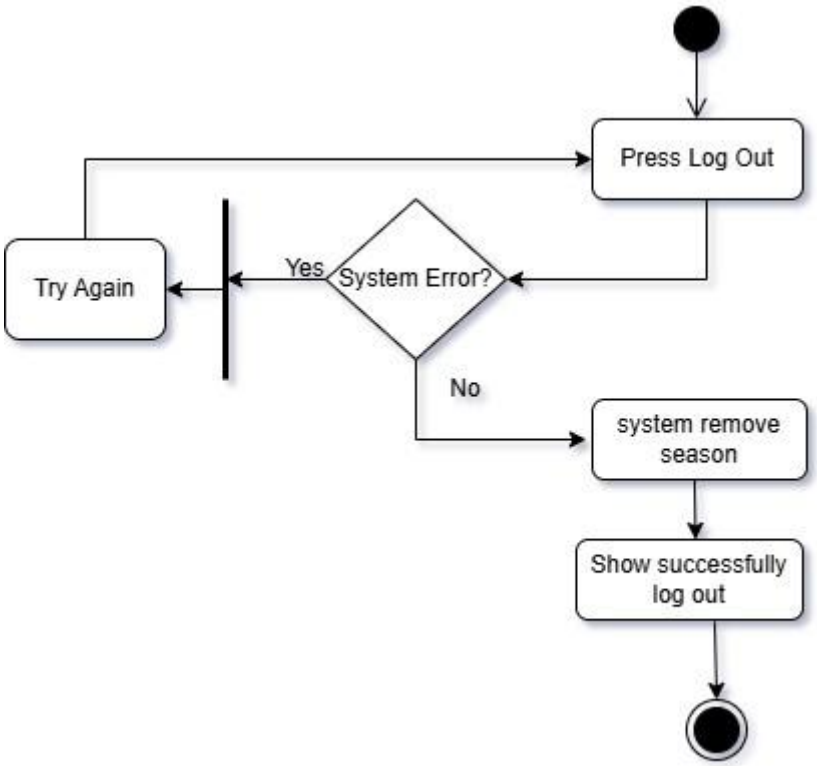


Figure 2.16 Activity diagram for log out

2.4.4 Sequence Diagram

For Sign Up

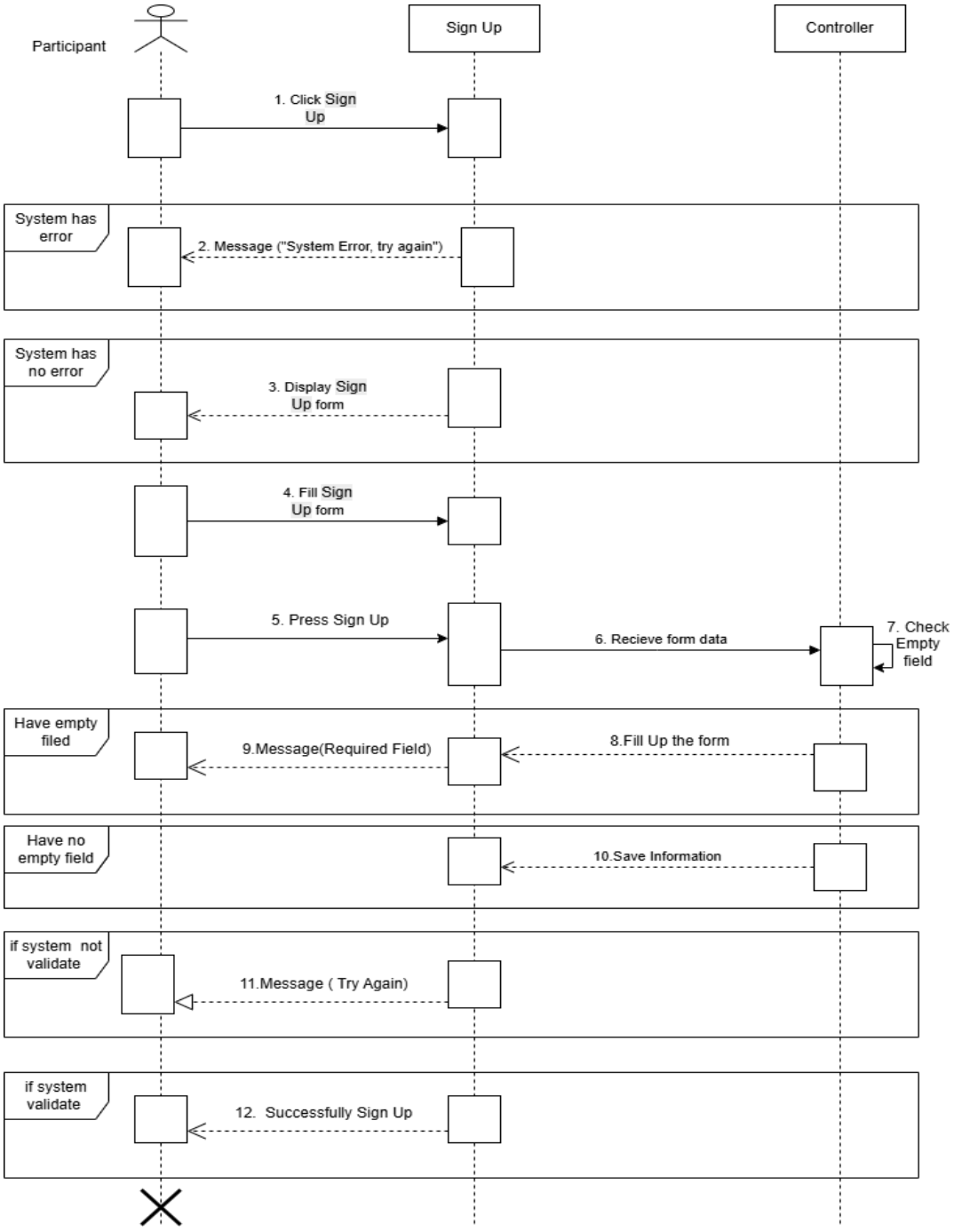


Figure 3.1:Sequence Diagram for Sign Up

For Log In

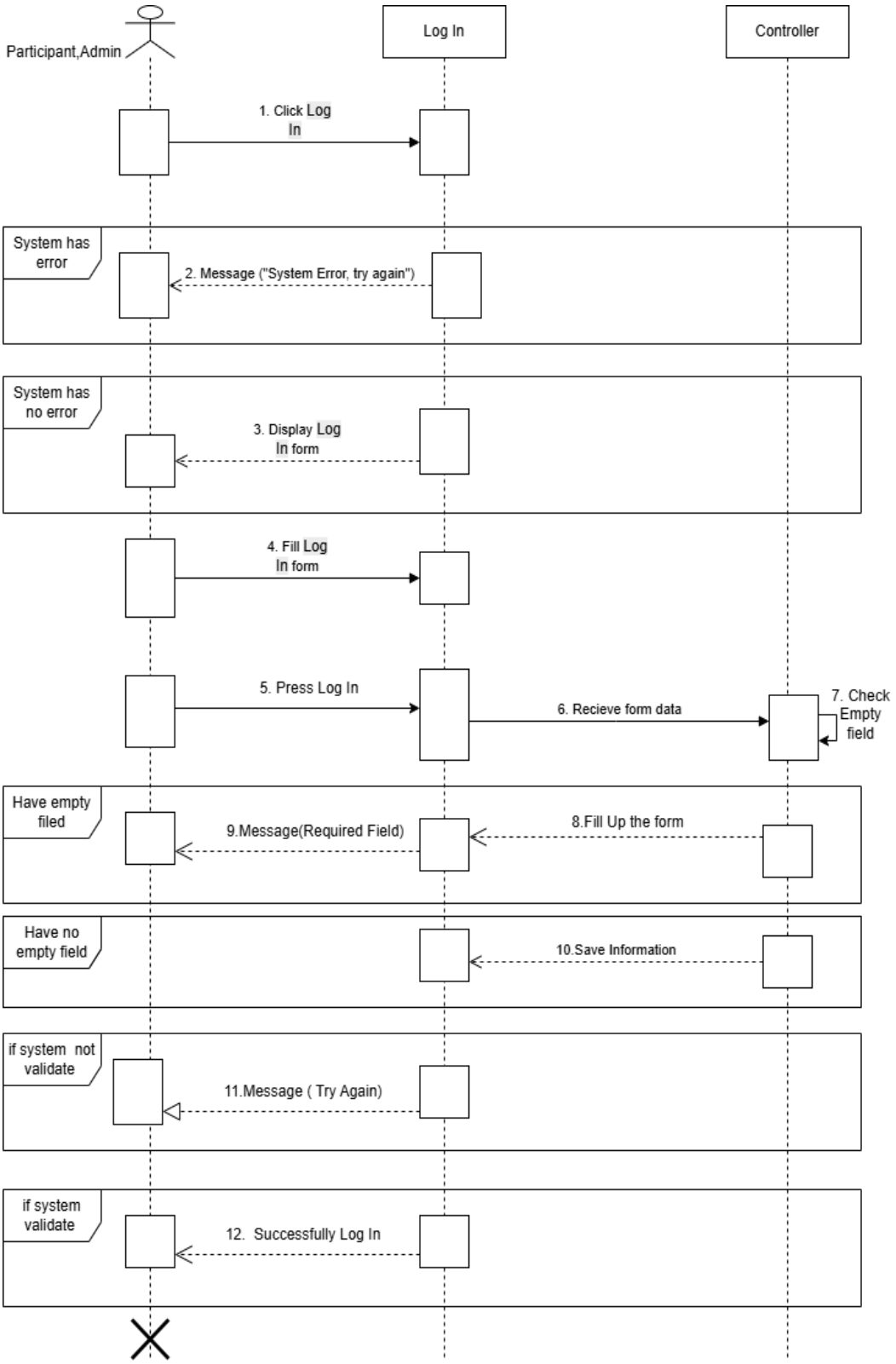


Figure 3.2: Sequence Diagram for log in

For Survey Creation

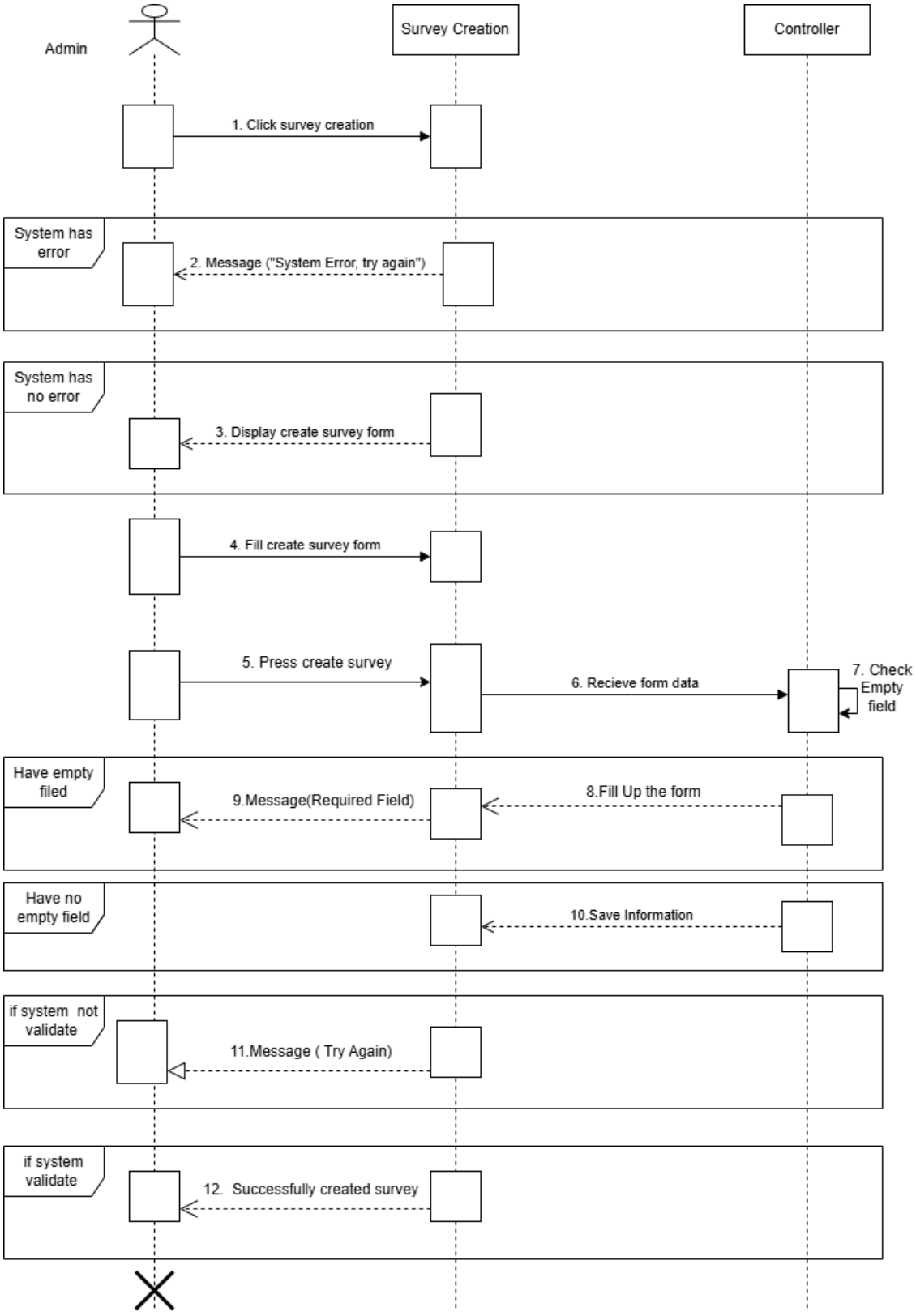


Figure 3.3: Sequence Diagram for Survey Creation

For Survey Participation

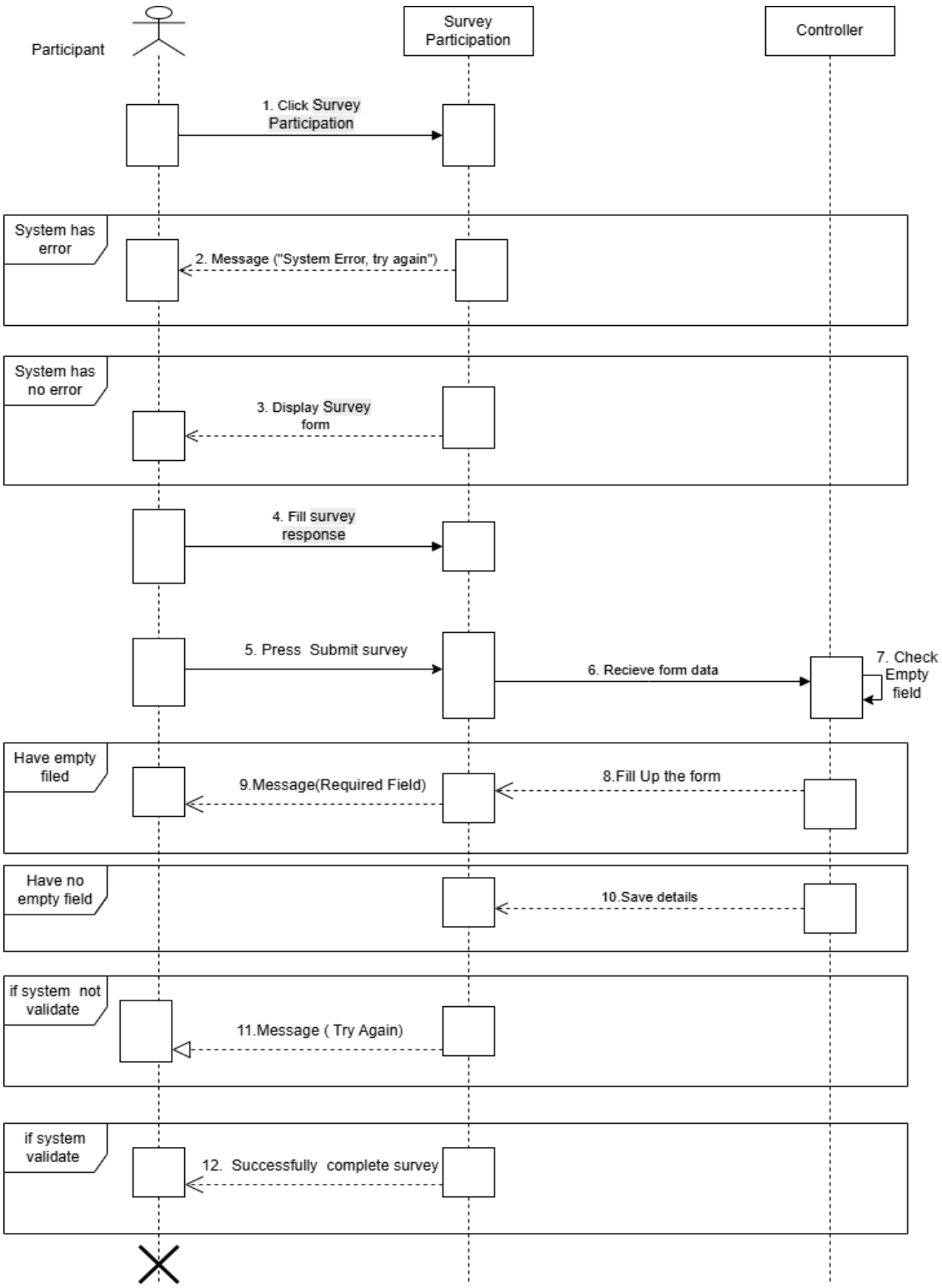


Figure 3.4: Sequence Diagram for Survey Participation

For Survey Editing

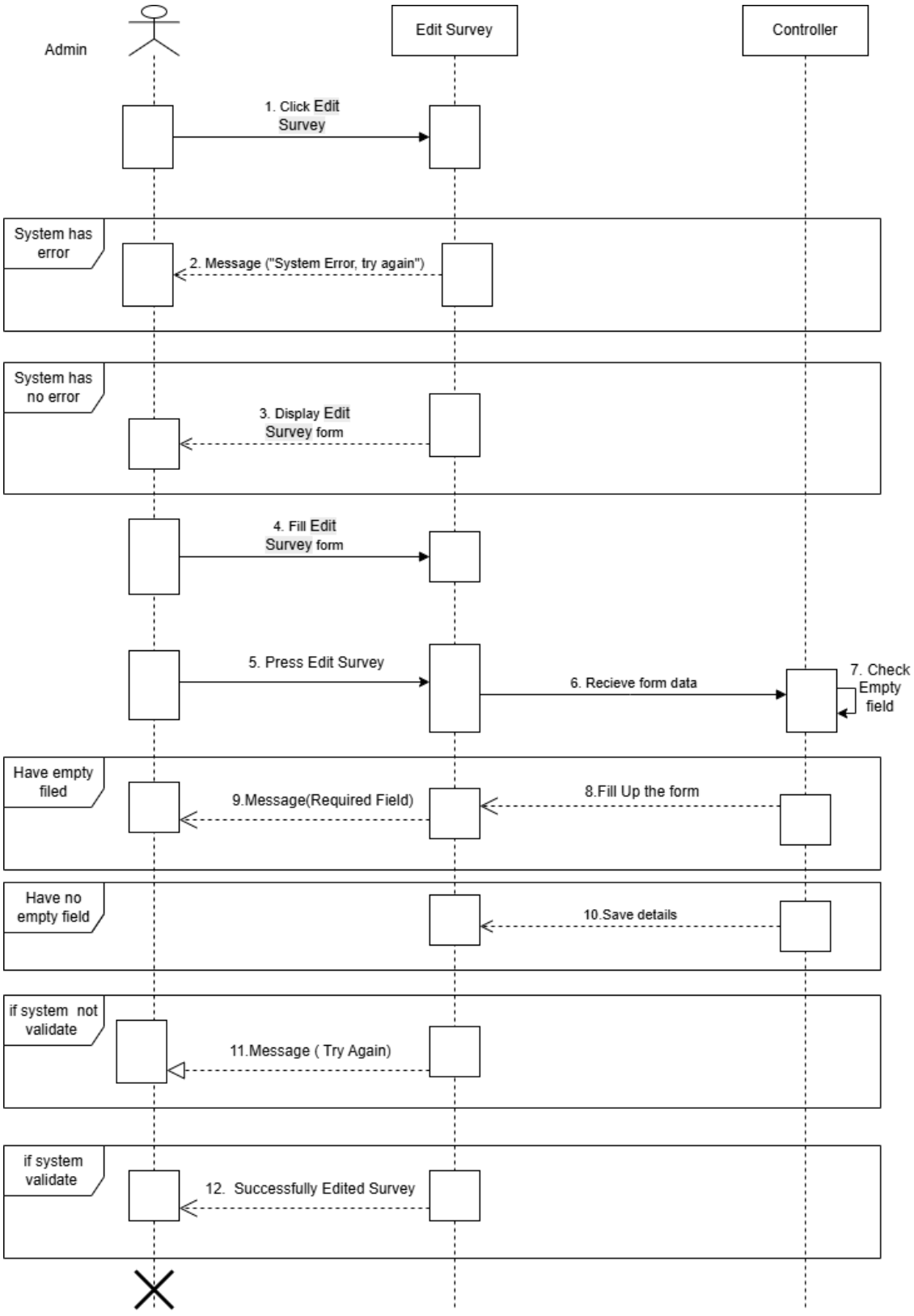
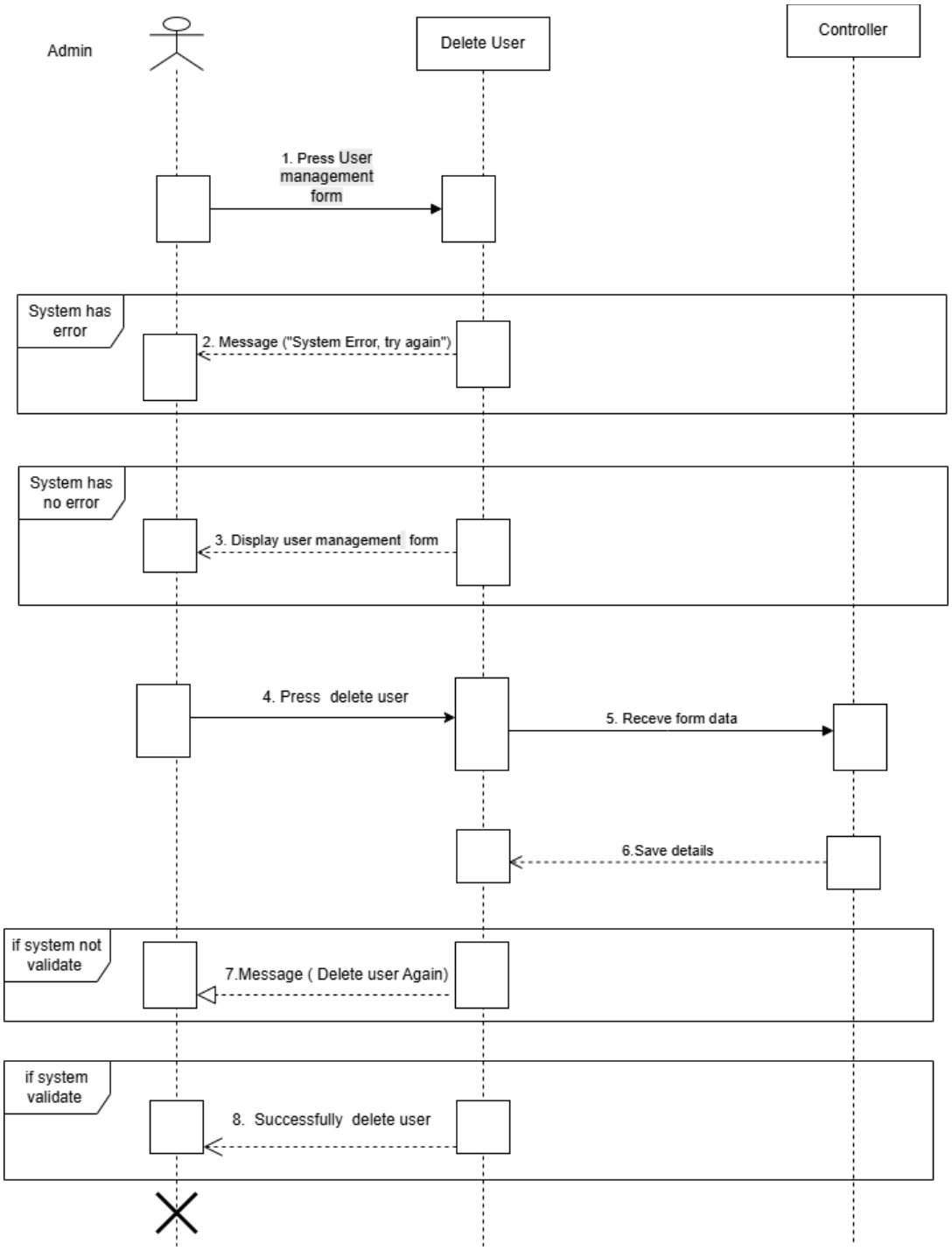


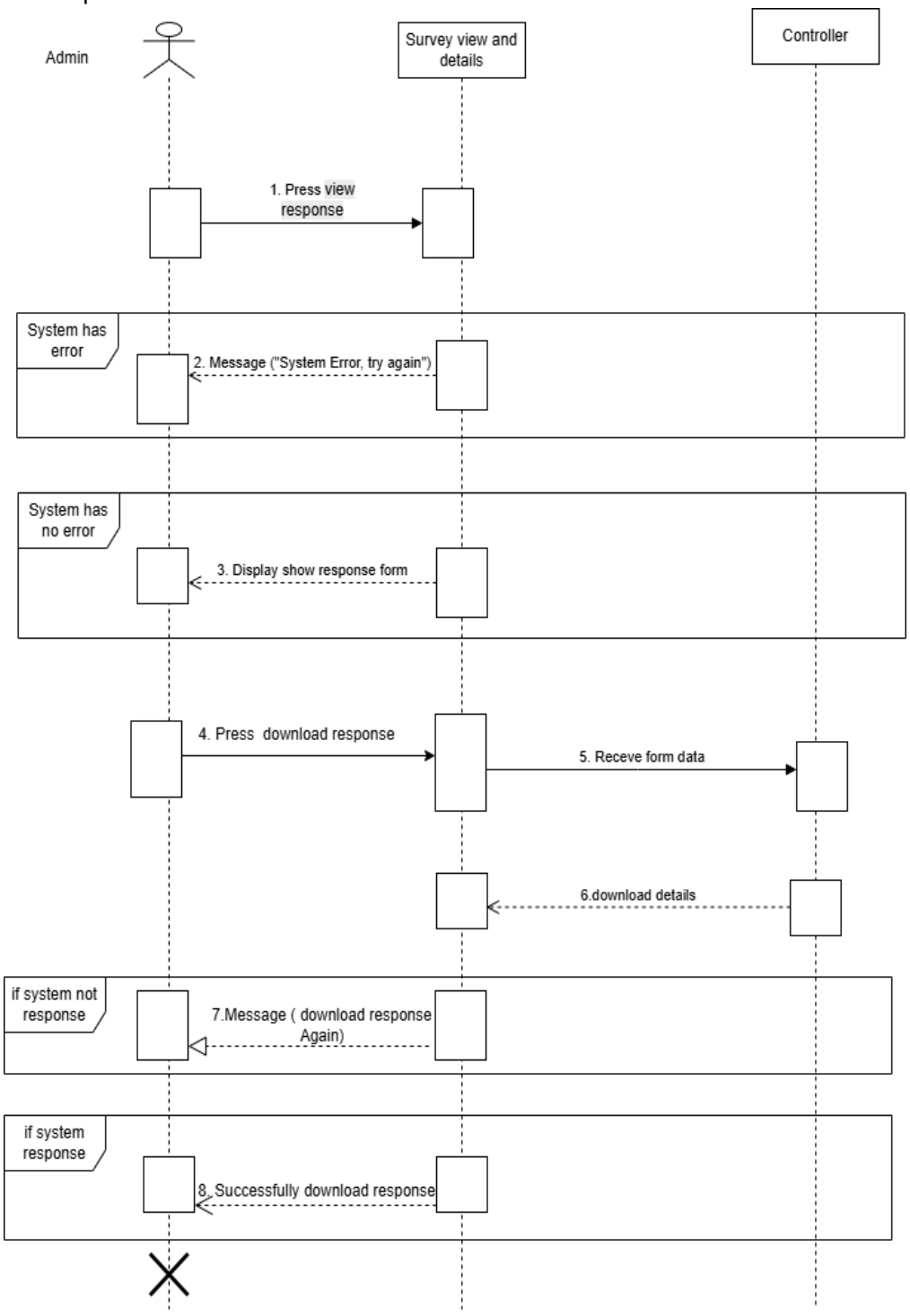
Figure 3.5: Sequence Diagram for Survey editing

For User Management



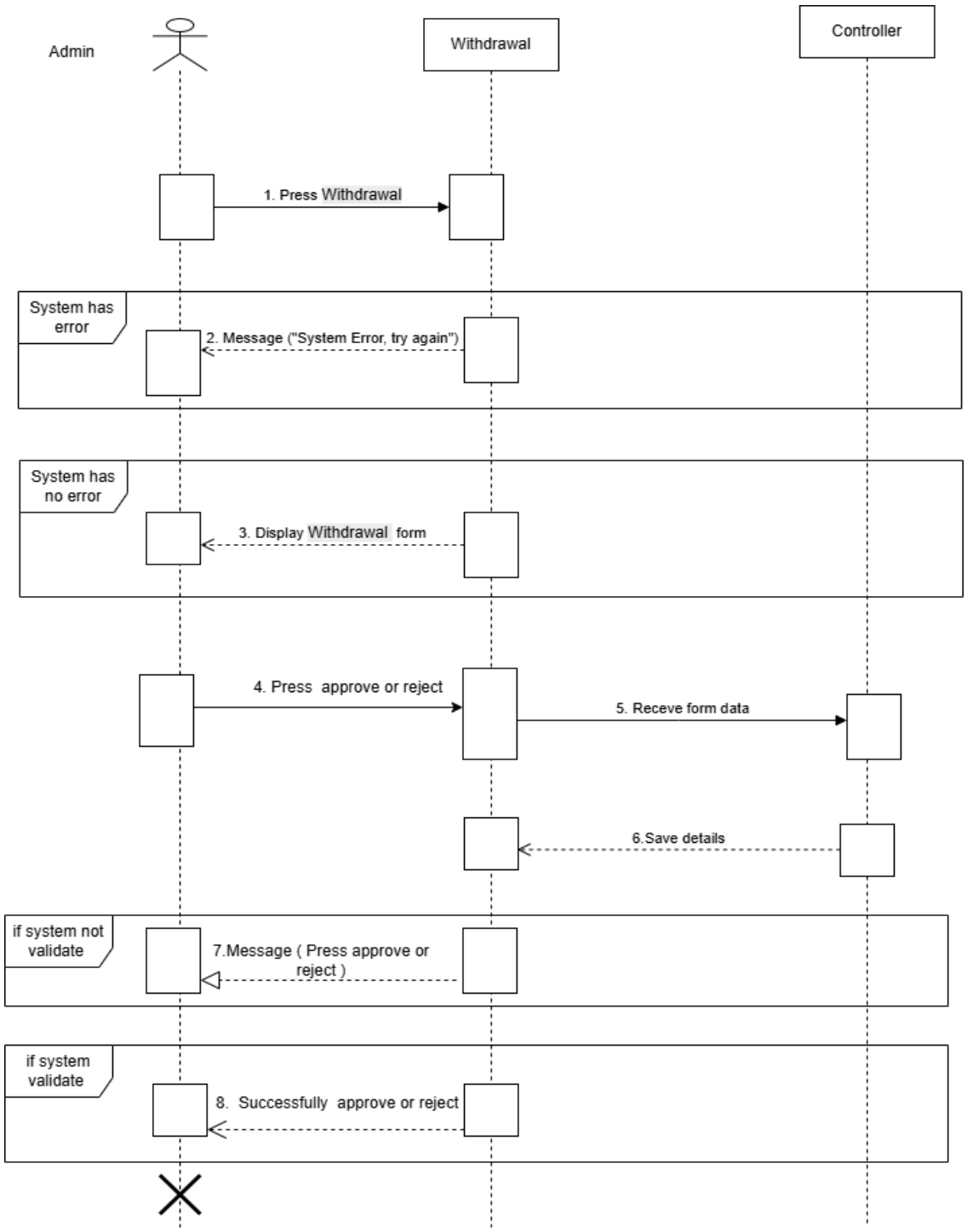
3.6: Sequence Diagram for User Management

For Response Submission



3.7: Sequence Diagram for response submission

For Withdrawals Request Handling



3.8: Sequence Diagram for withdrawal request handling

For Withdraw Money

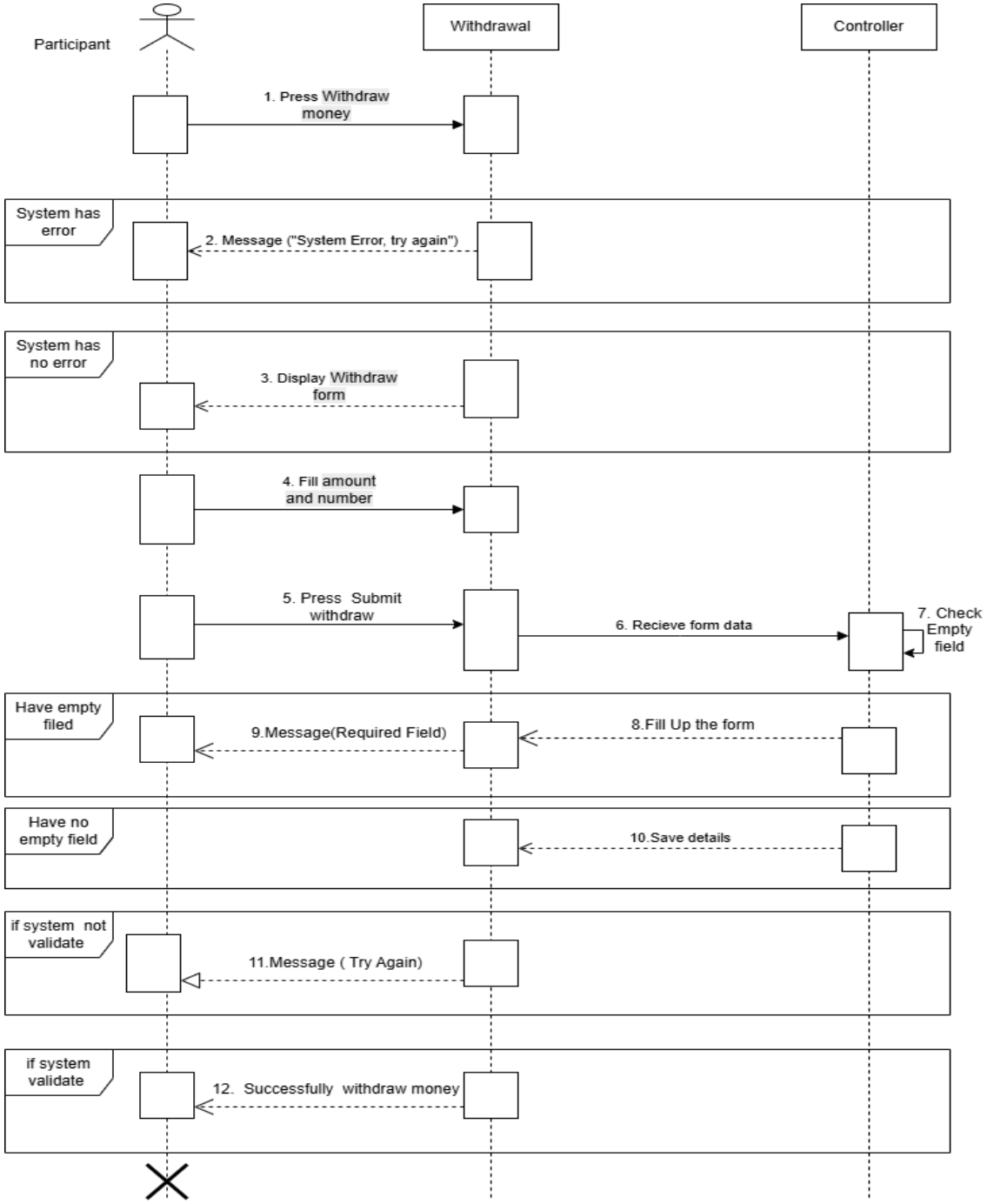


Figure 3.9: Sequence Diagram for withdraw money

For Support Submission

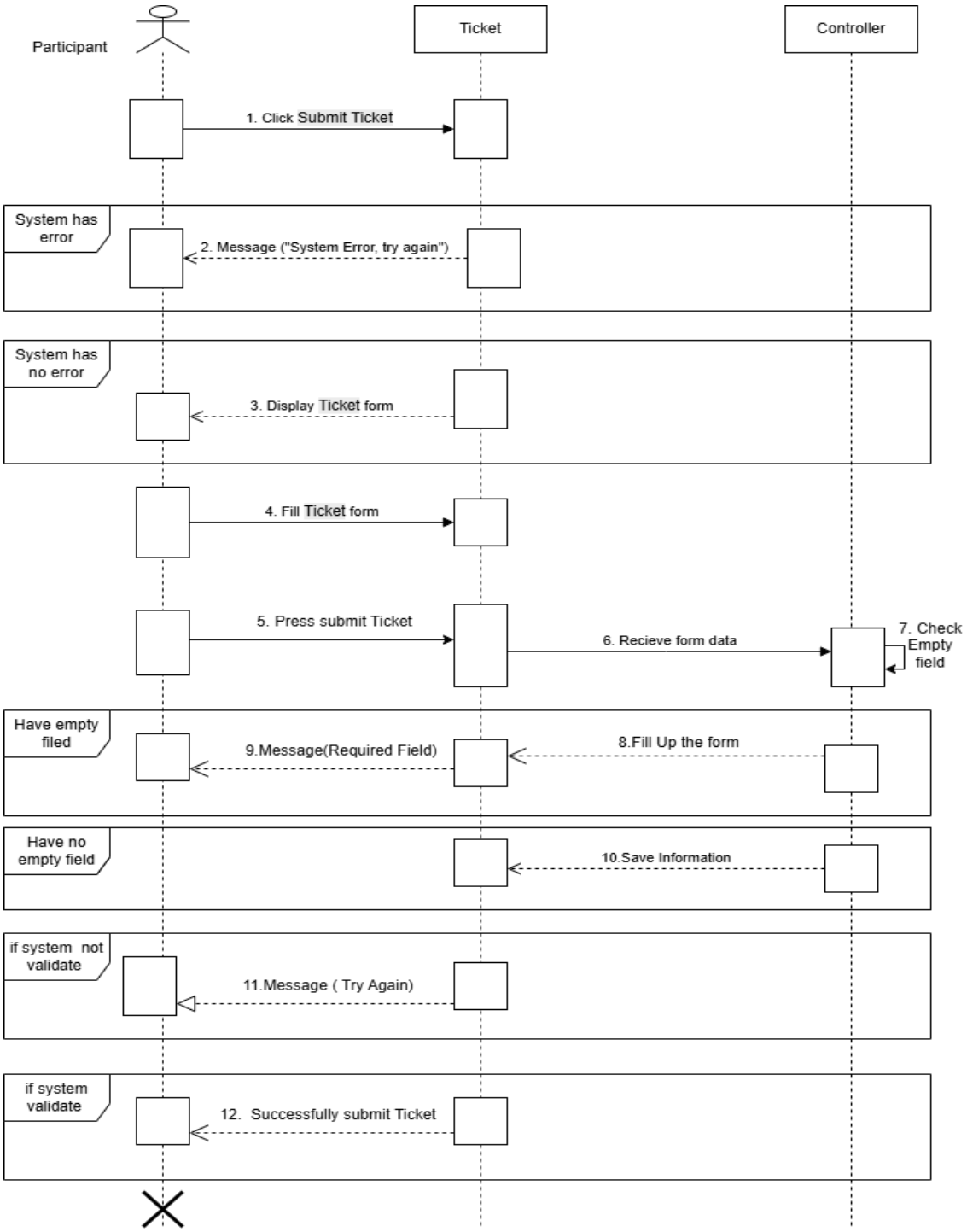


Figure 3.10: Sequence Diagram for Support Submission

For ticket history

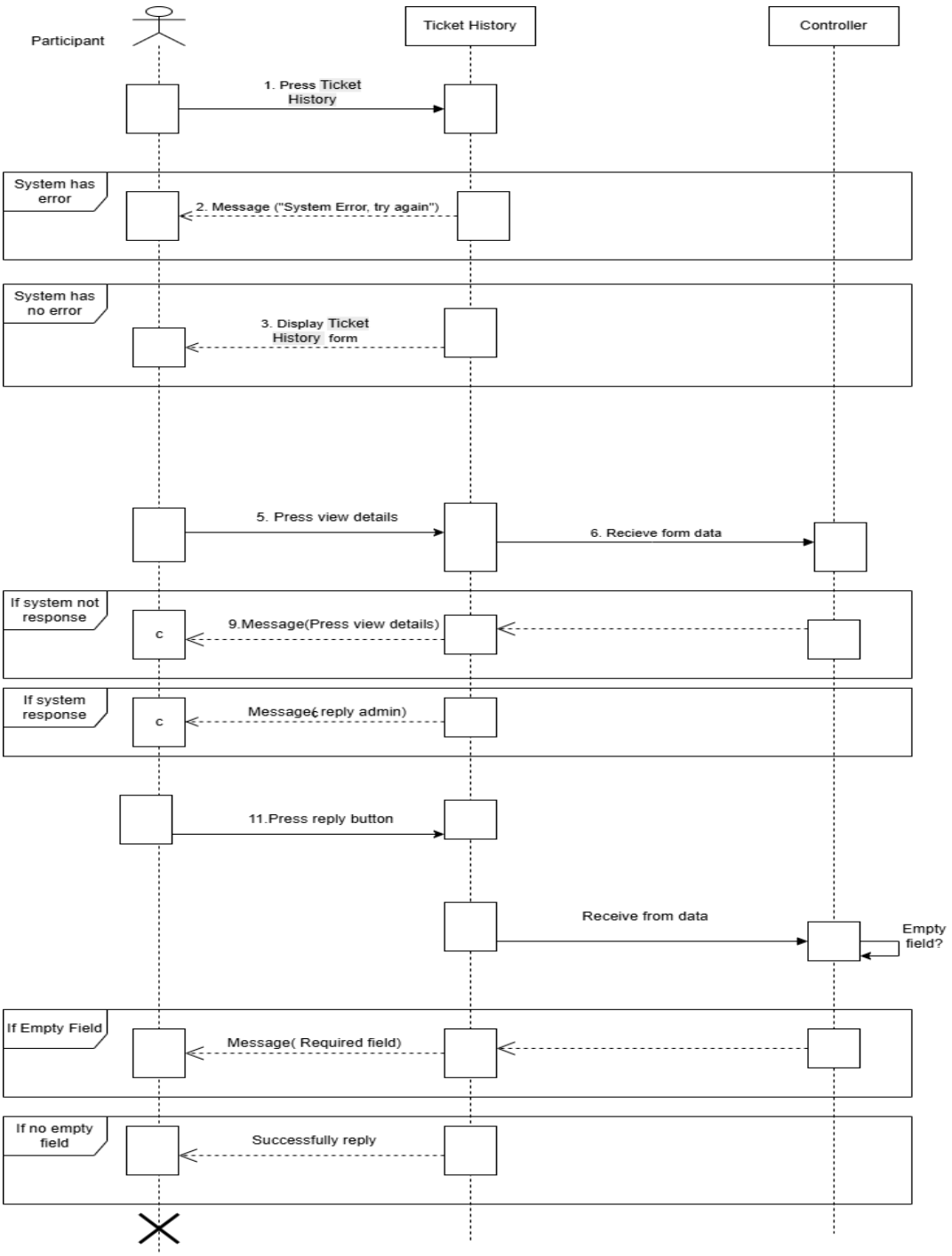


Figure 3.11: Sequence Diagram for ticket history

For Support Management

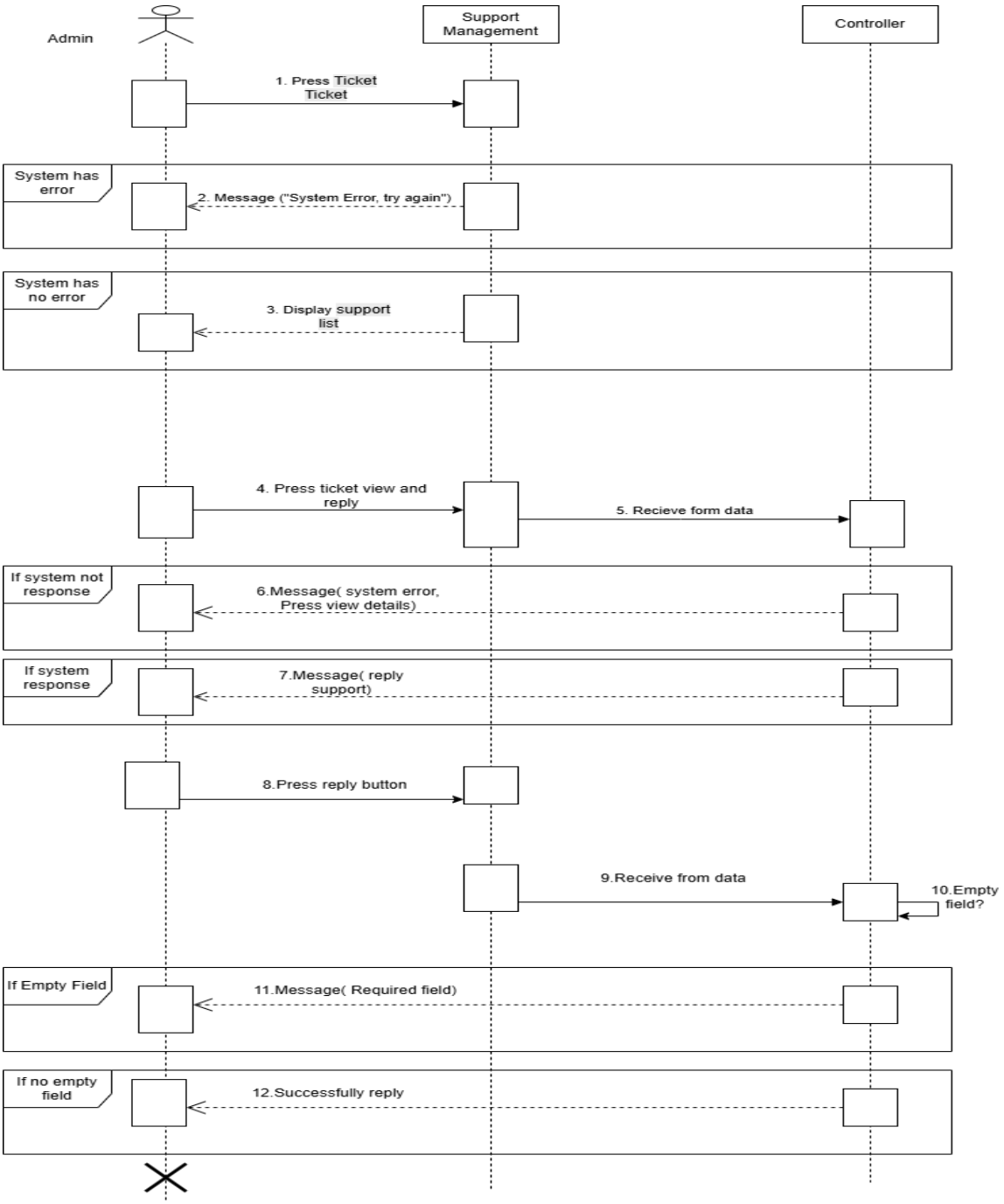


Figure 3.12: Sequence Diagram for Support management

For User Profile Management

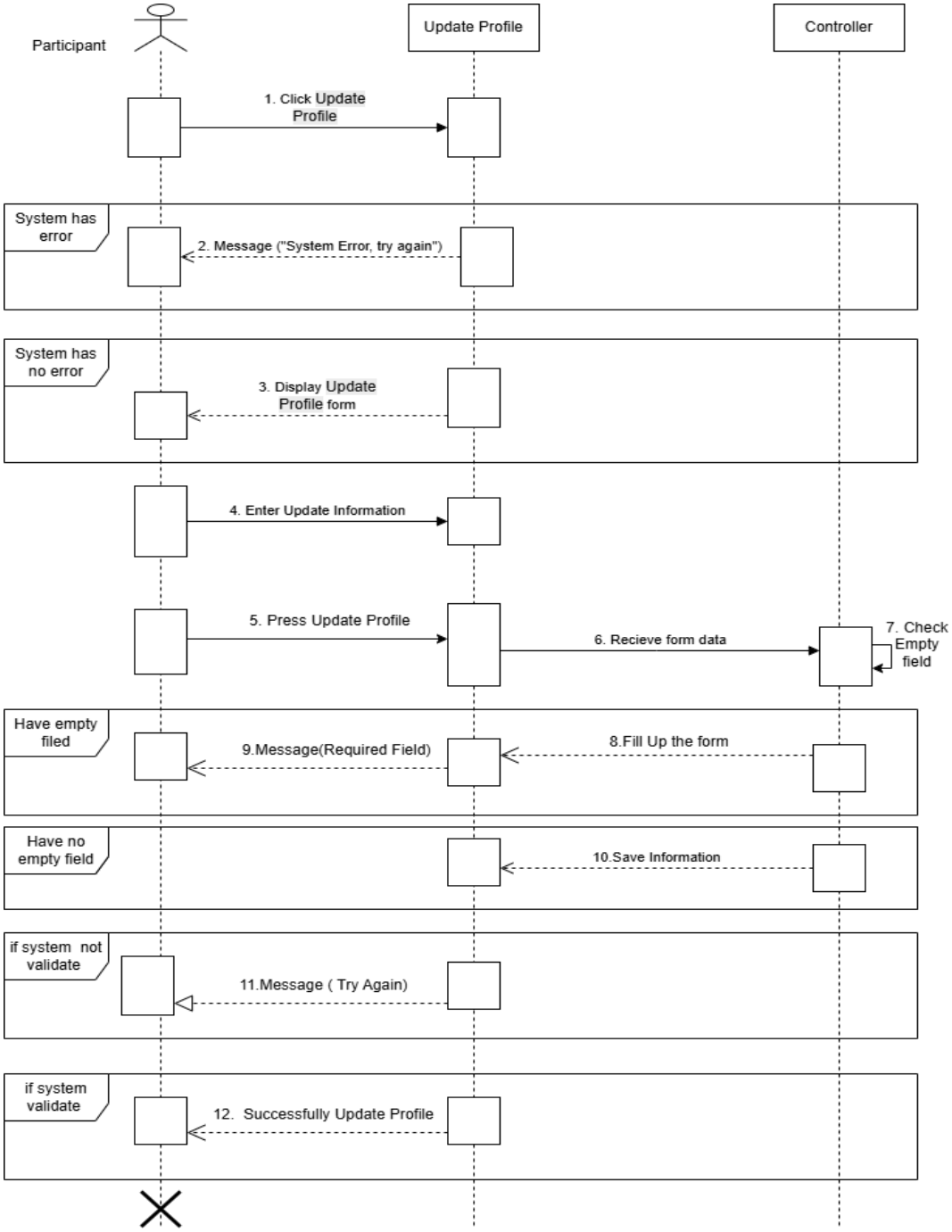


Figure 3.13: Sequence Diagram for User profile management

For Change Password

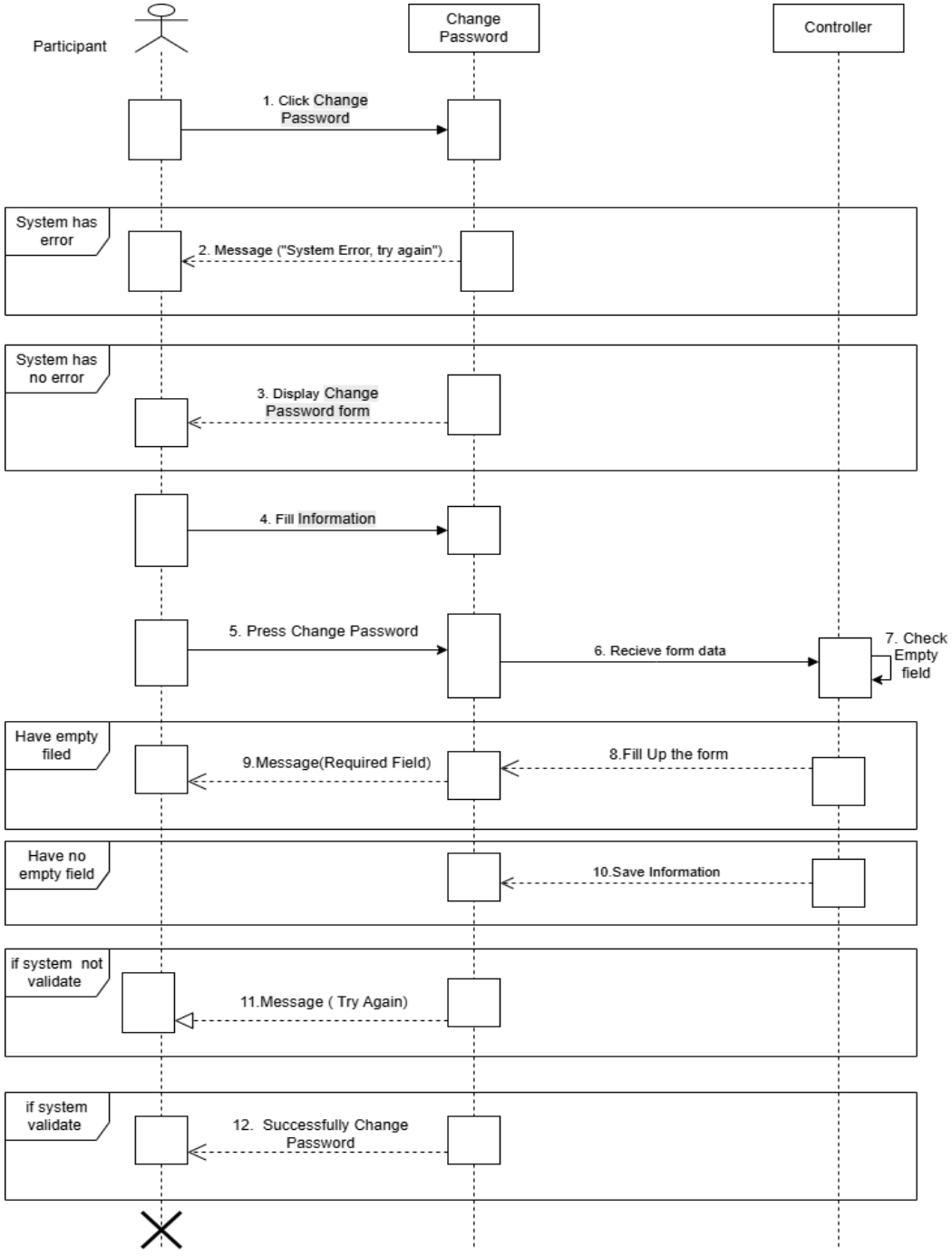


Figure 3.14: Sequence Diagram for change password

For Add Admin

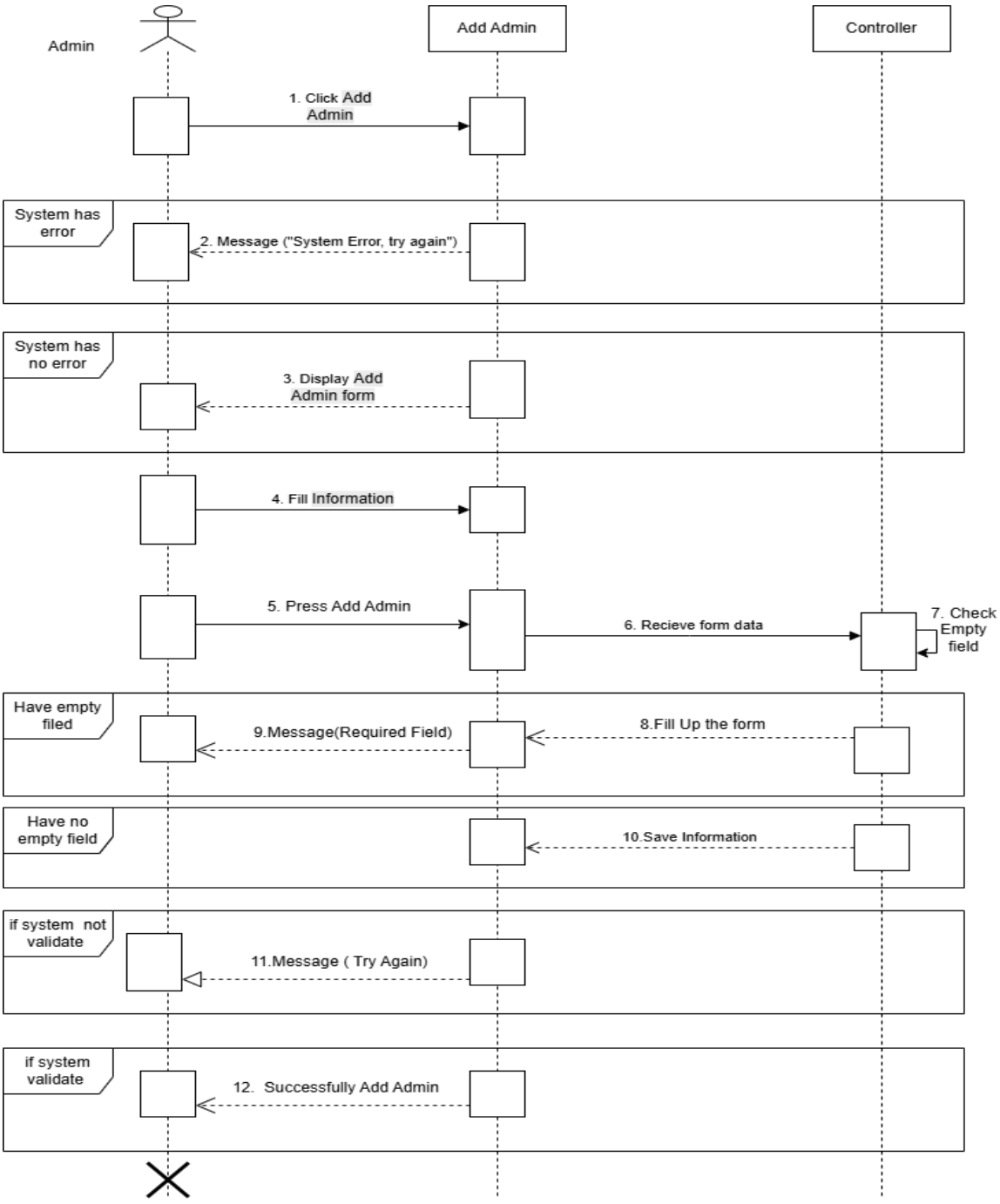


Figure 3.15: Sequence Diagram for add admin

For Log Out

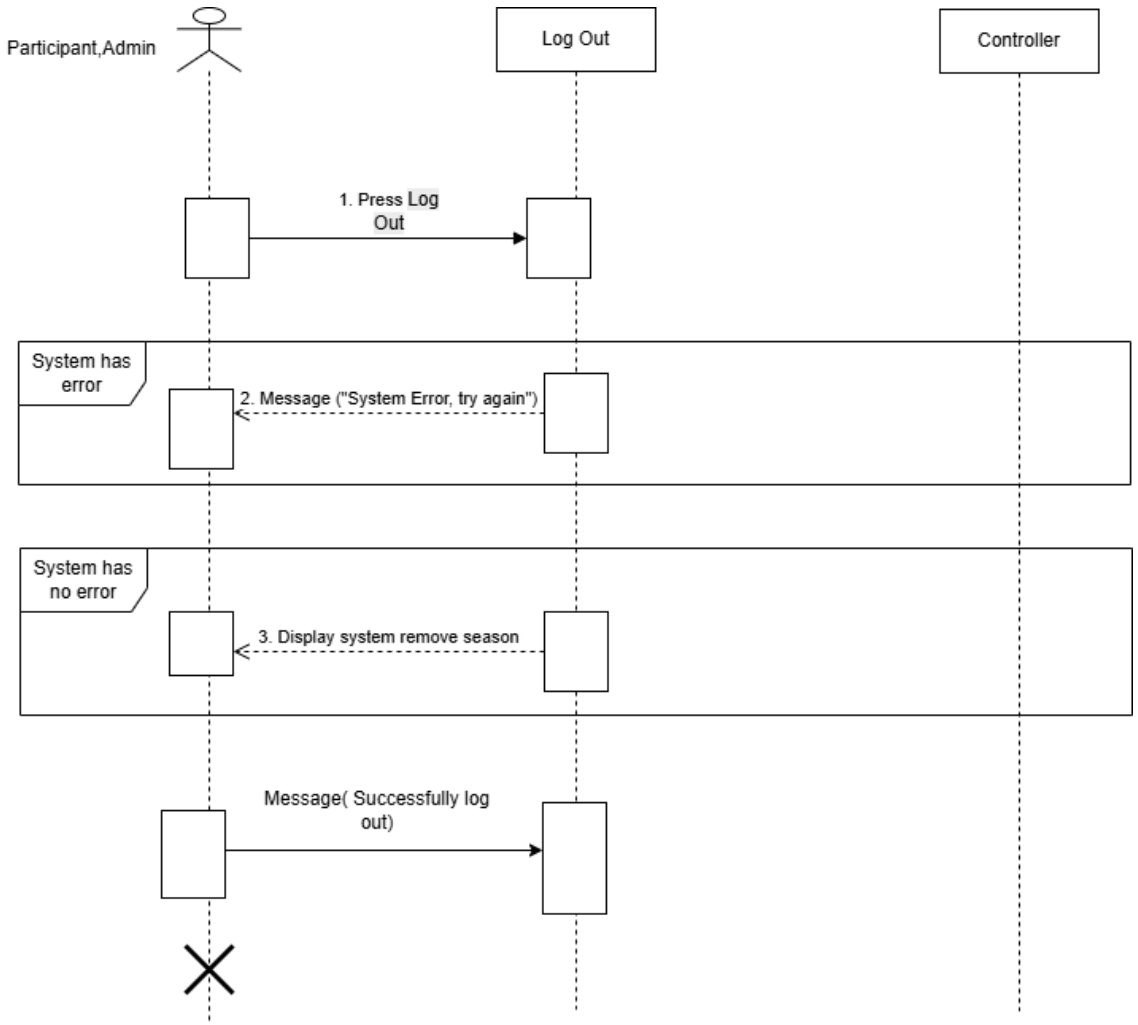


Figure 3.16: Sequence Diagram for log out

2.4.5 Class Diagram

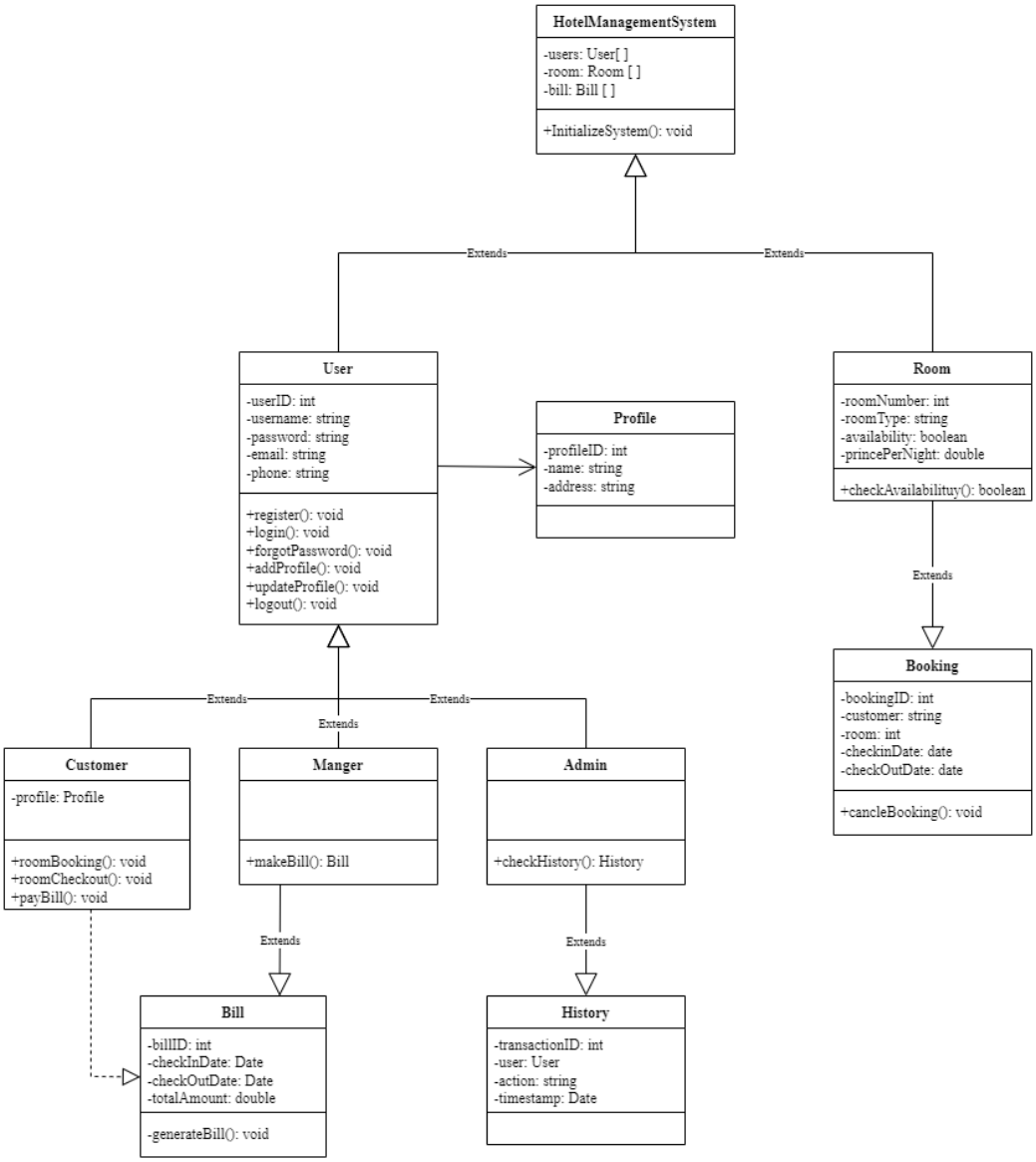


Figure 4: Class Diagram

2.4.6 ER Diagram

Figure 5: ER Diagram

2.5 Coding: Code Snip

This part contain important logic/part of the Taskbuddy Project code

For Sign Up

```php
$host = 'localhost';
$username = 'root'; // Default XAMPP username
$password = ''; // Default XAMPP password is empty
$dbname = 'task_buddy_db'; // Ensure this matches your database name

try {
    // Create connection using PDO
    $conn = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);
    // Set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    die("Connection failed: " . $e->getMessage());
}

// Handle form submission
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST['firstName'] . ' ' . $_POST['lastName']; // Combine first and last name for username
    $password = password_hash($_POST['password'], PASSWORD_DEFAULT); // Hash the password for security

    $email = $_POST['email']; // Get email from form

    // Check if passwords match
    if ($_POST['password'] !== $_POST['confirmPassword']) {
        $_SESSION['error'] = "Passwords do not match!";
        header("Location: register.php");
        exit();
    }

    // Insert user into the database using prepared statements
    $sql = "INSERT INTO users (username, email, password) VALUES (:username, :email, :password)";
    $stmt = $conn->prepare($sql);
    $stmt->bindParam(':username', $username);
```

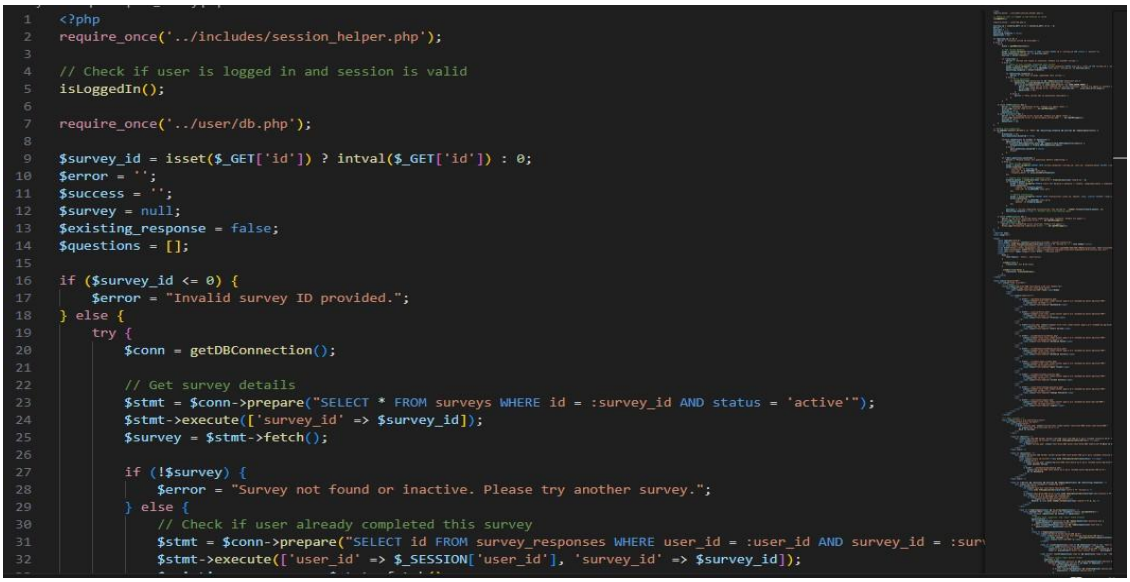
For Log In

```php
<?php
session_start();

// Handle logout
if (isset($_GET['logout'])) {
    session_destroy();
    header("Location: admin_login.php");
    exit;
}

if (isset($_SESSION['admin_id'])) {
    header("Location: admindashboard.php");
    exit;
}

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "task_buddy_db";

$error = '';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $admin_username = trim($_POST['admin_username'] ?? '');
    $admin_password = trim($_POST['admin_password'] ?? '');

    if ($admin_username === '' || $admin_password === '') {
        $error = "Please enter both username and password.";
    } else {
        try {
            $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
            $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

For survey Creation

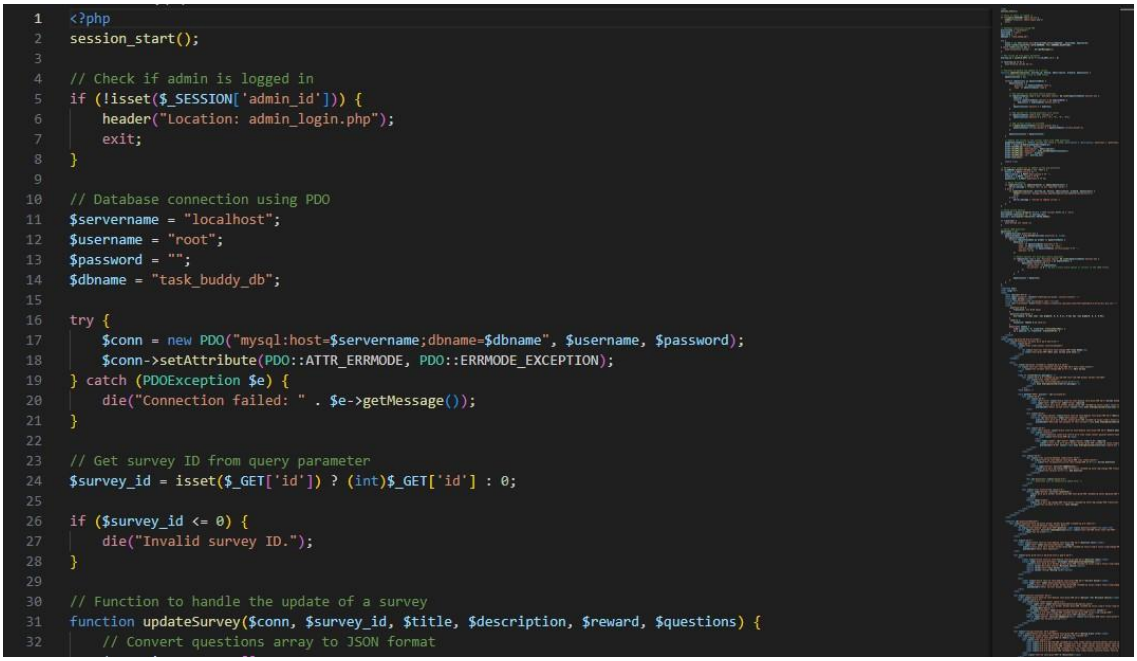
```php
<?php
// **IMPORTANT:  session_start() MUST be the very first thing in your PHP file, before any HTML or other output!*
session_start();

// Database connection using PDO
$servername = "localhost"; // Change if necessary
$username = "root"; // Change if necessary
$password = ""; // Change if necessary
$dbname = "task_buddy_db"; // Updated database name

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // Set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch(PDOException $e) {
    die("Connection failed: " . $e->getMessage());
}

// Function to handle the creation of a new survey
function createSurvey($conn, $title, $description, $reward, $questions) {
    // Convert questions array to JSON format
    $questionsJson = [];

    foreach ($questions as $questionData) {
        $questionJson = [
            'question' => $questionData['text'],
            'type' => $questionData['type']
        ];

        // Add options for multiple choice questions
        if ($questionData['type'] === 'multiple choice' && isset($questionData['options'])) {
            $options = [];
```

For survey participation



```php
<?php
require_once('../includes/session_helper.php');

// Check if user is logged in and session is valid
isLoggedIn();

require_once('../user/db.php');

$survey_id = isset($_GET['id']) ? intval($_GET['id']) : 0;
$error = '';
$success = '';
$survey = null;
$existing_response = false;
$questions = [];

if ($survey_id <= 0) {
    $error = "Invalid survey ID provided.";
} else {
    try {
        $conn = getDBConnection();

        // Get survey details
        $stmt = $conn->prepare("SELECT * FROM surveys WHERE id = :survey_id AND status = 'active'");
        $stmt->execute(['survey_id' => $survey_id]);
        $survey = $stmt->fetch();

        if (!$survey) {
            $error = "Survey not found or inactive. Please try another survey.";
        } else {
            // Check if user already completed this survey
            $stmt = $conn->prepare("SELECT id FROM survey_responses WHERE user_id = :user_id AND survey_id = :sur
            $stmt->execute(['user_id' => $_SESSION['user_id'], 'survey_id' => $survey_id]);
```

For edit survey



```php
<?php
session_start();

// Check if admin is logged in
if (!isset($_SESSION['admin_id'])) {
    header("Location: admin_login.php");
    exit;
}

// Database connection using PDO
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "task_buddy_db";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    die("Connection failed: " . $e->getMessage());
}

// Get survey ID from query parameter
$survey_id = isset($_GET['id']) ? (int)$_GET['id'] : 0;

if ($survey_id <= 0) {
    die("Invalid survey ID.");
}

// Function to handle the update of a survey
function updateSurvey($conn, $survey_id, $title, $description, $reward, $questions) {
    // Convert questions array to JSON format
```

For User management

```php
$password = "";
$dbname = "task_buddy_db";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // Fetch admin name
    $stmtAdmin = $conn->prepare("SELECT admin_name FROM admins WHERE id = :id");
    $stmtAdmin->execute(['id' => $_SESSION['admin_id']]);
    $adminData = $stmtAdmin->fetch(PDO::FETCH_ASSOC);
    $admin_name = $adminData['admin_name'] ?? "Admin";

    // Fetch users with their statistics
    $stmt = $conn->query("
        SELECT
            u.*,
            COUNT(DISTINCT sr.id) as completed_surveys,
            COALESCE(SUM(CASE WHEN t.type = 'withdrawal' AND t.status = 'approved' THEN t.amount ELSE 0 END), 0)
        FROM users u
        LEFT JOIN survey_responses sr ON u.id = sr.user_id
        LEFT JOIN transactions t ON u.id = t.user_id
        GROUP BY u.id
        ORDER BY u.created_at DESC
    ");
    $users = $stmt->fetchAll(PDO::FETCH_ASSOC);

} catch (PDOException $e) {
    die("Connection failed: " . $e->getMessage());
}
```

For Response submission

```php
<?php
session_start();

// Database connection using PDO
$servername = "localhost"; // Change if necessary
$username = "root"; // Change if necessary
$password = ""; // Change if necessary
$dbname = "task_buddy_db"; // Updated database name

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    die("Could not connect to the database $dbname :" . $e->getMessage());
}

// Handle form submission
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $survey_id = $_POST['survey_id'];
    $user_id = $_SESSION['user_id'] ?? null; // Use session user_id

    if (!$user_id) {
        http_response_code(401);
        echo json_encode(['success' => false, 'message' => 'User not logged in.']);
        exit;
    }

    // Check if the user has already participated in the survey
    $checkStmt = $conn->prepare("SELECT * FROM survey_responses WHERE user_id = :user_id AND survey_id = :survey_
    $checkStmt->execute(['user_id' => $user_id, 'survey_id' => $survey_id]);
    $existingParticipation = $checkStmt->fetch(PDO::FETCH_ASSOC);
```

For Withdrawal request handling

```php
<?php
session_start();

if (!isset($_SESSION['admin_id'])) {
    header("Location: admin_login.php");
    exit;
}

// Database connection using PDO
$servername = "localhost"; // Change if necessary
$username = "root"; // Change if necessary
$password = ""; // Change if necessary
$dbname = "task_buddy_db"; // Updated database name

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // Set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // Fetch admin name from database using session admin_id
    if (isset($_SESSION['admin_id'])) {
        $stmtAdmin = $conn->prepare("SELECT admin_name FROM admins WHERE id = :id");
        $stmtAdmin->execute(['id' => $_SESSION['admin_id']]);
        $adminData = $stmtAdmin->fetch(PDO::FETCH_ASSOC);
        if ($adminData && !empty($adminData['admin_name'])) {
            $admin_name = $adminData['admin_name'];
        } else {
            $admin_name = "Admin";
        }
    } else {
        $admin_name = "Admin";
    }
```

For withdraw money

withdrawals > withdraw.php

```php
<?php
session_start();

// Check if user is logged in
if (!isset($_SESSION['user_id'])) {
    header("Location: ../user/auth/login.php");
    exit();
}

// Database connection using PDO
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "task_buddy_db";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch(PDOException $e) {
    die("Connection failed: " . $e->getMessage());
}

// Handle form submission
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $amount = $_POST['amount'];
    $payment_method = $_POST['payment_method'];
    $user_id = $_SESSION['user_id'];

    // Initialize optional fields
    $bank_name = null;
    $account_number = null;
    $bkash_number = null;
```

For support submission

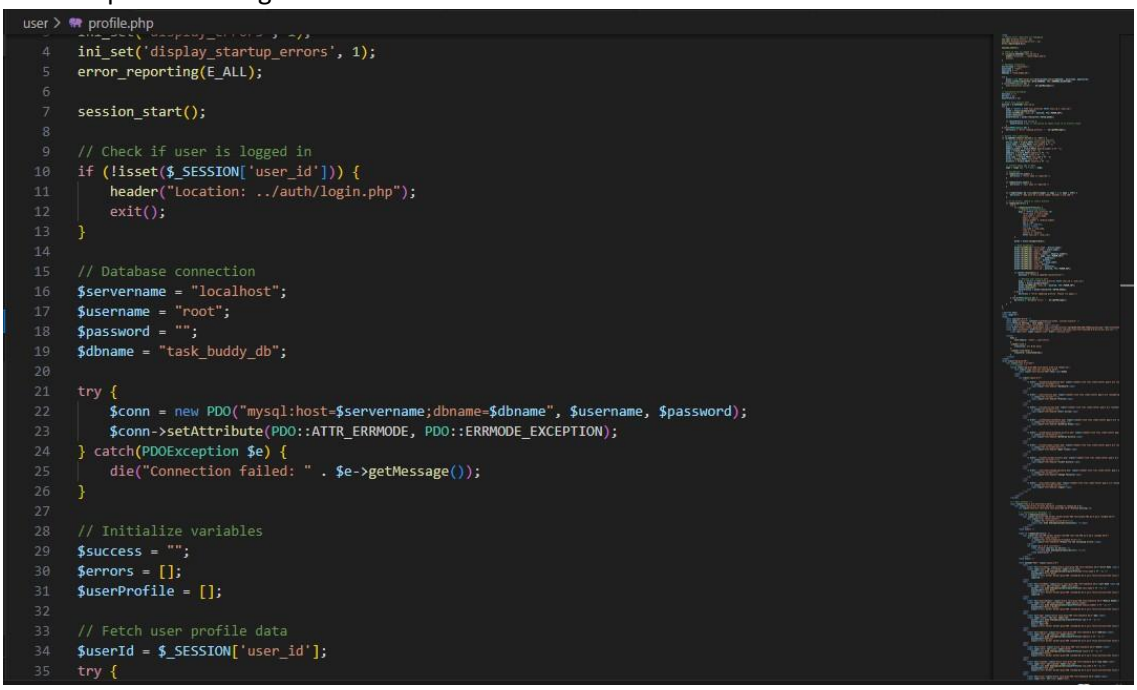
tickets > open_ticket.php

```php
<?php
session_start();

// Check if user is logged in
if (!isset($_SESSION['user_id'])) {
    header("Location: ../user/auth/login.php");
    exit();
}

// Database connection using PDO
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "task_buddy_db";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch(PDOException $e) {
    die("Connection failed: " . $e->getMessage());
}

// Handle form submission
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $subject = $_POST['subject'];
    $message = $_POST['message'];
    $user_id = $_SESSION['user_id'];

    // Insert ticket into database
    $stmt = $conn->prepare("INSERT INTO tickets (user_id, subject, message, status, created_at) VALUES (:user_id,
    $stmt->execute([
        'user_id' => $user_id,
```

For ticket history

For support management

For user profile management



```php
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

session_start();

// Check if user is logged in
if (!isset($_SESSION['user_id'])) {
    header("Location: ../auth/login.php");
    exit();
}

// Database connection
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "task_buddy_db";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch(PDOException $e) {
    die("Connection failed: " . $e->getMessage());
}

// Initialize variables
$success = "";
$errors = [];
$userProfile = [];

// Fetch user profile data
$userId = $_SESSION['user_id'];
try {
```

For Change password

```php
<?php
session_start();

// Check if user is logged in
if (!isset($_SESSION['user_id'])) {
    header("Location: login.php");
    exit();
}

require_once('../db.php');

$error = '';
$success = '';

// Handle form submission
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $current_password = $_POST['current_password'];
    $new_password = $_POST['new_password'];
    $confirm_password = $_POST['confirm_password'];

    // Validate inputs
    if (empty($current_password) || empty($new_password) || empty($confirm_password)) {
        $error = "All fields are required.";
    } elseif ($new_password !== $confirm_password) {
        $error = "New passwords do not match.";
    } elseif (strlen($new_password) < 6) {
        $error = "New password must be at least 6 characters long.";
    } else {
        try {
            $conn = getDBConnection();

            // Verify current password
```

For admin account setup

Admin > create_admin_with_password_8.sql

```sql
-- SQL to create a new admin with username 'admin' and password '8'
-- Hashed password for '8' using PHP's password_hash()

-- Ensure the admins table exists
CREATE TABLE IF NOT EXISTS admins (
    id INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
    admin_name VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- Insert new admin with username 'admin' and password '8'
-- The hashed password below is for the plain text '8'
INSERT INTO admins (admin_name, password)
VALUES ('admin', '$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi')
ON DUPLICATE KEY UPDATE password = '$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi';

-- Alternative: Insert with username 'H' and password '8'
INSERT INTO admins (admin_name, password)
VALUES ('H', '$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi')
ON DUPLICATE KEY UPDATE password = '$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi';

-- Query to verify the admin was created
SELECT id, admin_name, created_at FROM admins WHERE admin_name IN ('admin', 'H');
```

For log out

```php
// Unset all of the session variables
$_SESSION = array();

// If it's desired to kill the session, also delete the session cookie.
if (ini_get("session.use_cookies")) {
    $params = session_get_cookie_params();
    setcookie(
        session_name(),
        '',
        time() - 42000,
        $params["path"],
        $params["domain"],
        $params["secure"],
        $params["httponly"]
    );
}

// Destroy the session
session_destroy();

// Clear any remember me cookies
setcookie(session_name(), '', time() - 3600, "/");

// Set a logout success message
session_start();
$_SESSION['logout_success'] = "You have been successfully logged out.";

// Redirect to the login page
header("Location: ../index.php");
exit();
?>
```

Summary

In this chapter, I discussed the overall design and implementation of the TaskBuddy Online Survey System. The chapter explained how the major modules such as user authentication, survey creation, participation, response submission, withdrawal request handling, support system and admin operations are implemented. Different diagrams like use case, activity, sequence, class and ER diagrams were used to represent the system workflow clearly. The coding section also highlighted the important logic required to run the system. Overall, this chapter provides a complete overview of how TaskBuddy functions internally and how each module is connected to ensure a smooth online survey experience for users. (Fan, 2015)

Chapter 3 Software Testing

3.1 Introduction

Software testing is a very important part of SDLC that ensures all modules are working correctly (individually and collectively). The chapter represents the testing conducted to verify and validate all the functional and non-functional requirements. The main objective is to find and identify system errors, how system behaves, and interconnected modules are working as per requirements.

3.2 Testing Features

3.2.1 Feature to Be Tested

a. User Sign Up
b. User Login
c. Survey Creation
d. Survey Participation
e. Survey Editing
f. Withdraw Money
g. Support Submission
h. Ticket History
i. Support Submission

3.3 Testing Strategies

3.3.1 Test Approach

The testing approaches I adopted in this project are described below:

3.3.1 Black Box Testing:
Black box testing methodology was applied to test the input output behavior of TaskBuddy, without knowing the internal code on hand. It has been tested according to the expected outputs on functionalities such as registration, login, creation of survey, participation and

submission of response. This allowed every function to do the right thing on both legal and illegal inputs.

2. Functional Testing:
Every component of the system was tested separately to check if it behaves as expected. For instance creating a survey, editing it, viewing results, managing support tickets and handling withdrawal request were tested to make sure the flow is smooth.

3. Integration Testing:
This approach checked the behavior of interconnected modules working together. For example:

- Survey Creation → Survey Participation → Response Submission
- Withdrawal Request → Withdrawal Approval
- User Ticket Submission → Admin Support Reply

Integration testing ensured that the system flow remains stable across multiple modules.

4. Usability Testing:
System interfaces, validation of forms, messages of error, navigation and responsiveness were tested to ensure users can effectively use the system. This testing is designed to deliver a user-friendly experience for everyone.

5. Security Testing:
Security testing checked whether the system prevents unauthorized access. Examples:

- Invalid login attempts
- Session management
- Access limitations (admin/user roles)

These tests ensured that the system remains secure and protects user information

3.3.2 Pass/Fail Criteria

A test case is marked **Pass** if:

- The output matches the expected result
- The function works correctly with valid input
- The system shows proper error messages for invalid input
- Modules work without conflict
- The system handles invalid user behavior safely

A test case is marked **Fail** if:

- The output does not match the expected behavior
- The system fails to validate correct input
- Server errors occur during execution
- The interface behaves incorrectly
- The system fails during module integration

3.4 System Testing (Test Cases with Report)

Table: 3.4.1: Test Case: 01 Sign up

Test case:		**Test case name:** Sign Up				
System: Taskbuddy System		**Sub-system:** User Authentication				
Designed by: Sudip Datta		**Design Date:** 02/09/2025				
Executed by: Sudip Datta		**Execution Date:** 02/09/2025				
		Description: The user sign up on the system by using the valid information				
		Precondition: Users visit the sign up page				
Step	**Name**	**Username**	**Email**	**Password**	**Pass /Fail**	**Comments**
1	Sudip Datta	sudip	Sd121@gmail.com	992288	Pass	Valid data and sign up successful
2		sudip	sd@gmail.com	992288	Fail	User must enter name
3	Sudip Datta	sudip		992288	Fail	Email required for sign up

4	Dip Roy	dip	dip@gmail.com	657565	Pass	Valid data and sign up success
5	Dip Roy	dip	Dip1@gmail.com		Fail	Password is required
Post Condition: User needs to put all valid information to sign up on the system						

Table: 3.4.2: Test Case 02: Log In

Test case:		**Test case name:** log in		
System: Taskbuddy System		**Sub-system:** User Authentication		
Designed by: Sudip Datta		**Design Date:** 02/07/2025		
Executed by: Sudip Datta		**Execution Date:** 02/07/2025		
	Description: The user log in on the system by using the valid information			
	Precondition: must need to sign up before			
Step	**Email**	**Password**	**Pass /Fail**	**Comments**
1	Sd121@gmail.com	992288	Pass	Valid data and sign up successful

2	Sd111@gmail.com	992288	Fail	Email Invalid
3		992288	Fail	Email required
4	dip@gmail.com	657565	Pass	Valid data and sign up success
5	Dip1@gmail.com		Fail	Password is required
Post Condition: User needs to put all valid information to log in on the system				

Table: 3.4.3: Test Case 03: Survey Creation

Test case:	**Test case name:** Survey Creation
System: Taskbuddy System	**Sub-system:** User Authentication
Designed by: Sudip Datta	**Design Date:** 02/09/2025
Executed by: Sudip Datta	**Execution Date:** 02/09/2025
	Description: Admin can create survey
	Precondition: Admin must log in before

Step	Survey Title	description	Amount	Pass /Fail	Comments
1	JP Morgan	Write about JP Morgan	5	Pass	Valid data and survey creation successful
2		Write about JP Morgan	5	Fail	User must enter name
3	JP Morgan		5	Fail	Description required
4	JP Morgan	Write about JP Morgan		Fail	Amount required
Post Condition: User needs need to give all valid information for create survey					

Table: 3.4.4: Test Case 04: Survey Participation

Test case:	**Test case name:** Survey Participation
System: Taskbuddy System	**Sub-system:** User Authentication
Designed by: Sudip Datta	**Design Date:** 02/09/2025
Executed by: Sudip Datta	**Execution Date:** 02/09/2025

	Description: User can start survey for earn money				
	Precondition: User must log in first				
Step	**Your Name**	**Job Title**	**Rate Us**	**Pass /Fail**	**Comments**
1	Sudip Datta	CTO	5	Pass	Successfully completed survey
2		CTO	5	Fail	Name required
3	Sudip Datta		5	Fail	Job Title required
4	Sudip Datta	CTO		Fail	rate required
Post Condition: User needs need to give all information for complete survey					

Table: 3.4.5: Test Case 05: Survey Editing

Test case:	**Test case name:** Survey editing

System: Taskbuddy System		**Sub-system:** User Authentication			
Designed by: Sudip Datta		**Design Date:** 02/09/2025			
Executed by: Sudip Datta		**Execution Date:** 02/09/2025			
	Description: Admin can edit survey				
	Precondition: Admin must log in before				
Step	**Survey Title**	**description**	**Amount**	**Pass /Fail**	**Comments**
1	JP	Write about JP Morgan	5	Pass	Valid data and survey creation successful
2		Write about JP Morgan	5	Fail	User must enter name
3	JP		5	Fail	Description required
4	JP	Write about JP Morgan		Fail	Amount required
Post Condition: User needs need to give all valid information for edit survey.					

Table: 3.4.6: Test Case 06: Withdraw Money

Test case:	**Test case name:** Withdraw Money
System: Taskbuddy System	**Sub-system:** User Authentication
Designed by: Sudip Datta	**Design Date:** 02/09/2025
Executed by: Sudip Datta	**Execution Date:** 02/09/2025
	Description: User can withdraw money in a secure gateway
	Precondition: User must log in first and have balance

Step	**Withdraw Amount**	**Number**	**Pass /Fail**	**Comments**
1	1	01987936189	Pass	Successfully withdraw money
2		5	Fail	Withdraw amount required
3	1		Fail	Number required

Post Condition: User needs need to give all information for withdraw money

Table 3.4.7: Test Case: 07: Support Submission

Test case:		**Test case name:** Support Submission		
System: Taskbuddy System		**Sub-system:** User Authentication		
Designed by: Sudip Datta		**Design Date:** 02/07/2025		
Executed by: Sudip Datta		**Execution Date:** 02/07/2025		
	Description: The user can get support by admin			
	Precondition: must need to sign up before			
Step	**Problem Title**	**Description**	**Pass /Fail**	**Comments**
1	Don’t participate any survey	Need help for that matter	Pass	Successfully submit ticket
2		Need help for that matter	Fail	Title required
3	Don’t participate any survey		Fail	description required
Post Condition: User needs to give all information for submit ticket				

Table: 3.4.8:Test Case 08: Ticket History

Test case:		**Test case name:** Ticket History	
System: Taskbuddy System		**Sub-system:** Ticket system	
Designed by: Sudip Datta		**Design Date:** 02/07/2025	
Executed by: Sudip Datta		**Execution Date:** 02/07/2025	
	Description: The user see his ticket position and make reply		
	Precondition: must need to sign in before		
Step	**Reply Admin**	**Pass /Fail**	**Comments**
1	What?	Pass	Successfully reply ticket
2		Fail	Reply admin required
Post Condition: User needs to give all information for reply ticket			

Table: 3.4.9: Test Case 09: Support Management

Test case:	**Test case name:** Ticket History

System: Taskbuddy System		**Sub-system:** Ticket system (Admin)	
Designed by: Sudip Datta		**Design Date:** 02/07/2025	
Executed by: Sudip Datta		**Execution Date:** 02/07/2025	
	Description: The admin can reply user ticket		
	Precondition: must need to sign in before		
Step	**Reply Admin**	**Pass /Fail**	**Comments**
1	What?	Pass	Successfully reply ticket
2		Fail	Reply participant required
Post Condition: User needs to give all information for reply ticket			

3.5 Summary

In this chapter, the testing of the entire TaskBuddy Online Survey is elaborated to make sure that each module is accurate and stable. The main features also have been tested with various mode of testing such as user authentication and survey creation, participation, answer submission, support system, user management and withdrawal process. The individual and inter module behavior was tested by black box testing,

functional testing, integration testing, usability testing and security testing. Valid and invalid test cases have run to ensure the system processes every case correctly. On the whole, this chapter demonstrated that TaskBuddy executes as expected and is robust for practical use.

Chapter 4 Deployment and Maintenance

4.1 Introduction

Here you will find a description about how TaskBuddy Online Survey System was implemented in the real world and how the system will be maintained once it is implemented. Deployment makes the application go live, moves from development to a live server and access by any user anywhere. In the maintenance phase, these are bug fixes, performance enhancements, feature updates and system monitoring/stability as a whole. Release management The SRLC model is followed in order to have a controlled and orderly release process. This chapter describes the process by which TaskBuddy is packaged, released and supported over time. (Heerwegh, 2006)

4.2 Try to follow the SRLC (software release life cycle)

Alpha Phase

The alpha phase includes developing the core modules and performing early-stage testing. In this phase:

1. Core features like registration, login, survey creation and participation are implemented.
2. Basic response submission and ticket system functions are added.
3. Initial testing is done without real users.
4. Early bugs are identified and fixed.
5. Database connectivity and form validation are tested.

Beta Release

The beta release includes a mostly completed version of the TaskBuddy system. In this phase:

1. Survey workflow is integrated (Create → Participate → Submit Response).
2. Admin dashboard modules like user management, support handling and withdrawal management are added.
3. Editing surveys, viewing responses and ticket reply system are introduced.
4. Testing is performed using sample users.
5. System UI is improved for better user experience.

Release Candidate

This version is almost ready for final deployment.
In this phase:

1. Bugs are fixed and system performance is optimized.
2. Security features such as password hashing, session handling and access restrictions are reviewed.
3. Database cleanup and optimization are done.
4. Email notification system is checked (if included in project).
5. Final testing is done to ensure stability.
6. Preparation of documentation and user manual.

Product Release

This is the final version of TaskBuddy available for users. In
this phase:

1. Deploy the system on live server (using hosting with Apache/Nginx).
2. Upload database to live environment.
3. Configure domain and SSL certificate for secure access.
4. Make the system publicly available for admin and users.
5. Monitor errors and fix initial deployment issues.

Maintenance Phase

After deployment, the system enters the maintenance phase.
In this phase:

1. Fix bugs reported by real users.
2. Improve system speed, performance and security.
3. Update features such as survey analytics, notifications or interface improvements.
4. Add new modules based on user demand (ex: advanced reporting, improved withdrawal system). Schedule backup to prevent data loss.

Chapter 5 User Manual

5.1 Introduction

The following is a step by step guide of how to use the TaskBuddy Online Survey

System for Admin and Users. You will learn here how the most common tasks (registration, login as well as creating and participating on surveys, answering questions, requesting withdrawal of your earnings or treating support tickets …) are carried out. All functions are well explained so that even beginner users have no difficulty using the system.

The purpose of this user manual is to assist the users with comprehending and utilizing TaskBuddy system effectively and without any ambiguity or confusion in order to accomplish their tasks. 5. 2 Project Functionalities

5.2 Project Functionalities

For Participant **dashboard**

1. Go to website first
2. Log in in as user

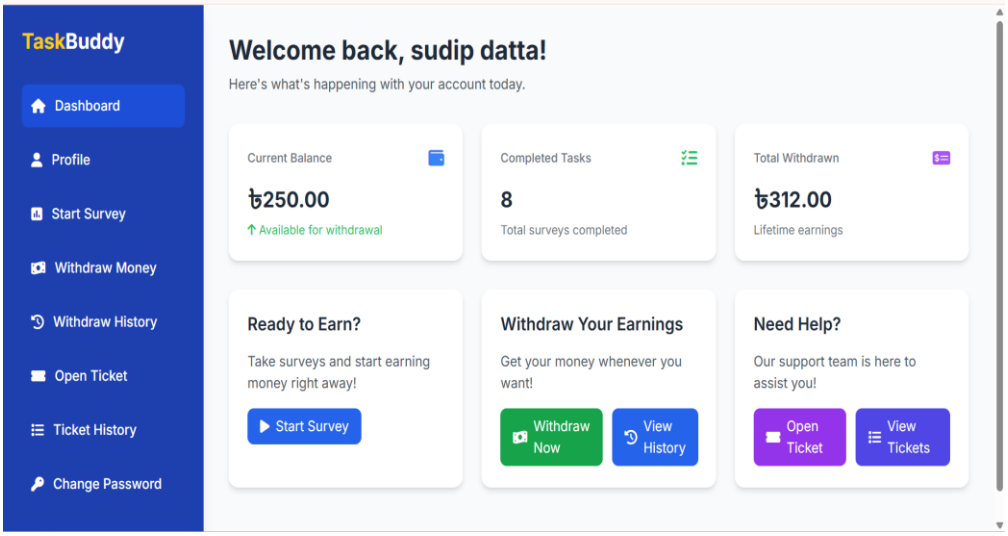


Figure 6.1: Participant Dashboard

For admin dashboard

1. Go to website
2. Log in ad admin

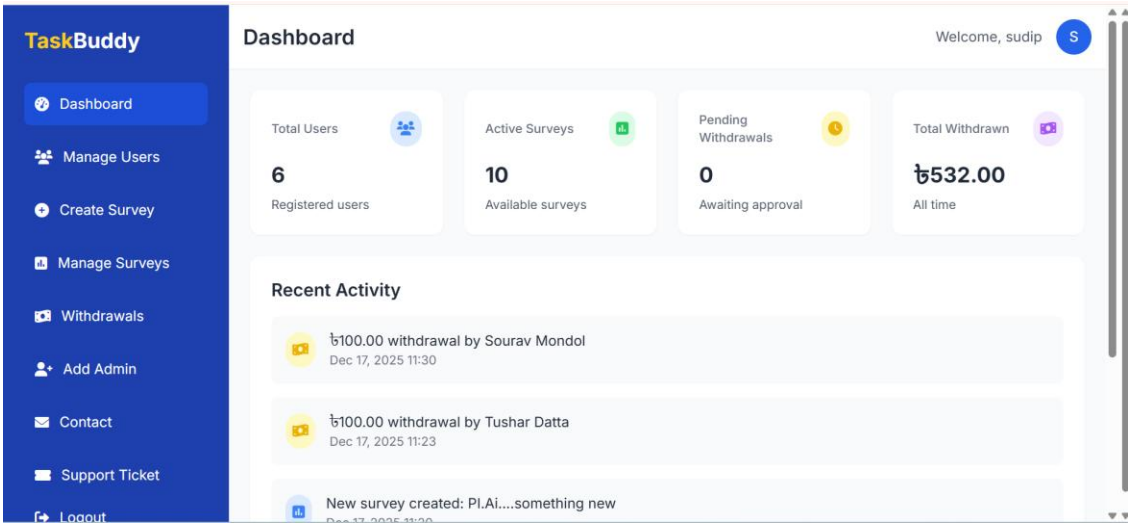


Figure 6.2: Admin Dashboard

For User Sign Up

1. First go to website
2. Then click get started button
3. Then fill up all form
4. At last click sign up

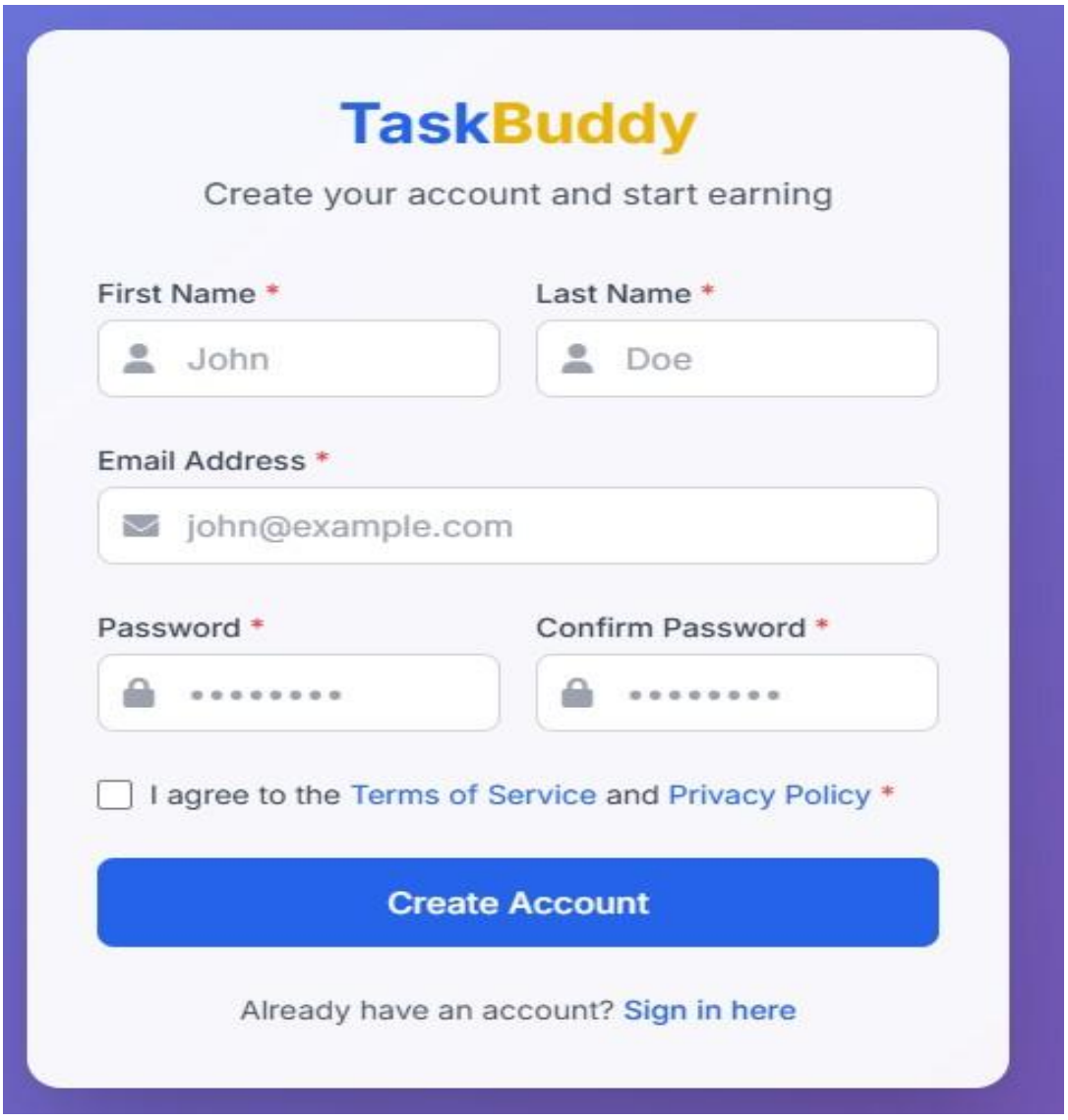


Figure 6.3: Sign Up

For Log In

1. Click log in button
2. Enter email and password
3. Click sign in

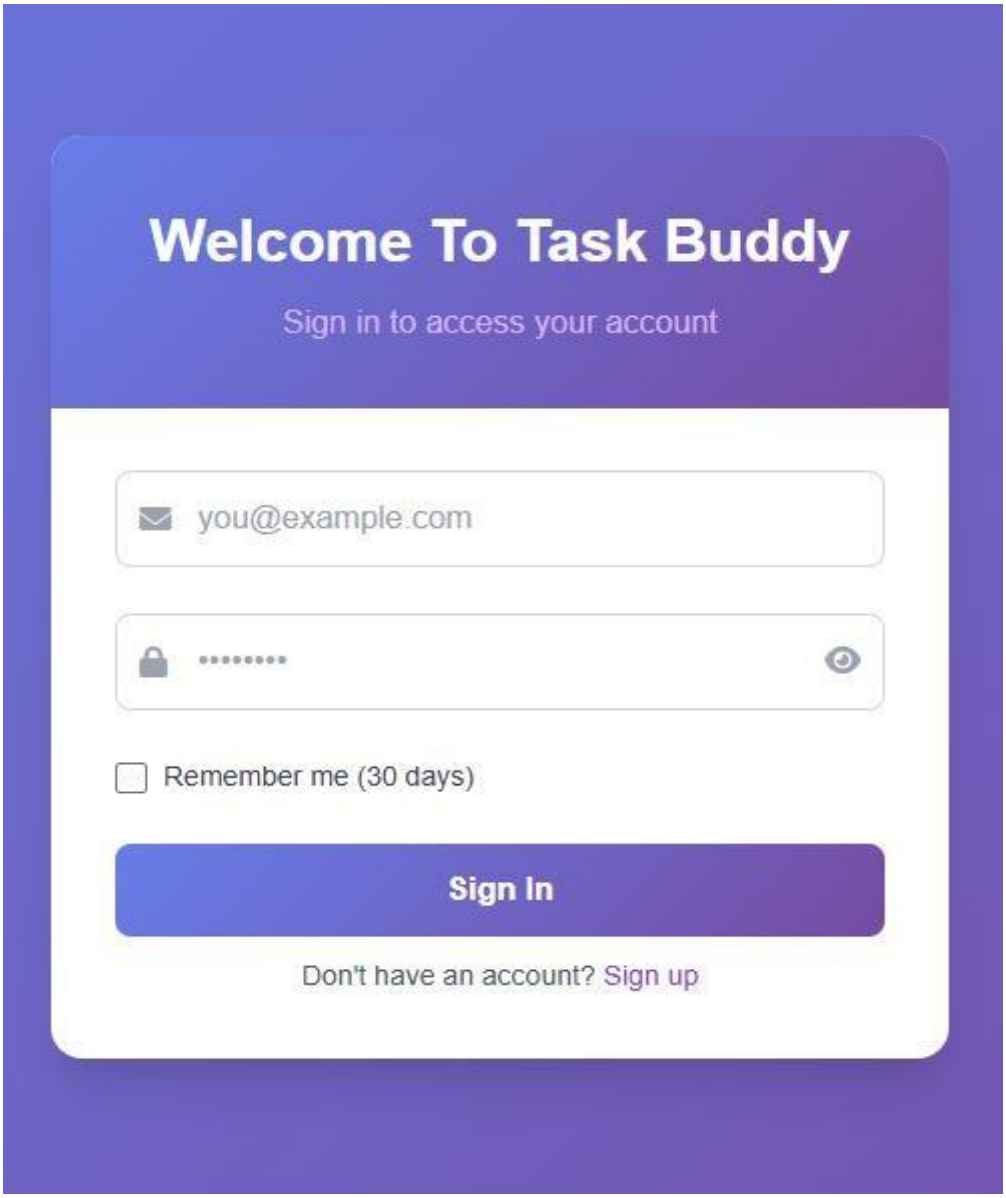


Figure 6.4: Log In

For survey creation

1. Click create survey option
2. Enter all required field and information
3. Press create survey button

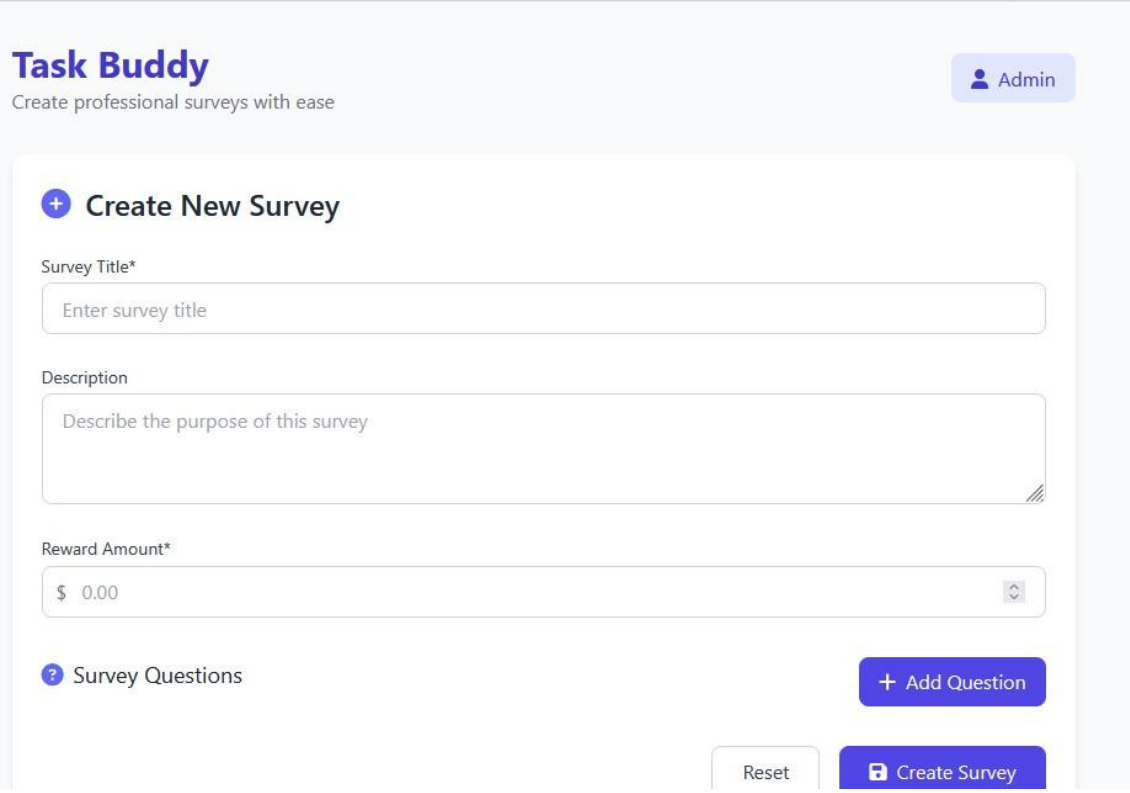


Figure 6.5: Survey Creation

For survey participation

1. click start survey
2. click start task
3. enter survey response
4. click submit survey

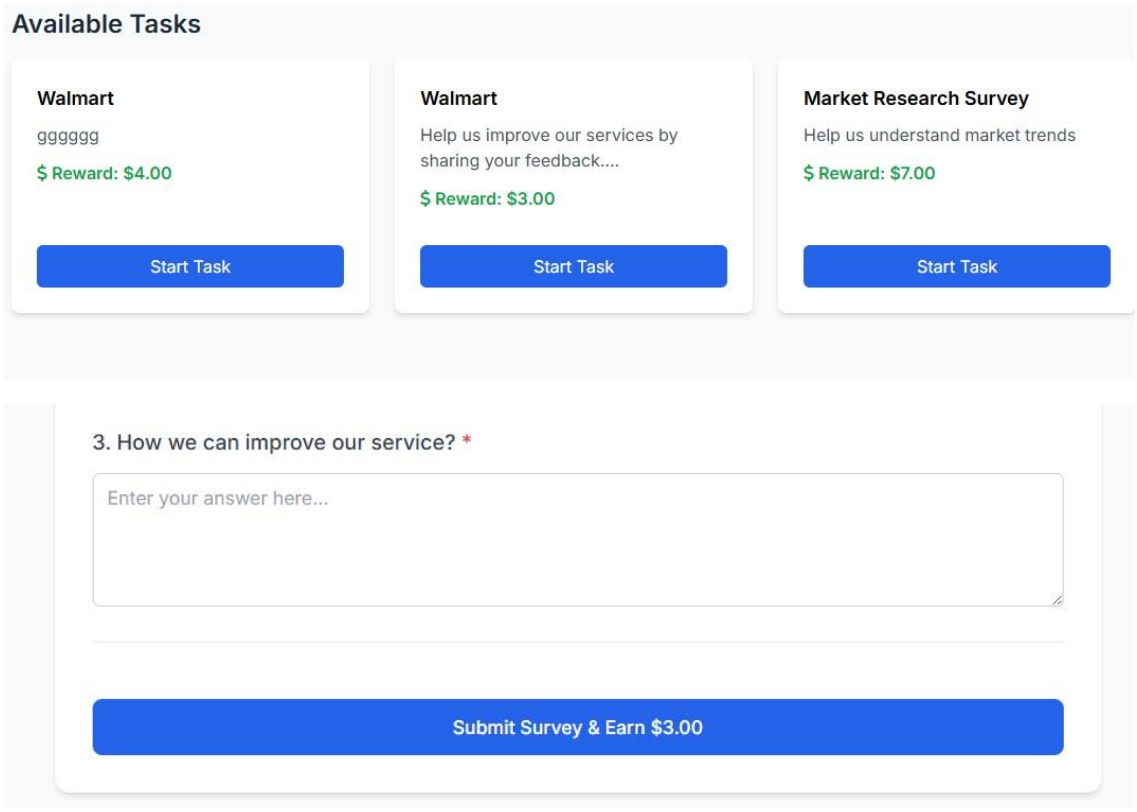


Figure 6.6: Survey Participation

For survey editing

1.click manage survey

2. click edit survey

3. enter information

4. Click save change

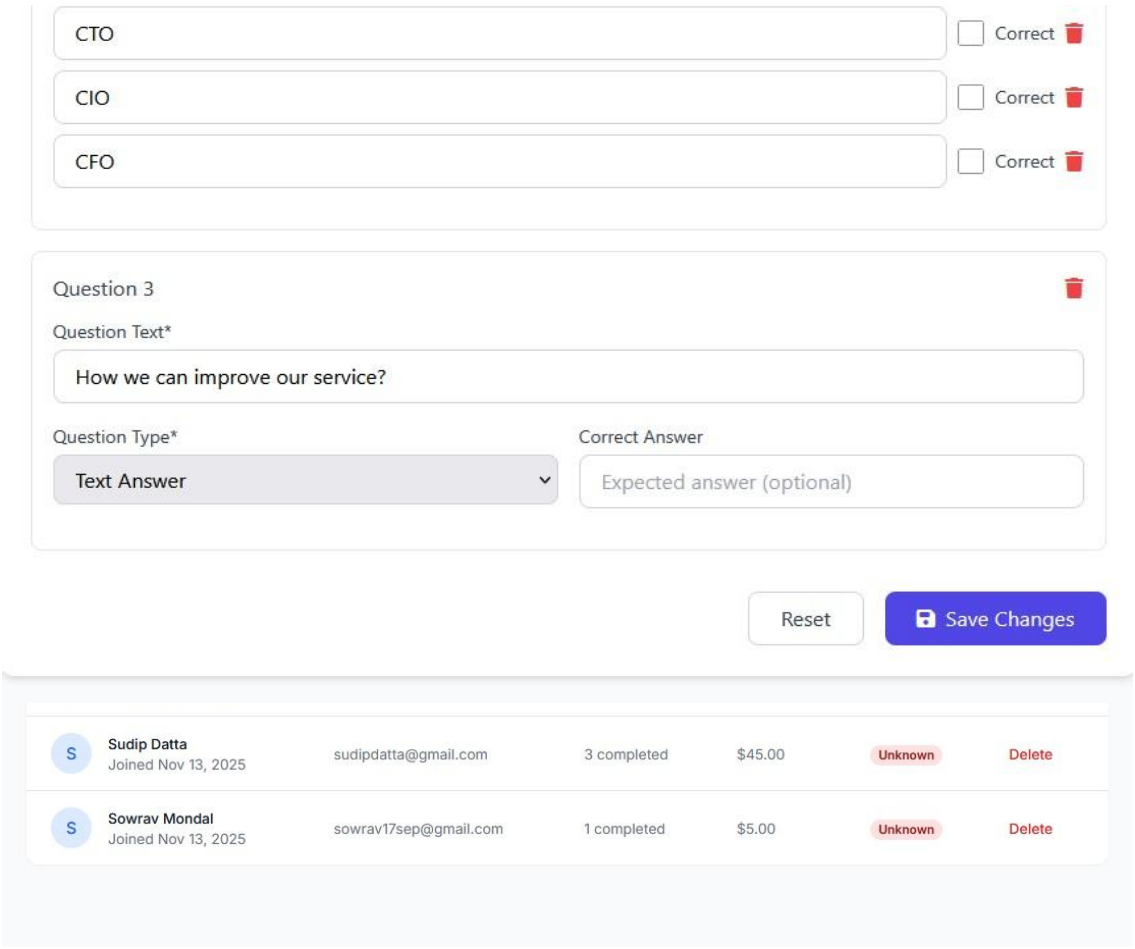


Figure 6.7: Survey Editing

For response submission

1. click manage survey
2. click view and details
3. download response

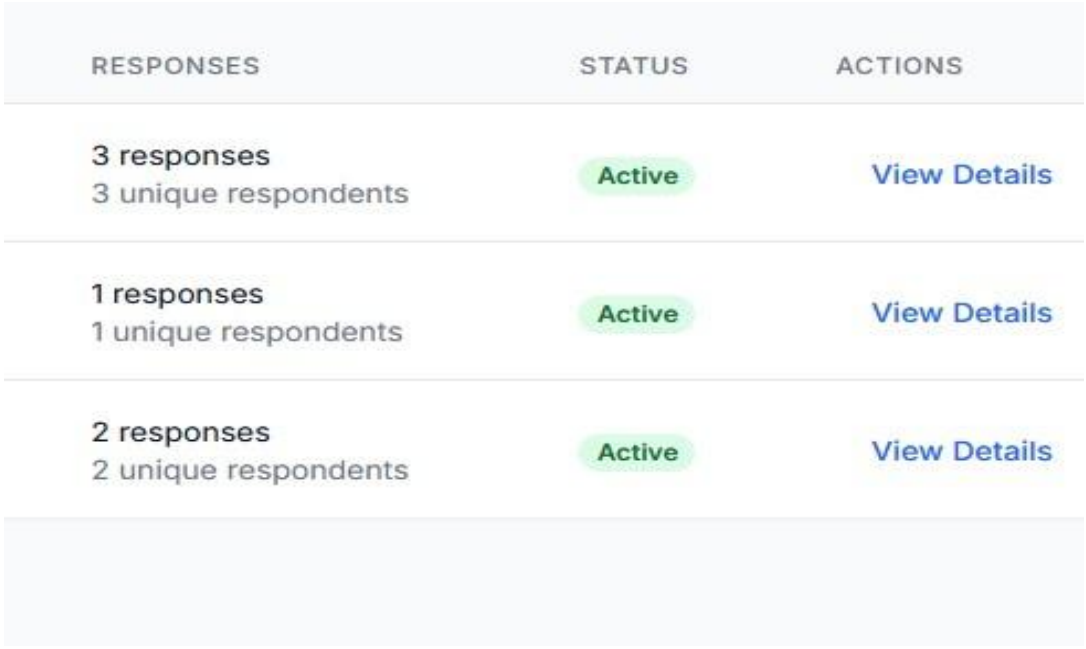


Figure 6.8: Response Submission

For Withdraw request handling

1. click withdrawal
2. click approve or reject

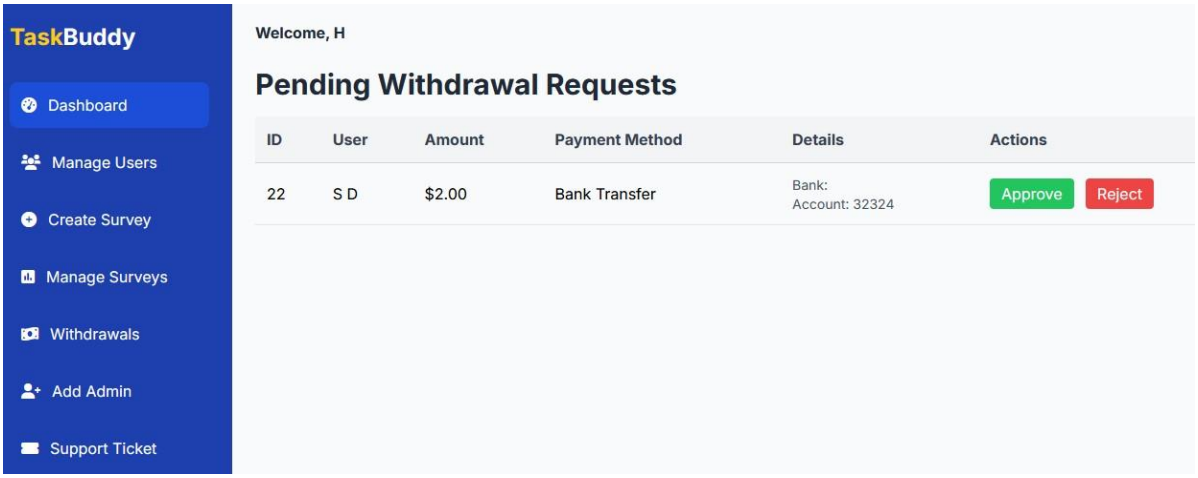


Figure 6.9: Withdraw Request Handling

For withdraw money

1.click withdraw money

2. enter amount and number

3. click withdraw money

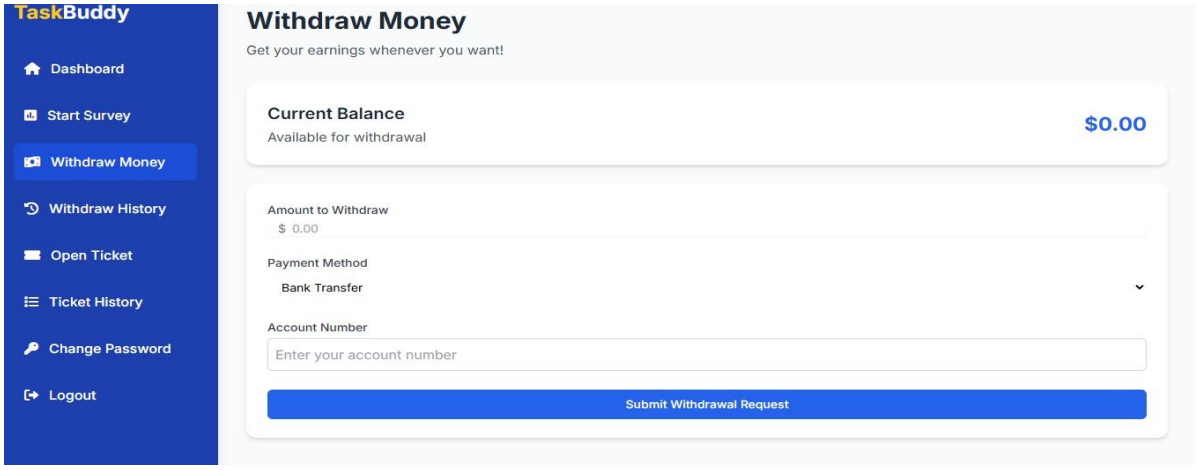


Figure 6.10: Withdraw Money

For support submission

1. click open ticket
2. enter subject and description
3. click submit ticket

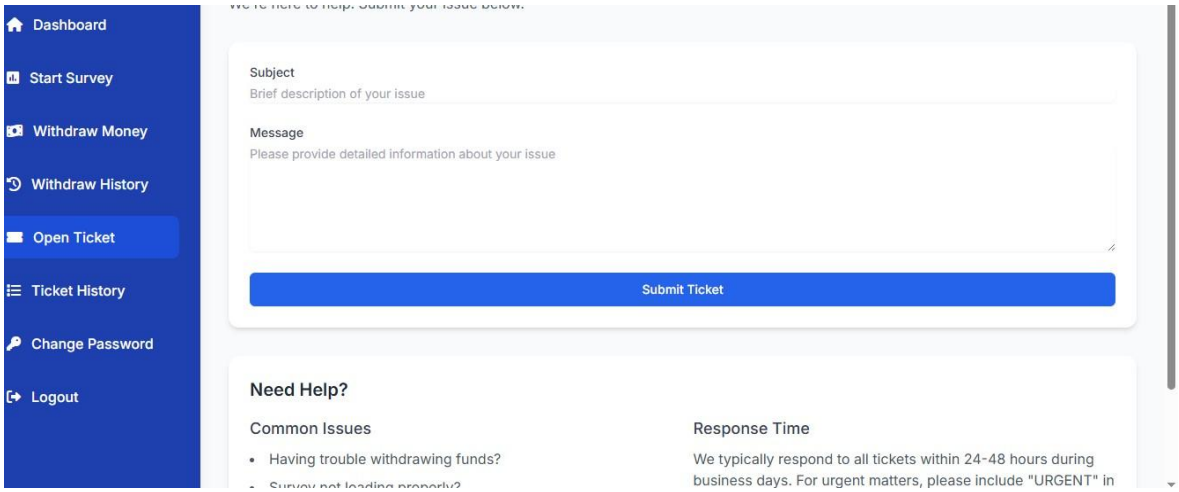


Figure 6.11: Support Submission

For Ticket history

1. click ticket history
2. click view and reply
3. reply admin response

4. click send reply

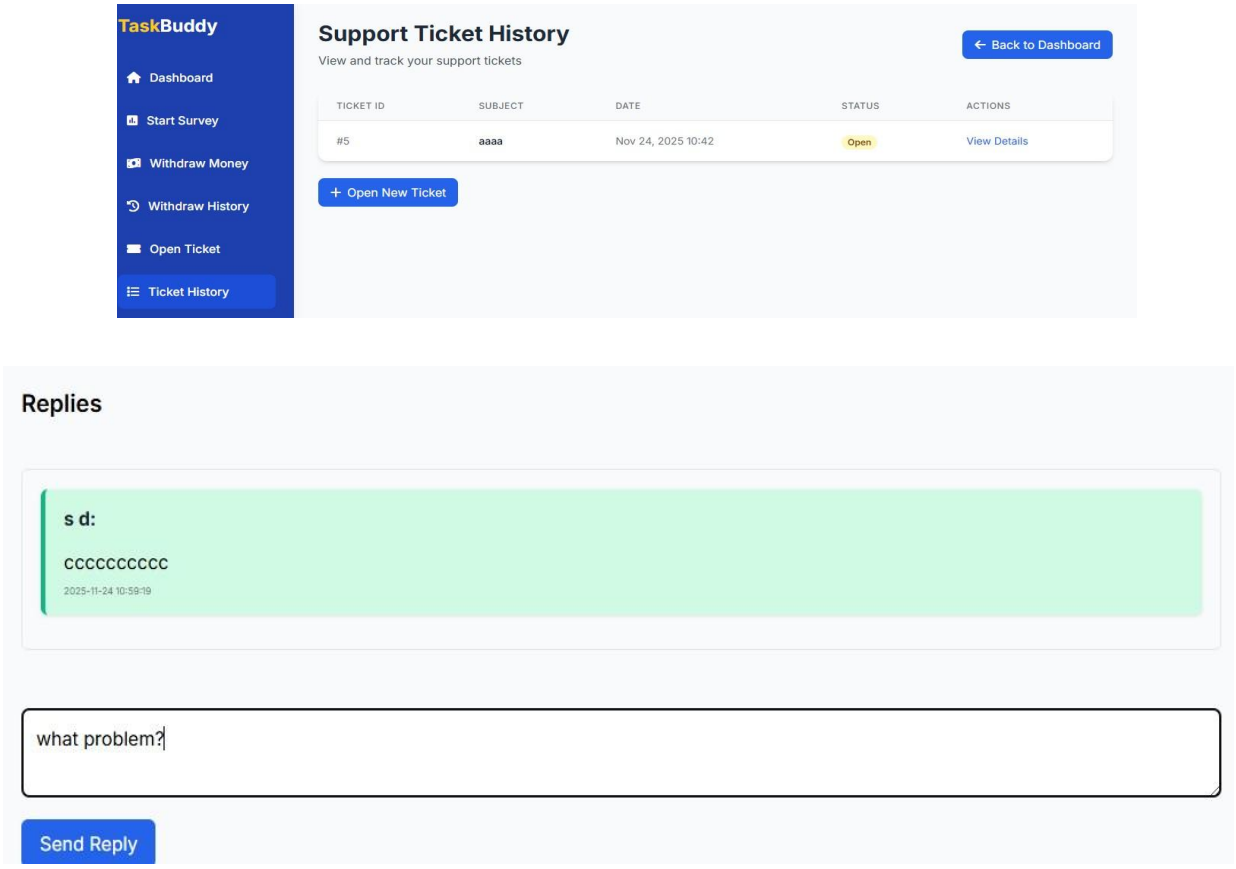


Figure 6.12: Ticket History

For user profile management

1. Click profile button
2. Enter update information
3. Click update profile

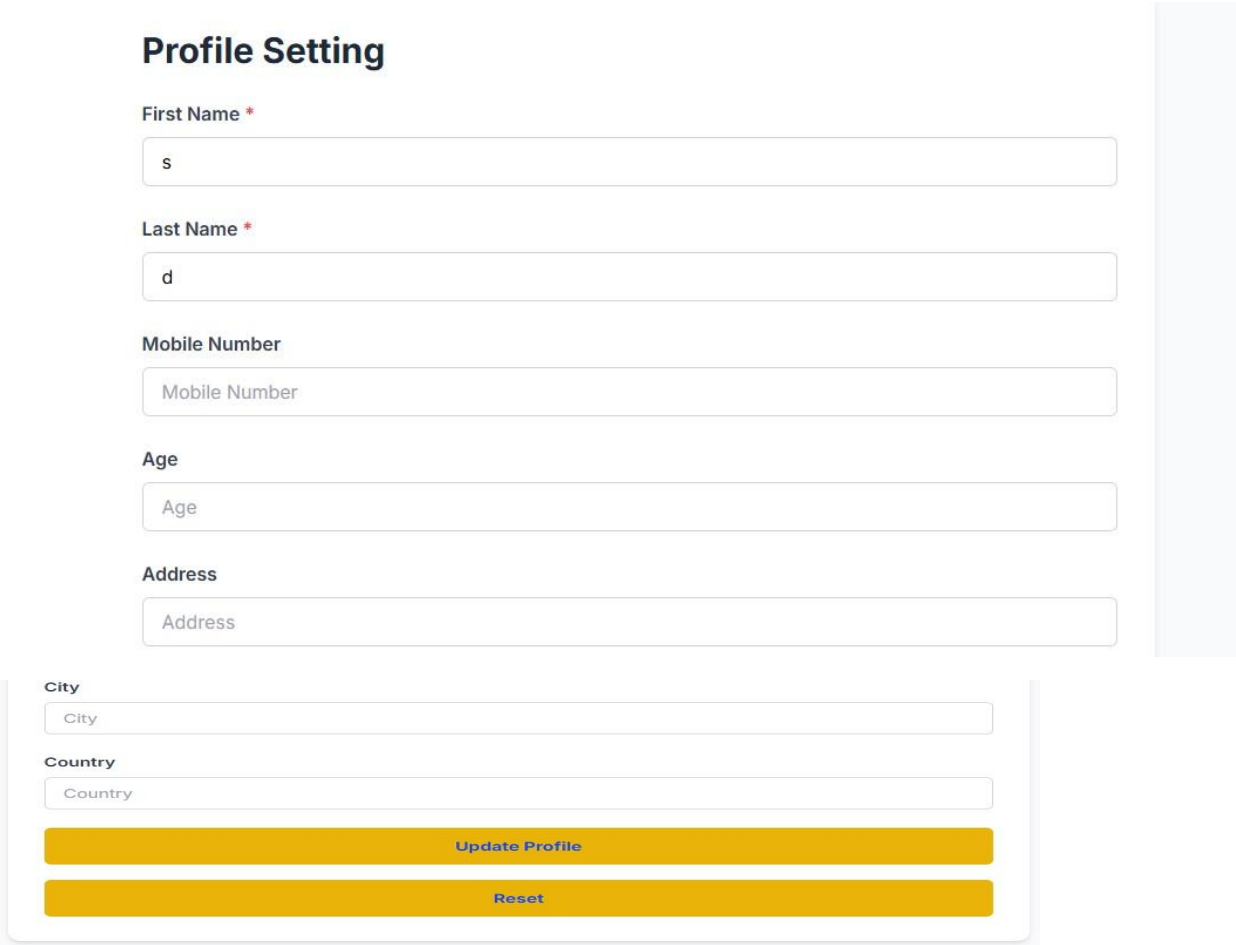


Figure 6.13: User Profile Management

For change password

- change password
- enter current password and new password
- click change password

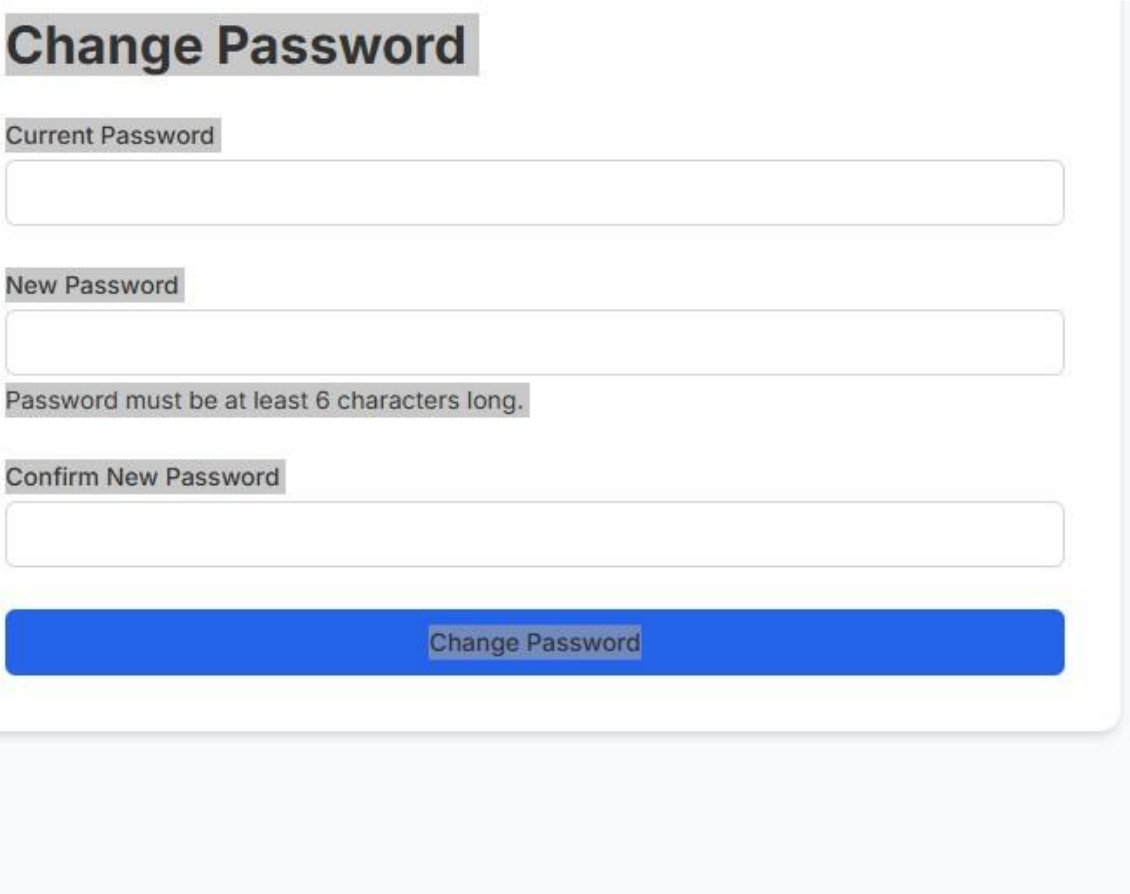


Figure 6.14: Change Password

For admin account setup

1. Click add admin button
2. Enter admin username and password
3. Click create admin

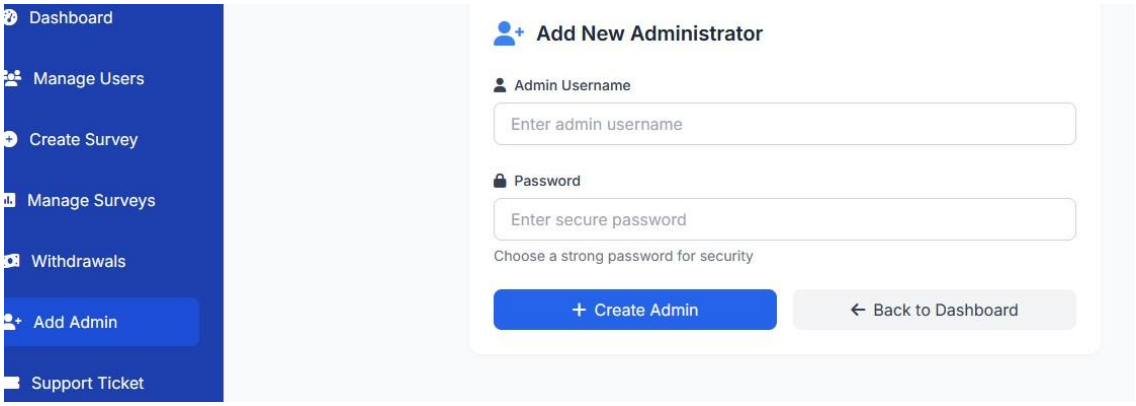


Figure 6.15 : Admin Account Setup

For log out

1. Click log out

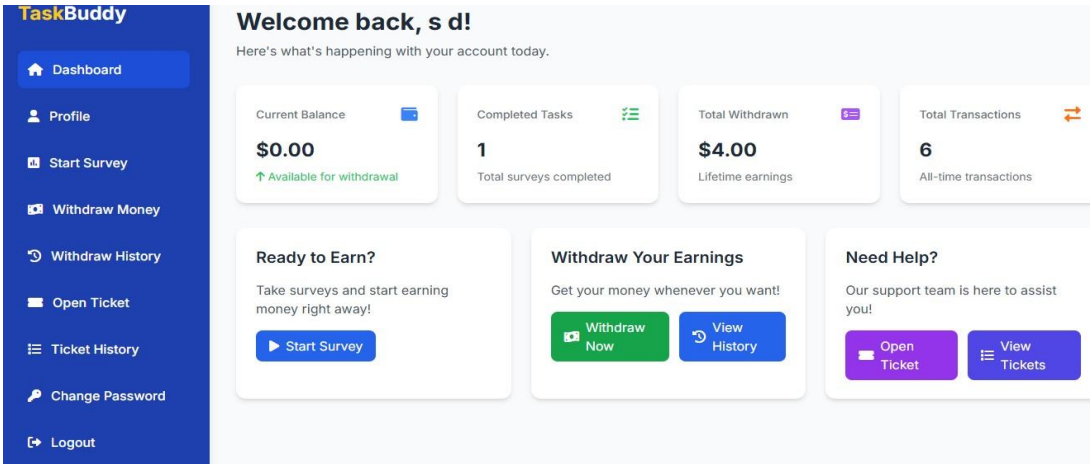


Figure 6.16: Log Out

5.3 Summary

Here, the user manual is very bright and easy to understand and use. All functionality works perfectly and the UI is bold and looks attractive. From sign up to log out any user feels confident in using that website feature. Here I already provided the line-by-line step for how to use every single feature. Here using HTML5, CSS3 and JS in front end the UI is very flexiable and smooth.

Chapter 6 Project Summary

6.1 Introduction

This paper has briefly described the TaskBuddy Online Survey System. It presents the goals of the project, an overview of implemented features and tools developed, as well as difficulties encountered in implementation and testing, with a description of the final result. The chapter also presents limitations of the project, scope and future work, which can be further worked upon to enhance the system. •Lastly, a brief conclusion is given to demonstrate how TaskBuddy achieves its original targets and where it can be extended for better perfection.

6.2 Project Limitation

While TaskBuddy is able to successfully provide the fundamental functions of an incentive-based online survey platform several restrictions were imposed during development as a result of real-world limitations. Below are the major differences.

6.2.1 Time Constraints

The project was developed within academic timeline..

6.2.2 Budget Constraints

Open source tools are used in project and low cost hosting is done deployments. * Paid services -> Not used; Robust cloud hosting, CDN Easy in heavy load.

· Payment gateways were setup also to work in sandbox/test modes so we don't pay fees while developing.

6.6.3 Technical Constraints

- · Native applications (Android/iOS), has not been developed, but system is web-only.
- · Advanced analytics (AI-driven analysis, predictive models) would not be applied because of the complexity and time.

- Real-time notification (SMS) integration was considered but wasn't included. .

6.2.4 Requirement Constraints

Some future work, such as support for multiple languages and a more complete fraud detection process, was postponed.

· Audit logging and a complete compliance module for large clients was never properly finalized.

6.3 Scope

The TaskBuddy Online Survey System covers the essential modules required for a functional reward-based survey platform. The scope of this project includes:

Included in scope:

- Secure user registration and login (participants, company clients, admins).
- Survey creation and management (admin/company).
- Survey participation by registered users.
- Response storage and basic export/reporting.
- User wallet and withdrawal request handling (integration with local payment flow placeholders such as bKash/Nagad).
- Support ticket system and ticket history.
- Role-based admin dashboard for user, survey and withdrawal management.
- Basic analytics and participation statistics.
- Responsive web interface for desktop and mobile browsers.

6.4 Future Work

I planned some future work such as:

1. build mobile app for my Taskbuddy website
2. Taskbuddy will work with international company and survey website
3. Apply multi language system

6.5 Conclusion

The TaskBuddy Online Survey System was created to simplify and organize the collection of surveys for the user and admin alike. In the context of this project, all main functionalities like user authentication, survey creation and participation, as well as submission of answers and withdrawal requests were developed successfully.

Of course, no system is perfect (There weren't enough resources to add or complete some of the more powerful features), I think it will still be great that as an all in one online survey platform. This job gave me a new perspective on system development and enhanced my technical skills as well problem solving techniques.

In summary, TaskBuddy achieves the basic project goals and is expandable in future with additional features and performance improvements.

REFERENCES

Dillman, D. A. (2007). Design effects in the transition to web-based surveys. *American Journal of Preventive Medicine*.

Fan, W. Y. (2015). Factors affecting response rates of the web survey. *A systematic review. Computers in Human Behavior*.

Heerwegh, D. (2006). Effects of personalizing email invitations on response rates. *International Journal of Social Research Methodology*.

Kaplan, A. M. (The challenges and opportunities of social media. Business Horizons). Kaplan, A. M. *Users of the world, unite!*

221-35-880

ORIGINALITY REPORT

18%	10%	3%	15%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	7%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	2%
3	Submitted to NCC Education Student Paper	1%
4	ftkkp.umpsa.edu.my Internet Source	1%
5	Submitted to Midlands State University Student Paper	1%
6	Submitted to Higher Education Commission Pakistan Student Paper	1%
7	Submitted to HELP UNIVERSITY Student Paper	1%
8	Submitted to Kuala Lumpur Infrastructure University College Student Paper	<1%
9	indah.ump.edu.my Internet Source	<1%
10	www.jisc.ac.uk Internet Source	<1%