A Software requirements specification
On

Airport Feedback App

Submitted in Partial Fulfillment for the Award of Degree of Bachelor of Technology in Computer Science and Engineering from Rajasthan Technical University, Kota

MENTOR

Name of faculty: Dr. Nilam Choudhary

(Dept. of Computer Science & Engineering)

SUBMITTED BY:

Akshita Sharma (19ESKCS021)

Akshat Gadodia (19ESKCS021)

COORDINATOR:

Mr. Sumit Mathur

(Dept. of Computer Science & Engineering)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SWAMI KESHWANAND INSTITUTE OF TECHNOLOGY, MANAGEMENT & GRAMOTHAN

Ramnagaria (Jagatpura), Jaipur – 302017

SESSION 2022-23

Table of Contents

Introduction

The Airport Feedback system is the web-based feedback collecting system from the passengers and provides the automatic generation of feedback which is given by passengers. By using this technology we can give feedback in an online system as fast as compared to the existing paper feedback system. This web app is available on the default Wi-Fi of the airport. The user can rate the different departments of the airport by giving them a rating between 1 to 5 and can also provide feedback in the form of a message. The admin can view all the feedback provided by the passengers and can also sort the feedbacks on certain criteria

1.1 Methodology

Waterfall Model:

The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. The waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete

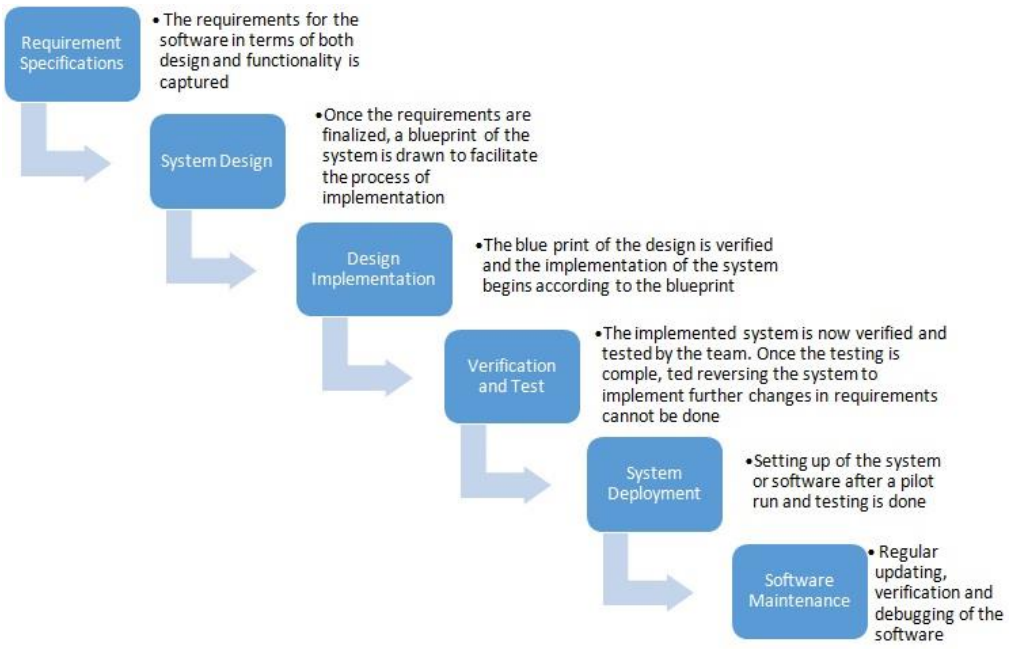

Fig 1.1: Phases of Waterfall Model

The sequential phases in Waterfall model are:

- **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.
- **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
- **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
- **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system:** Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.
- **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap.

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

1.2 Purpose

Using digital feedback software, you can hear the voice of each passenger – about airport staff, airport amenities and general passenger perception about the airport and about the flights. It gives you a platform to recognize key trends, track repetitive issues and resolve them faster than before.

- 100% more passenger feedback at airports and in airlines
- Real-time passenger feedback
- Resolve passenger issues faster than before
- Track staff and teller performance at airports and airlines
- Compare passenger satisfaction different touch-points
- Improve passenger delight and happiness

1.3 Scope

Airport Feedback App helps the staff in doing everything they can to make the journey less stressful. The real-time feedback system has expanded to allow passengers to rate lounges, staff, store, etc., in addition to the cleanliness of facilities. Prime focus on actionable insights has led to increased airport productivity. This has refined the service standards for better passenger experience.

Our project aims at Business process automation, i.e. we have tried to computerize various processes of Airport Feedback App

- In computer system, the person has to fill various forms & number of copies of the forms can be easily generated at a time.
- In computer system, it is not necessary to create the manifest but we can directly print it, which saves our time.
- To assist the staff in capturing the efforts spent on their respective working areas.
- To utilize resources in an efficient manner by increasing their productivity through automation.
- The system generates types of information that can be used for various purposes.
- It satisfies the user requirement.
- Be easy to understand by the passenger and the operator.
- Be easy to operate

- Have a good user interface
- Be expandable
- Delivered on scheduled within the budget.

1.4 Definitions, Acronyms and Abbreviations

- **Admin (Administrator)**

  He has the authority to add/delete users, grant permission to doctors and
- **UML (Unified Modeling Language)**

  It is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct and document
- **XML (Extensible Markup Language)**

  It is a text based format that let developers describe, deliver and exchange structured data between a range of applications to client for display and manipulation.
- **HTTP (Hypertext Transfer Protocol)**

  It's a service protocol.
- **JSON (JavaScript Object Notation)**

  It is a text format for storing and transporting data.
- **NPM (Node Package Manager)**

  It is a package manager for the JavaScript programming language.
- **API (Application Program Interface)**

  It's a way for two or more computer programs to communicate with each other.
- **JSX (JavaScript XML)**

  It's a syntax extension to JavaScript which allow us to write HTML in React.

1.5 Tools Used

1.5.1 Application architecture – Node.js, Express.js, React.js

- **Node.js**

  Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on a JavaScript Engine and executes JavaScript code outside a web browser, which was designed to build scalable network applications.

- **Express.js**

  Express.js, or simply Express, is a back end web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.

- **React.js**

  React is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta and a community of individual developers and companies.

1.5.2 Development tool – Visual Studio Code

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

1.5.3 Database platform – MongoDB

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License which is deemed non-free by several distributions.

1.5.4 Design tool - Creately

Creately is a SaaS visual collaboration tool with diagramming and design capabilities designed by Cinergix. Creately has two versions: an online cloud edition and a downloadable offline edition for desktop which is compatible with Windows, Mac and Linux.

1.5.5 Web server – Node Server

A Node.js server makes your app available to serve HTTP requests. It provides the interaction between users and your application. Creating and starting a server is easy with Node.js's built-in http module. Node. js runs on chrome v8 engine which converts JavasSript code into machine code, it is highly scalable, lightweight, fast, and data-intensive.

1.6 Technologies to be used

- MongoDB : Non-Relational Database Management System.
- Git, Github : Version Control System
- React.js : Web Development
- Node.js, Express.js : Web Development

1.7 Overview

Existing System:

- Feedback forms
- Registration for passengers

Drawbacks:

- Machine Learning is not included

Proposed System:

- Registration for passengers, admin
- Access of feedback data to admin

Our Plan:

- Registration for users
- Provide flight details to user
- Apply machine learning to generate meaningful trends and results.

Overall Description

2.1 Product Perspective

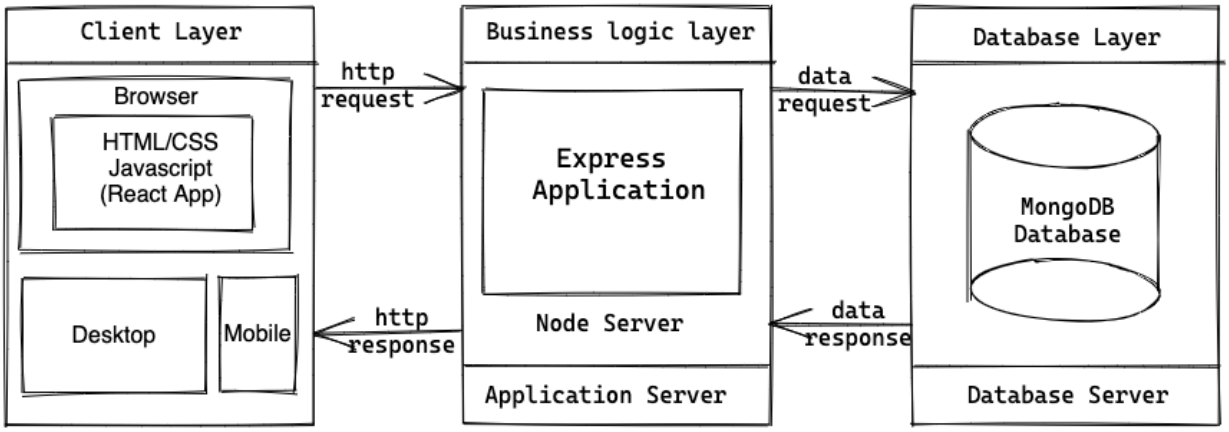

Fig 2.1: Product Perspective

2.2 Software Interface

- **Client on Internet**

  Web Browser, Operating System (any)

- **Client on Intranet**

  Web Browser, Operating System (any)

- **Web Server**

  WASCE, Operating System (any)

- **Data Base Server**

  MongoDB, Operating System (any)

- **Development End**

  Node.js, Express.js, React.js, JavaScript, OS(Windows)

2.3 Hardware Interface

Minimum Requirements:

Client Side			
	Processor	**RAM**	**Disk Space**
Google Chrome	Intel Pentium III or AMD - 800 MHz	128 MB	100 MB

Server Side			
	Processor	**RAM**	**Disk Space**
Node.js	Intel Pentium III or AMD - 800 MHz	1 GB	3.5 GB
MongoDB	Intel Pentium III or AMD - 800 MHz	256 MB	500 MB (Excluding Data Size)

Recommended Requirements:

Client Side			
	Processor	**RAM**	**Disk Space**
Google Chrome	All Intel or AMD - 1 GHZ	256 MB	100 MB

Server Side			
	Processor	**RAM**	**Disk Space**
Node.js	All Intel or AMD - 2 GHZ	2 GB	3.5 GB
MongoDB	All Intel or AMD - 2 GHZ	512 MB	500 MB (Excluding Data Size)

2.4 Communication Interface

- Client (passenger) on Internet will be using HTTP/HTTPS protocol.
- Client (system user) on Internet will be using HTTP/HTTPS protocol.

2.5 Constraints

- GUI is only in English.
- Login and password is used for the identification of admin.
- Only admins will be authorized to access or view the feedback data.
- Limited to HTTP/HTTPS.
- This system is working for single server.

2.6 E-R Diagram

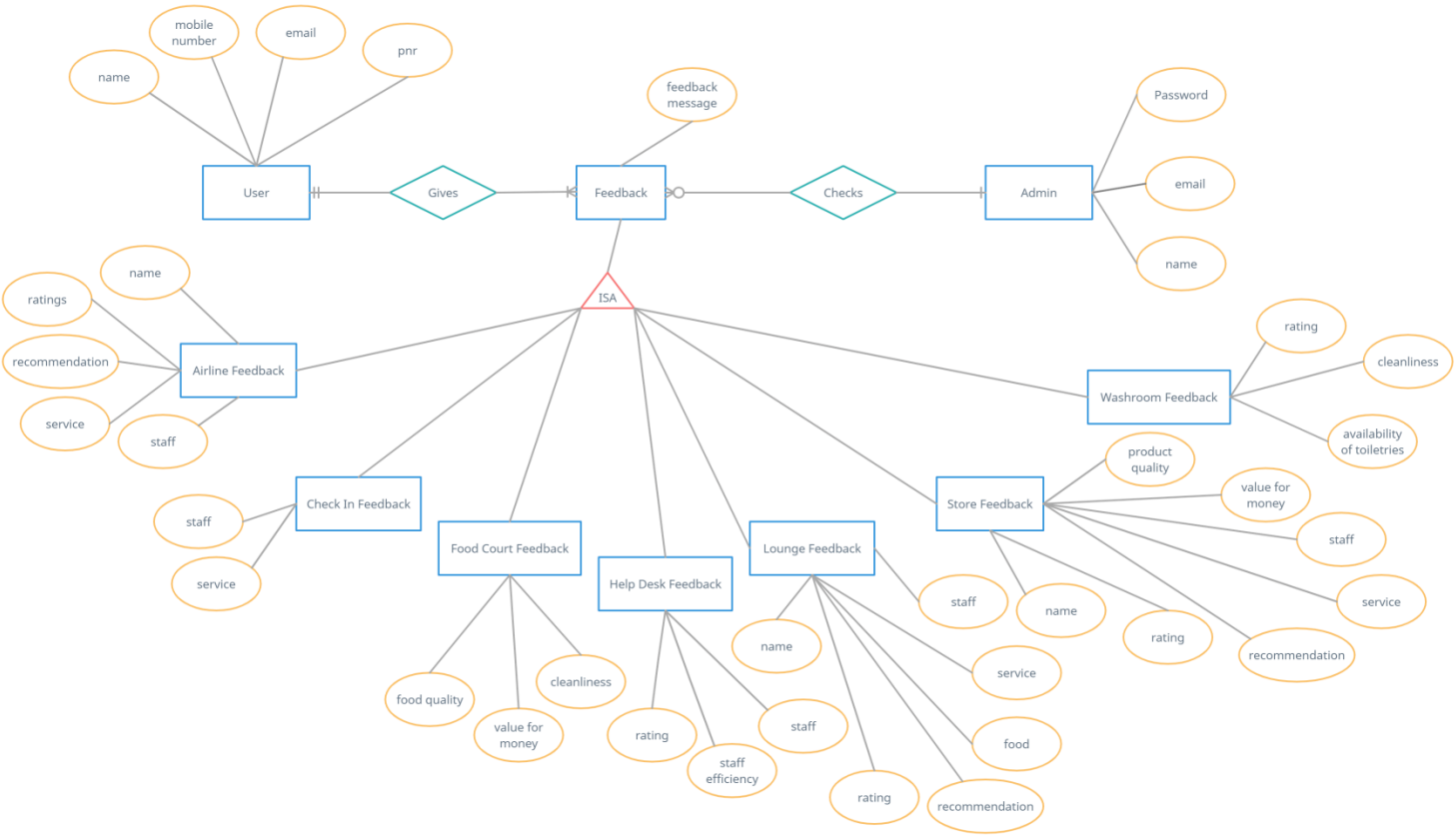

Fig2.1: E-R Diagram

2.7 Use Case Model Survey

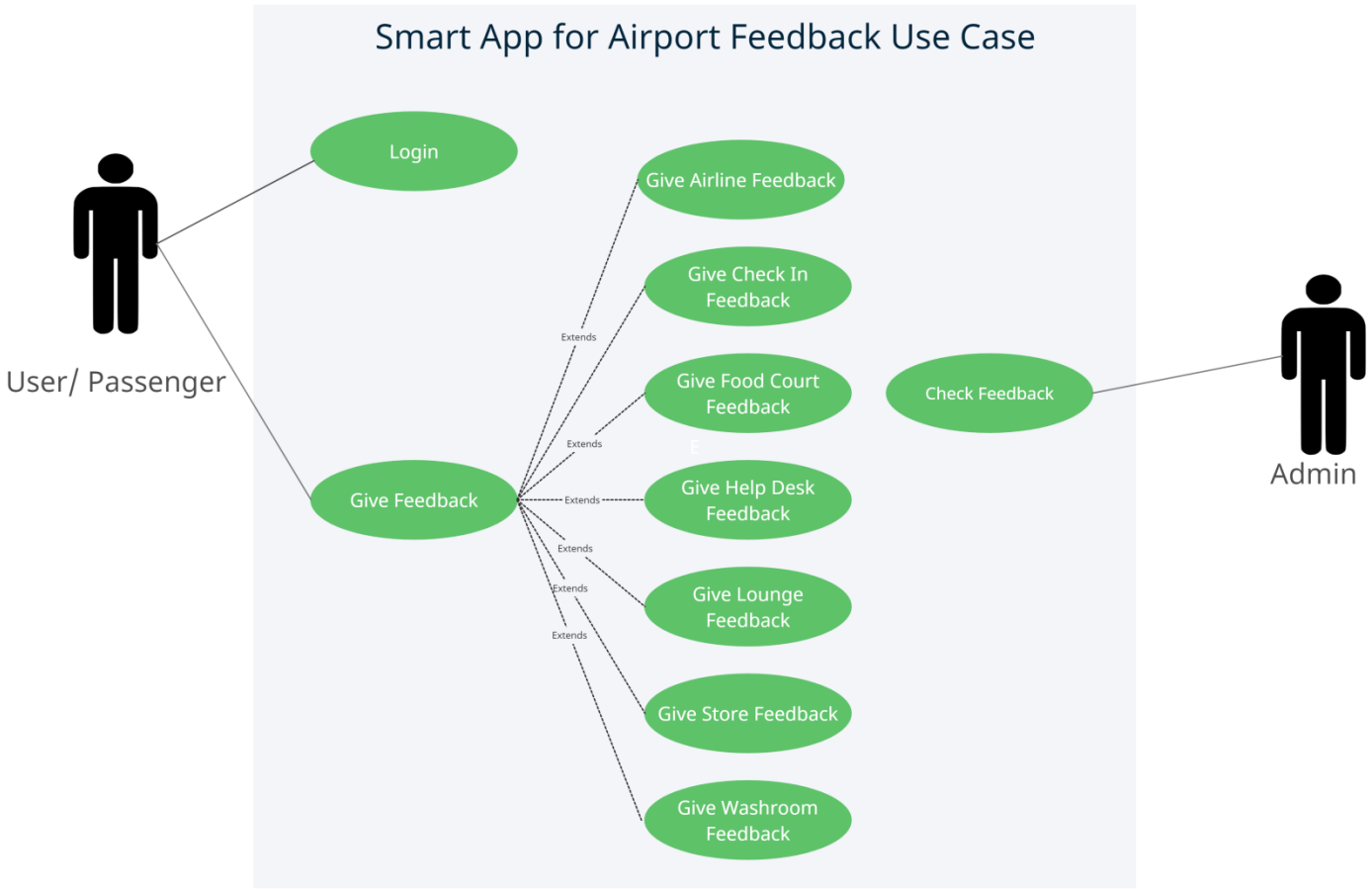

Fig2.2: Use Case Model Survey

- **Passenger:**

  Passenger can give the feedback regarding the different departments such as airlines, check-in counter, food court, help desk, lounge, store, etc.

- **Admin:**

  Admin has the authority to view or access the feedback provided by the passengers. Admin can also perform different types of operations on the feedback provided by the passengers to conclude different results

2.8 Architecture Diagram

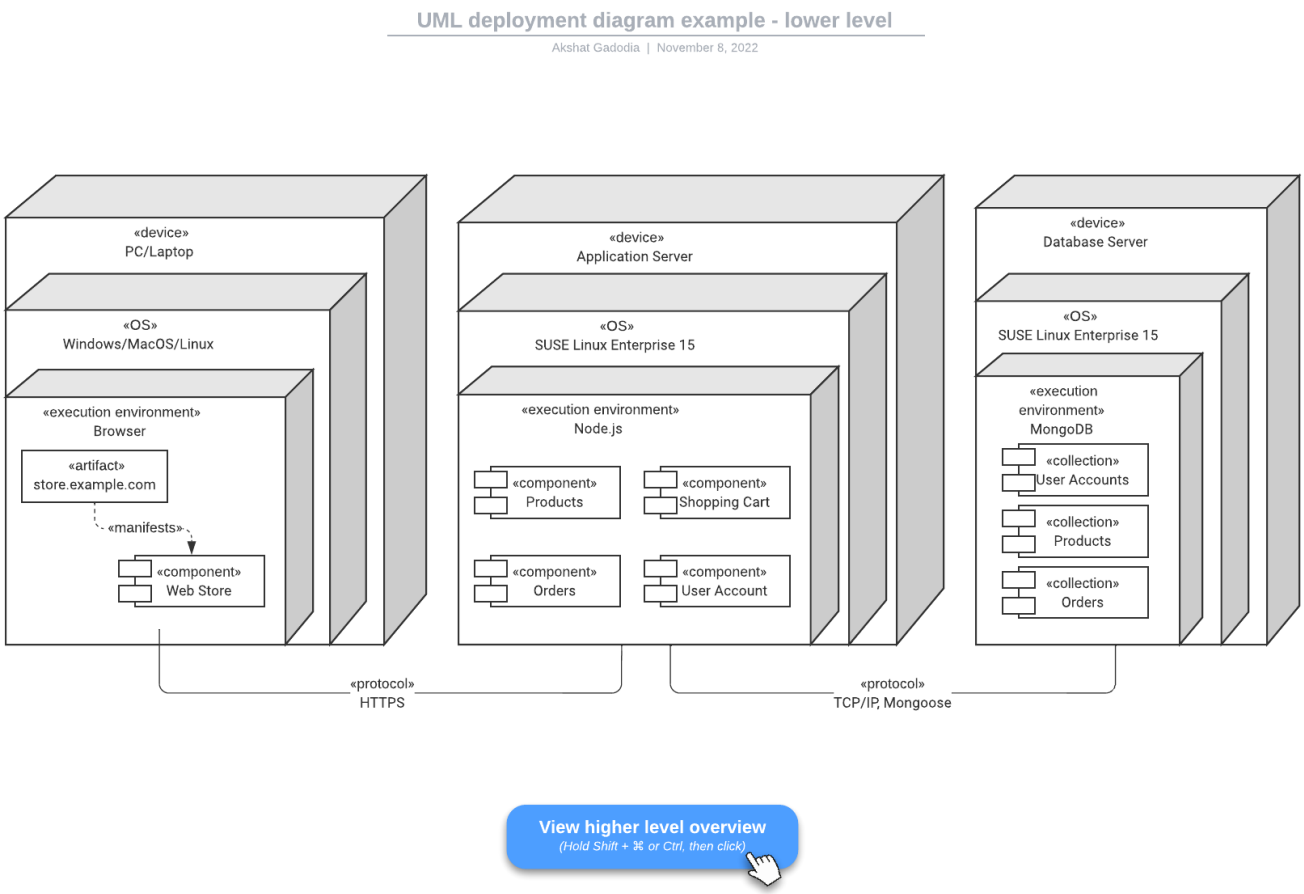

Fig2.3: Architecture Diagram

2.9 Database Design

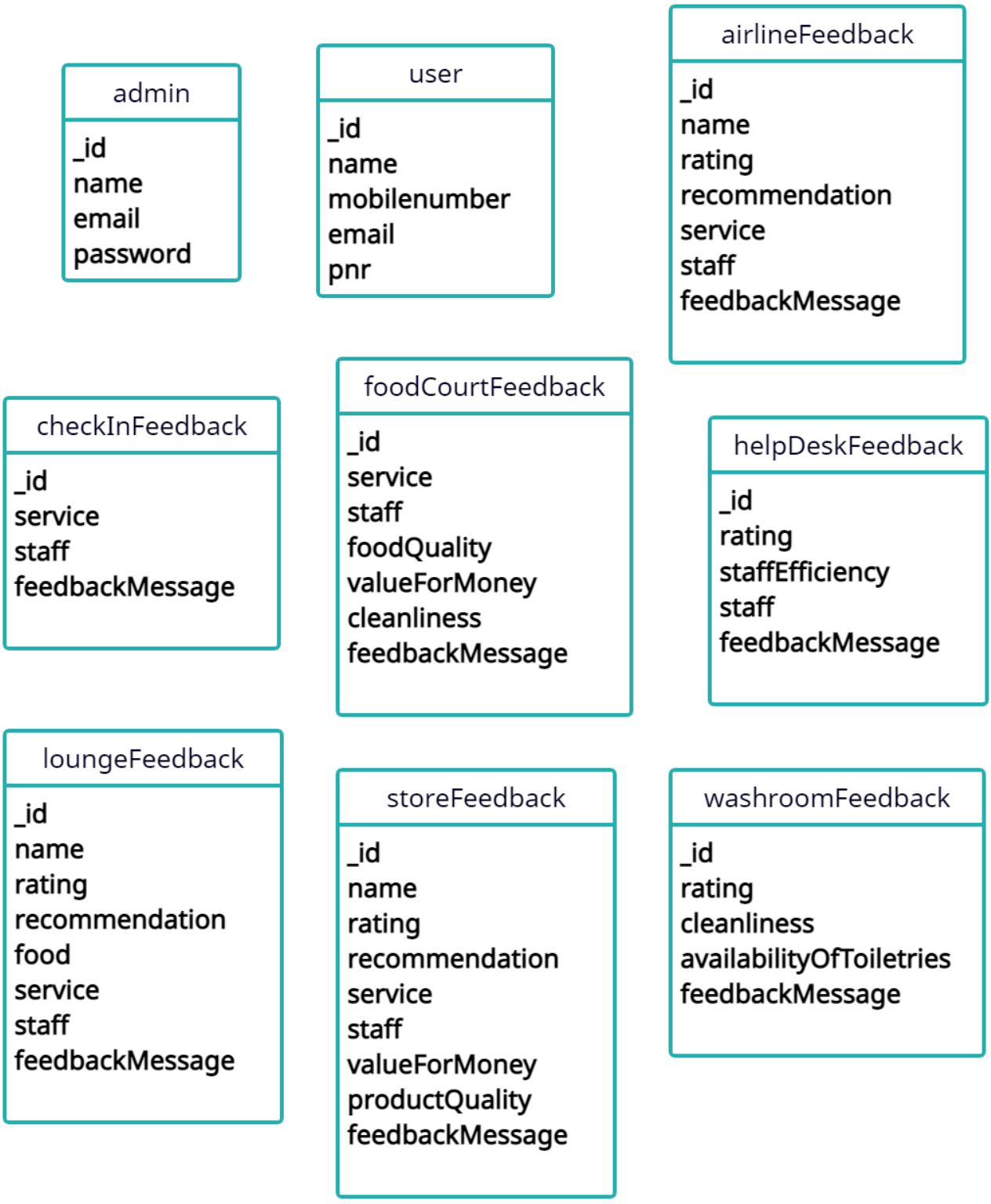

Fig2.4: Database Design

Specific Requirements

3.1 Use Case Reports

3.1.1 Passenger use-case report

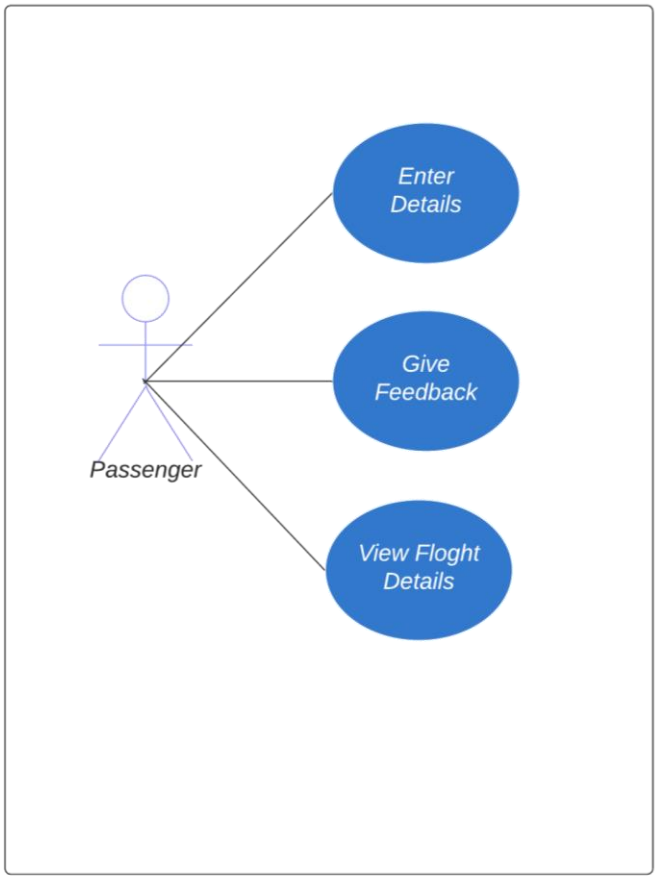

Fig3.1: Use case diagram for passenger

Use Case	Description
Enter Details	User needs to enter his personal details including PNR number.
Give Feedback	User can give the feedback to the desired department.
View Flight Detail	User can view any flight details using flight number.

3.1.2 Admin use-case report

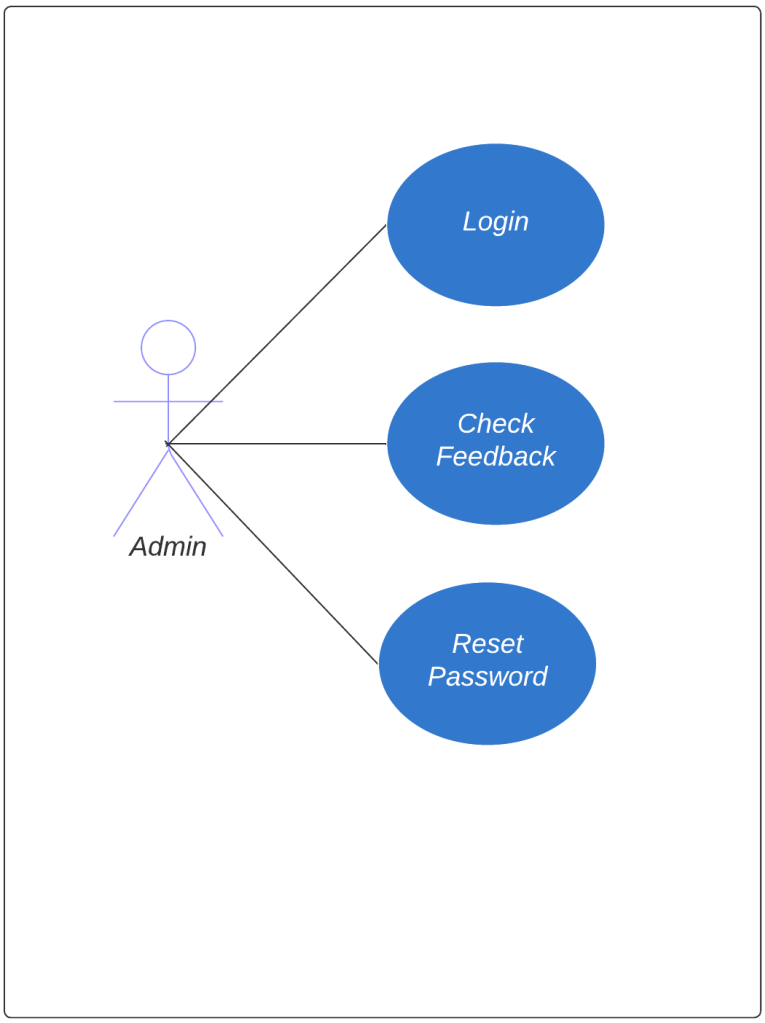

Fig3.2: Use case diagram for Admin

Use Case	Description
Login	Admin needs to login to access his account.
Check Feedback	Admin can check the feedbacks and can apply desired filters to generate required results.
Reset Password	Admin can reset his account password.

3.2 Activity Diagrams

3.2.1 Passenger Registration Activity

First of all the user enters his details. If all the details are correct he is redirected to home page. If any detail is incorrect then an error message is shown and user needs to correct the details to redirect to home page.

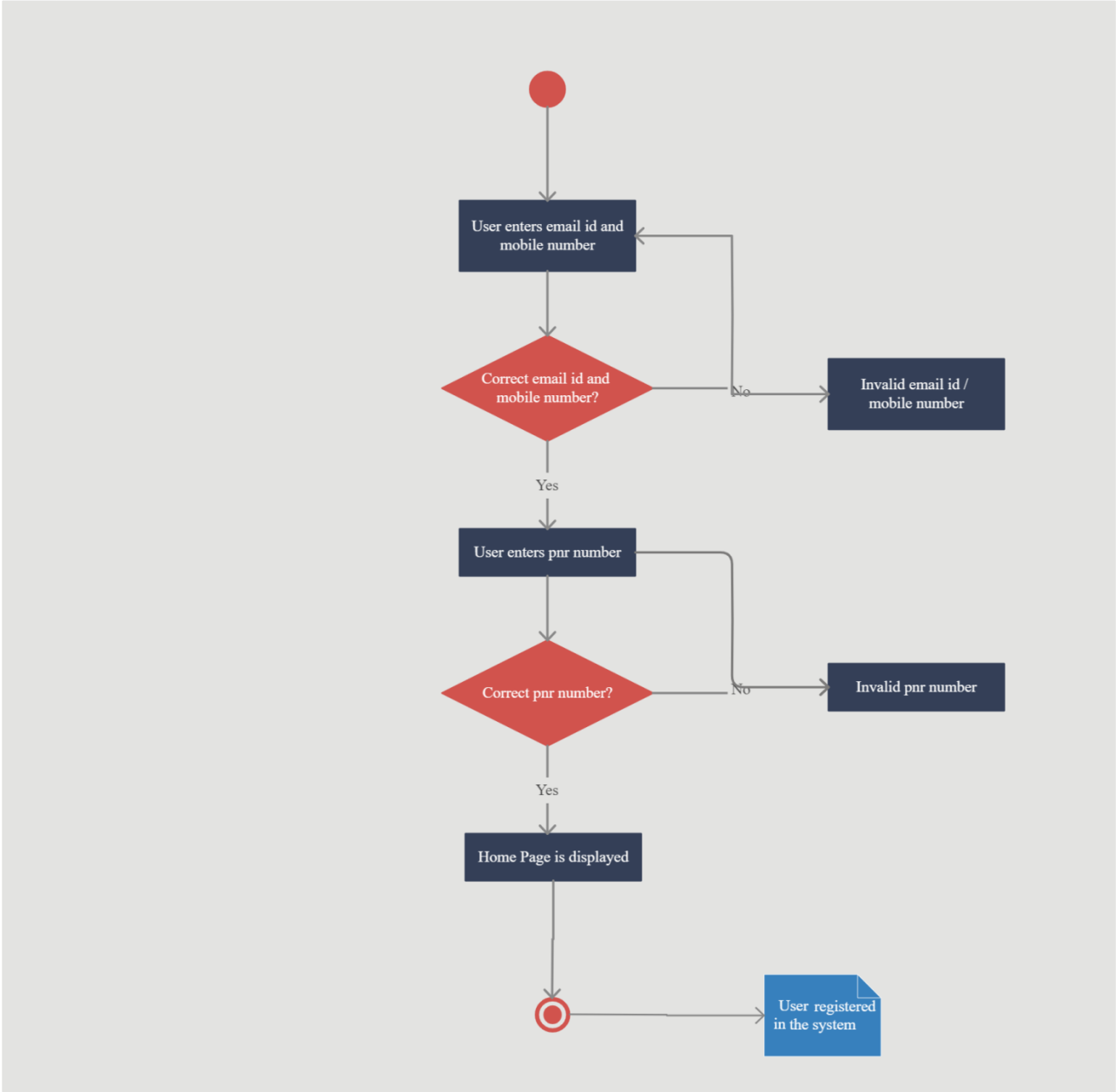

Fig 3.1: Activity Diagram Representing Passenger Registration

3.2.2 Passenger Feedback Activity

User selects the feedback he wants to give and enters all the feedback details. Entering all the feedback details is mandatory for the user to submit the feedback. If any detail is not enter by the user he will get an alert message indicating that he need to give the specified rating to proceed.

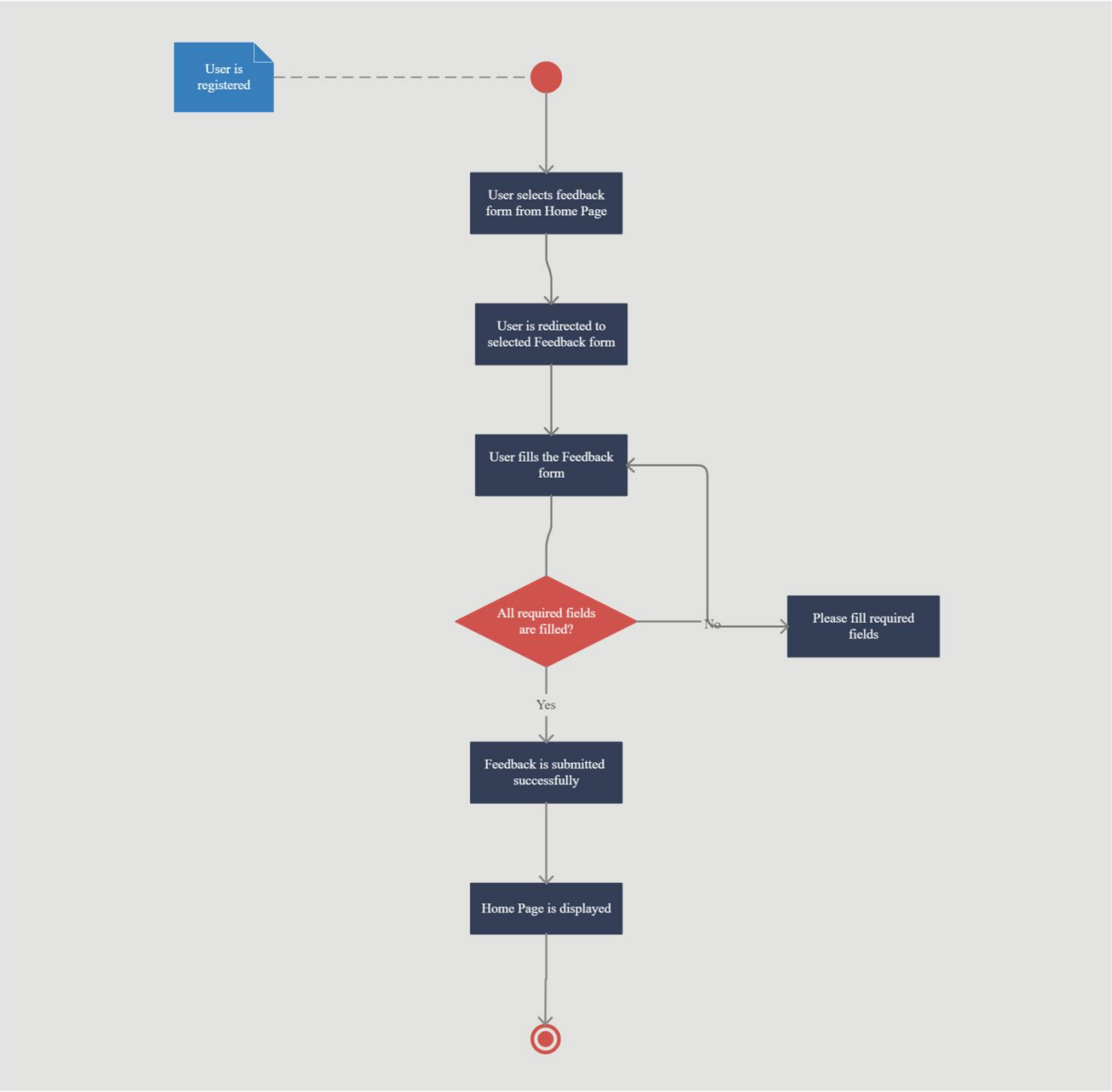

Fig 3.2: Activity Diagram Representing Passenger Feedback

3.2.3 Admin Login Activity

Admin need to login to check the feedbacks given by the users. If the entered credentials are correct then he will be redirected to the feedback page otherwise he would be shown an alert message to enter the correct details.

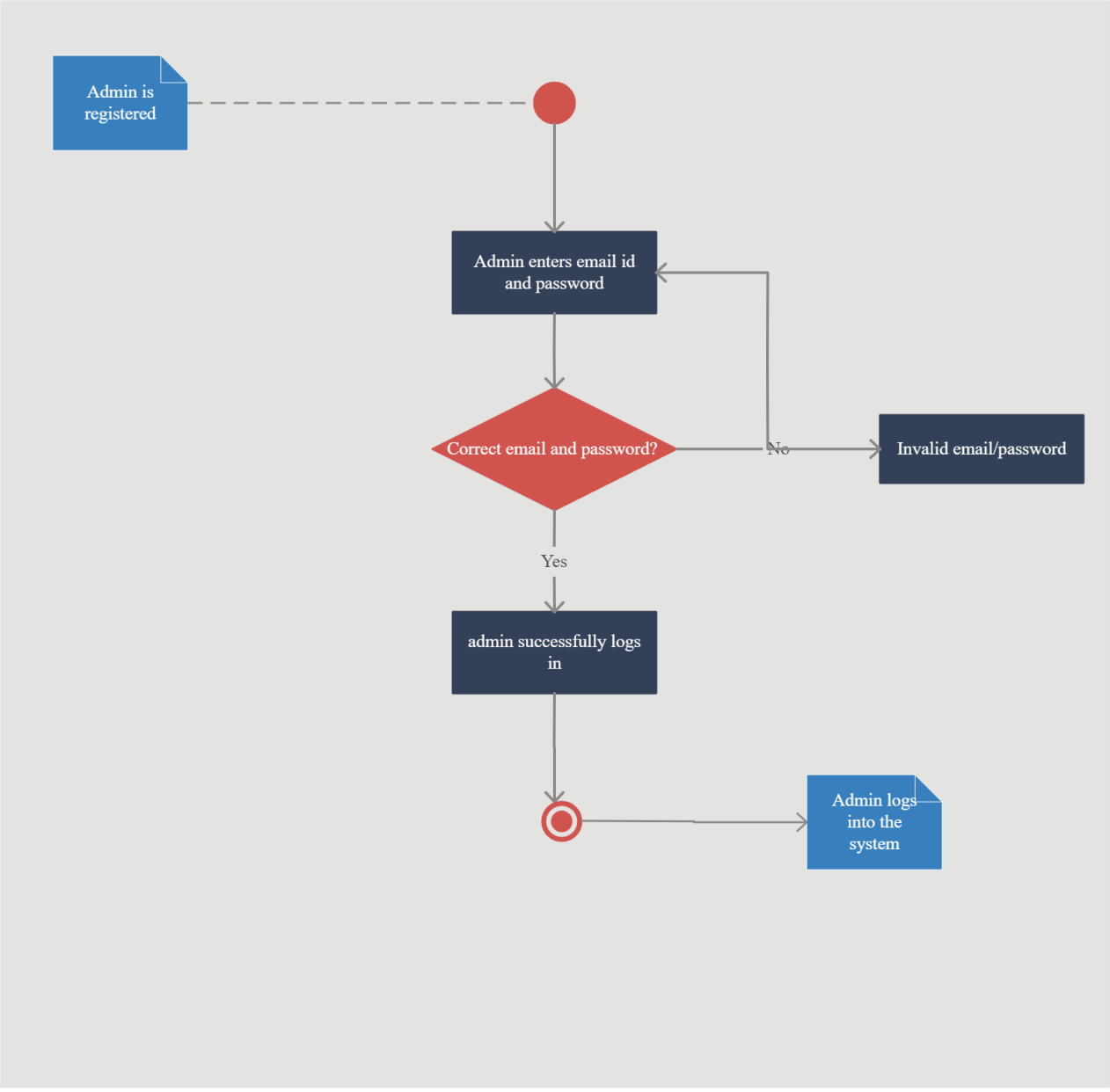

Fig 3.3: Activity Diagram Representing Admin Login

3.2.4 Admin View Feedback Activity

Admin can select the feedback he want to check at the dashboard and he can put desired filters to get the desired feedbacks.

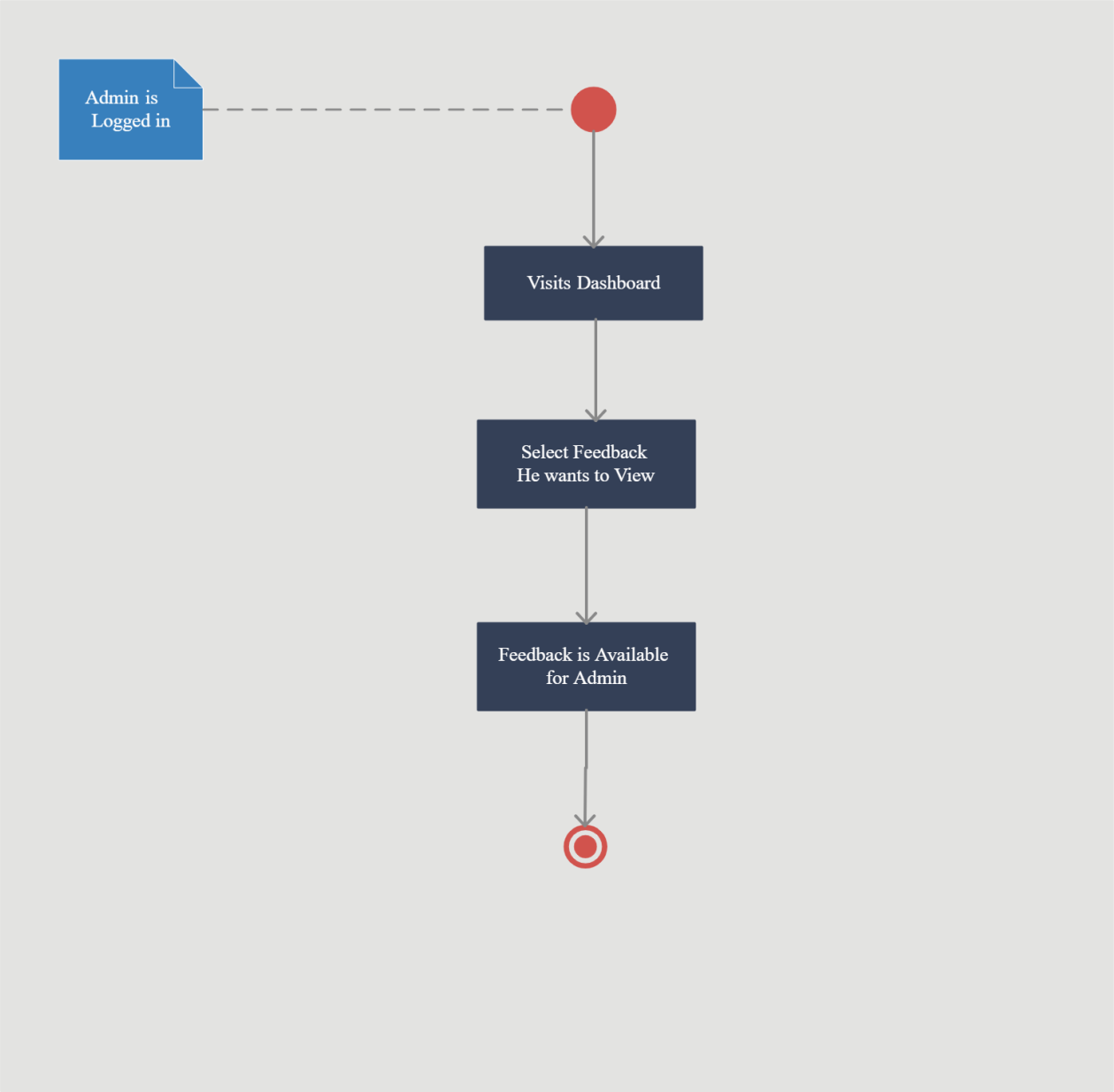

Fig 3.4: Activity Diagram Representing Admin View Feedback

3.2 Sequence Diagrams

3.3.1 Passenger Sequence Diagram

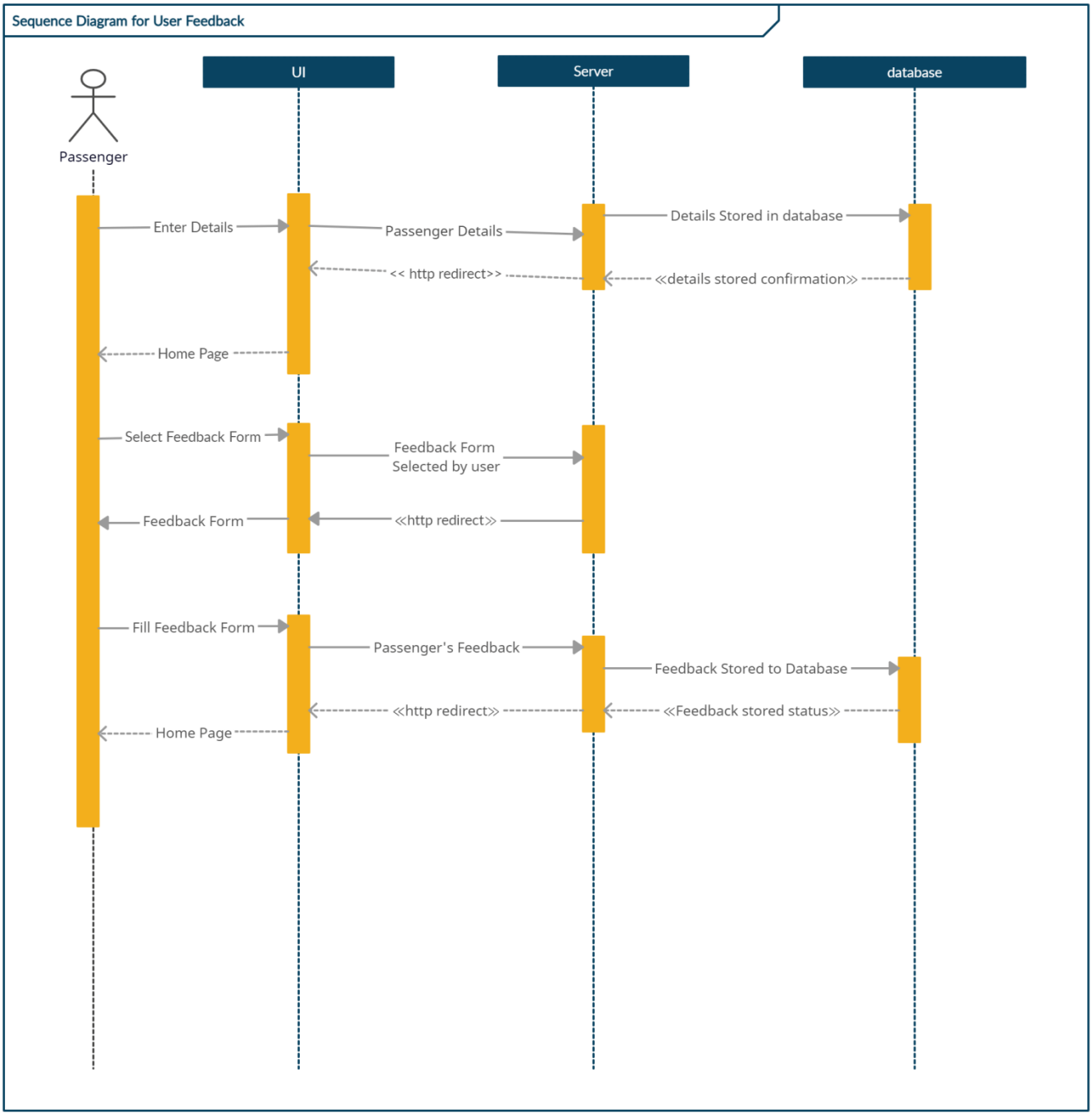

Fig 3.1: Sequence Diagram Representing Passenger

3.3.2 Admin View Feedback Sequence Diagram

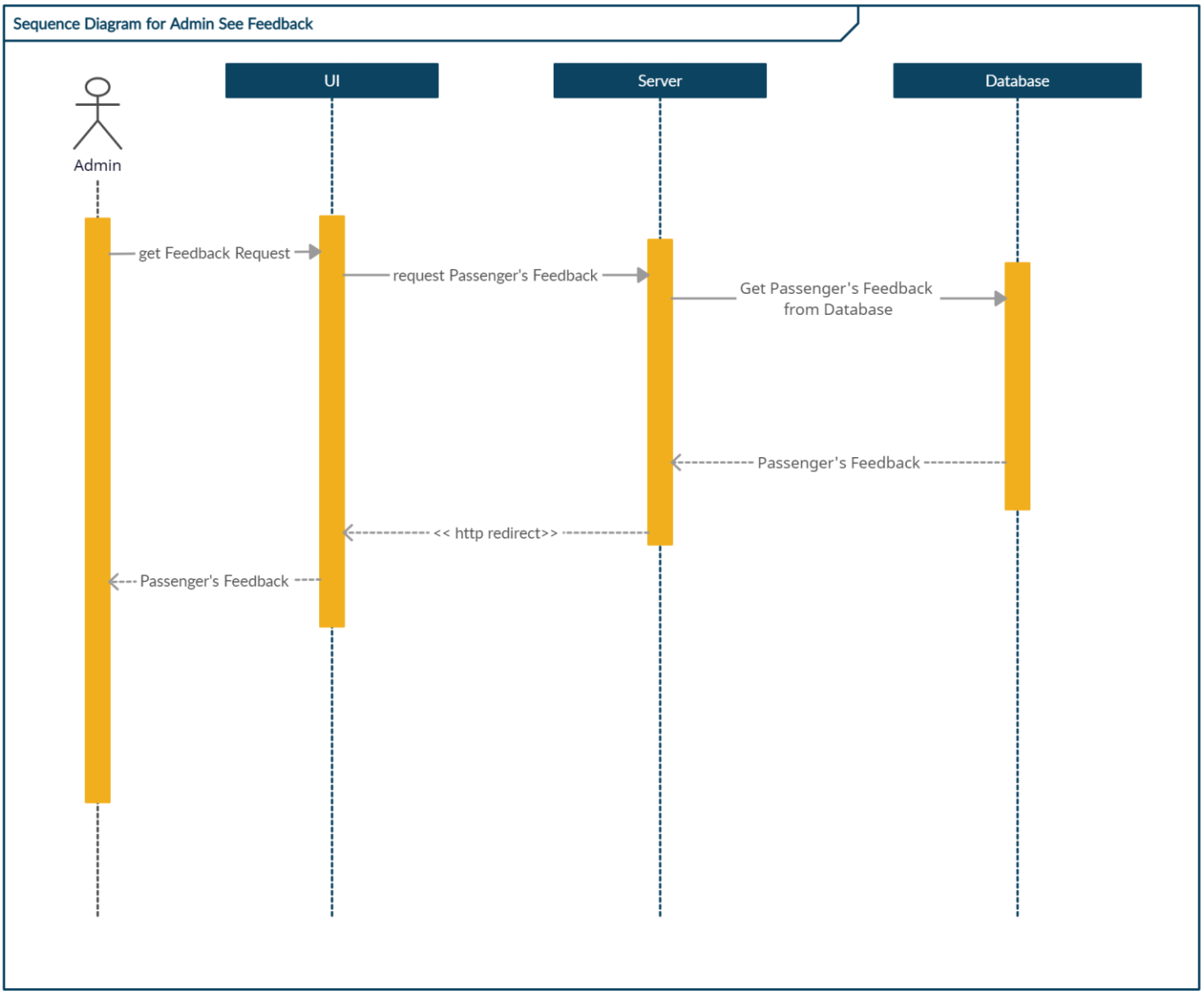

Fig3.2: Sequence Diagram Representing Admin View Feedback

3.3.3 Admin Login Sequence Diagram

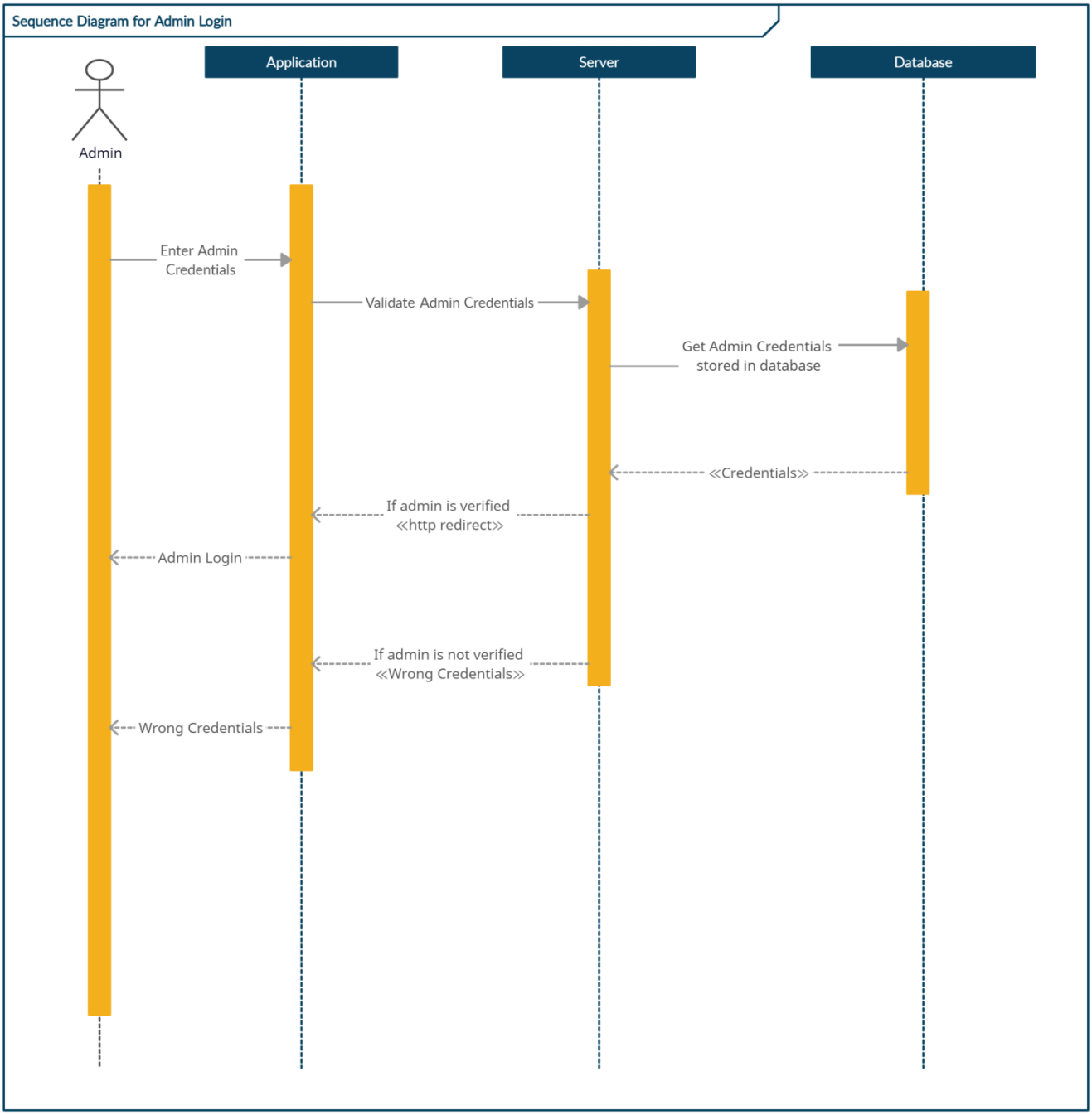

Fig3.3: Sequence Diagram Representing Admin Login

4. References

- Object Oriented Modeling and Design with UML-Michael Blaha, James Rambaugh.
- Software Engineering, Seventh Edition, Ian Sommerville.
- Wikipedia - *www.wikipedia.com*
- MongoDB - *www.mongodb.com*
- Node.js- *www.w3schools.com*
- Express.js -*www.tutorialspoint.com*
- React.js - *www.reactjs.org*