



First Deliverable

First Deliverable

1.Introduction

Our project aims to create a dynamic and user-friendly crowdfunding platform that empowers creators, entrepreneurs, and innovators to bring their ideas to life. Inspired by the success of Kickstarter, our app will provide a space where passionate individuals and creative projects can connect with a supportive community of backers.

With a focus on accessibility and innovation, our app will offer an intuitive user interface, robust project creation tools, and a secure payment system. Whether you're an aspiring artist, a tech visionary, or a community activist, our platform will enable you to raise the funds needed to turn your dreams into reality. We are excited to introduce a new, inclusive, and engaging crowdfunding experience, designed to bridge the gap between vision and achievement.

1.1 Project/Product Feasibility Report

There are many types of feasibilities:

- Technical
- Operational
- Economic
- Schedule
- Specification

- Information
- Motivational
- Legal and Ethical

1.1.1 Technical Feasibility

The technical feasibility of our crowdfunding app is a pivotal aspect in determining the project's success from a technological perspective. In this section, we examine several key factors that have been thoroughly considered to ensure the app's technical viability:

1.1.1.1 Infrastructure: Our crowdfunding app's functionality relies heavily on a robust technological infrastructure. Fortunately, we are in a favourable position, as the infrastructure for internet connectivity and mobile networks in our target market is well-established and widely accessible, providing a solid foundation for our app's operation.

1.1.1.2 Mobile Platforms: To maximise our platform's reach and impact, we are developing it for popular mobile platforms, including iOS and Android, in addition to a web-based version. The availability of mature and feature-rich development frameworks, tools, and resources for these platforms ensures the technical feasibility of our project.

1.1.1.3 Integration: Seamless integration with various payment gateways, social media platforms, and crowdfunding-specific APIs is essential for our app's success. We have conducted a comprehensive assessment of integration capabilities, established necessary partnerships, and secured agreements with data providers to ensure that our app can access real-time information vital for project creators and backers.

1.1.1.4 Scalability: Anticipating the dynamic growth of our user base, we have implemented a scalable technical architecture. Our infrastructure is designed to handle a substantial number of concurrent users, with provisions for server scaling and load balancing. This ensures that as our platform gains popularity and usage increases, our app will maintain high performance and reliability.

By carefully considering these technical aspects, we are confident in the technical feasibility of our crowdfunding app project and our ability to provide a seamless and reliable platform for both project creators and backers.

1.1.2 Operational Feasibility:

1.1.2.1 User Acceptance: Our app is designed with user-friendliness and intuitive navigation in mind. Extensive research has been conducted ensuring that the app caters to users with varying levels of technical expertise. This fosters usability and widespread acceptance.

1.1.2.2 Training and Support: We are committed to providing ample training and support mechanisms to assist users in effectively utilising the app. Documentation, tutorials, and responsive customer support channels will be made available to address any queries or issues that may arise.

1.1.2.3 Stakeholder Collaboration: Collaboration with key stakeholders, including project creators, backers, and payment partners, has been established to ensure a smooth operation and widespread adoption of the app. Regular communication and coordination will be maintained to address operational challenges effectively.

1.1.3 Economic Feasibility:

1.1.3.1 Cost-Benefit Analysis: A comprehensive cost-benefit analysis has been diligently conducted. This analysis encompasses development costs, infrastructure expenses, marketing investments, and projected revenue streams. The results indicate a positive return on investment (ROI) within a reasonable timeframe.

1.1.3.2 Revenue Generation: Our revenue model involves multiple channels, including advertisements, service provider partnerships, and premium features. Rigorous market research and financial forecasting have been performed to estimate revenue potential, further solidifying the economic feasibility of our project.

1.1.3.3 Market Potential: To gauge economic feasibility, we've undertaken extensive market research to assess the demand and potential user base for our crowdfunding app. The findings indicate a growing market for crowdfunding services in our target region, creating favourable conditions for economic viability.

1.1.4 Schedule Feasibility:

Our project is scheduled for completion in an estimated timeframe of 12 months. We are committed to diligent project management to ensure that we deliver ahead of schedule where possible. Detailed analysis and milestones are provided in the project's Gantt chart (section 1.5).

1.1.5 Specification Feasibility:

To address the clarity and definiteness of requirements, we have meticulously documented both functional and non-functional specifications. The scope boundaries have been clearly assessed to ensure that all necessary features and constraints are understood and met.

1.1.6 Information Feasibility:

Meaningful Information: We will leverage APIs and databases to access the relevant data needed for our crowdfunding app, ensuring that accurate and up-to-date information is available.

Completeness Information: As web applications have a rich history, our website will be designed to attract users through a user-friendly interface and the full range of functionalities that cater to their crowdfunding needs.

1.1.7 Motivational Feasibility:

Our motivation for creating this crowdfunding app is multifaceted. We aim to enhance convenience and efficiency for project creators and backers, improve accessibility, promote sustainable funding, gather valuable data for analysis and planning, and capitalise on the burgeoning crowdfunding market.

By bringing together project creators and backers on a single platform, we aspire to simplify the funding process, empower creators, and contribute to a more vibrant and sustainable crowdfunding ecosystem.

1.1.8 Legal & Ethical Feasibility:

We are fully committed to legal and ethical compliance in the development and operation of our app:

Compliance with Laws: Our app will adhere to all relevant laws and regulations governing crowdfunding, privacy, data protection, and intellectual property rights.

User Privacy and Data Protection: We prioritise user privacy and data protection by clearly communicating data collection and usage practices and obtaining user consent for data processing.

Fairness and Accessibility: Our app ensures fair access to crowdfunding opportunities for all users, regardless of demographics, and avoids any form of discrimination or bias in project recommendations or funding decisions.

Transparency and Accountability: We provide transparent information about our app's operations, algorithms, and decision-making processes. We establish mechanisms for addressing user concerns, handling complaints, and resolving disputes in a fair and accountable manner.

1.2 Project/Product Scope

The scope of our crowdfunding app project is of paramount importance in defining the boundaries of our endeavour and the objectives we aim to achieve. As the saying goes, "TIME IS MONEY," our primary goal is to create a web-based platform that simplifies the process of crowdfunding for both project creators and backers, ultimately enhancing the efficiency of the crowdfunding ecosystem.

Our project scope is determined by the set of requirements that are allocated to it. These requirements encompass the features, functionalities, and capabilities that our crowdfunding app will offer. Our primary focus is to facilitate the creation and funding of projects across diverse categories, allowing project creators to connect with potential backers easily.

Effective management of project scope is imperative for the success of our crowdfunding app. It entails aligning our project's goals with available resources, including time, human resources, and financial investments. To achieve this, we employ an iterative and incremental development approach, which enables us to break down the project scope into smaller, more manageable components.

In managing project scope, we rely on attributes such as priority, effort, and risk as the basis for negotiating the inclusion of specific features and functionalities. This approach allows us to focus on the essential

characteristics of each requirement, facilitating smoother and more productive negotiations.

1.3 Project/Product Costing

In our crowdfunding app project, metrics and cost estimation hold significant sway. We utilise metrics to gain insights into project progress and enable data-driven decision-making while managing project costs efficiently.

Within our project context, metrics fall into two essential categories:

1. Knowledge-Oriented Metrics: These metrics guide monitoring, evaluation, prediction, and process enhancement. They empower us to streamline our workflow and make informed decisions.
2. Achievement-Oriented Metrics: These metrics assess specific product aspects of our crowdfunding app, offering a comprehensive view of its quality and functionality, central to a superior user experience.

Cost estimation in our project aligns with algorithmic cost modelling.

Mathematical formulas connect resource inputs to metrics, delivering precise cost estimates rooted in historical data analysis.

1.3.1 Project Cost Estimation by Function Point Analysis

In our crowdfunding app project, we employ Function Point Analysis (FPA) as a pivotal technique for cost estimation. FPA uses a measure of the functionality delivered by the application to normalise cost estimation. Since the measurement of 'functionality' cannot be made directly, it is derived indirectly using other measurable factors.

Complexity Weighting Factors:

To calculate Function Points (FP), we consider a set of complexity weighting factors, each assigned a value on a scale from 0 to 5. These factors are instrumental in quantifying the intricacies and demands of the project:

Number	Complexity Weighting Factors	Value (0-5)
1	User Account Management	4
2	Payment Processing	5

3	Project Creation	4
4	User Interaction	3
5	Data Security	5
6	Search Functionality	4
7	Reporting and Analytics	3
8	External Integrations	4
9	Admin Panel Functionality	3
10	Notifications	2
Total = $\sum F_j$		38

Number of User Inputs: Each user input that provides distinct application-oriented data to the software is counted. It's important to note that user inputs are distinguished from inquiries, which are counted separately.

Number of User Outputs: Each user output that provides application-oriented information to the user is counted. In this context, outputs include reports, screens, error messages, and other relevant components. Individual data items within a report are not counted separately.

Number of User Inquiries: An inquiry is defined as an on-line input that results in the immediate generation of software responses in the form of on-line outputs. Each distinct inquiry is counted.

Number of Files: Each logical master file, which is a logical grouping of data that may be part of a larger database or a separate file, is counted.

Number of External Interfaces: We count all machine-readable interfaces, such as data files on storage media, used to transmit information to another system.

Counting Function Points:

To calculate Function Points (FP), we consider several categories and their counts:

Measurement Parameters	Count	Low	Average	High
Number of User Inputs	30	3x3	20x4	7x6
Number of User Outputs	20	2x4	14x5	4x7
Number of Inquiries	10	0x3	2x4	4x6
Number of Files	2	0x7	0x10	1x15
Number of Interfaces	3	0x5	0x7	1x10

The total Function Points (FP est.) are calculated as:

FP est. = Count Total * [0.65 + 0.01 * (CF)]

FP est. = 65 * [0.65 + 0.01 * (38)]

FP est. = 65 * 1.03

FP est. ≈ 66.95

Total Project Cost and Effort:

Finally, the total project cost and effort are calculated, considering the average productivity parameter for the system. The formulae are given as follows:

Cost / FP = Labor Rate / Productivity Parameter

Total Project Cost = FP est. * (Cost / FP)

Total Estimated Effort = FP est. / Productivity Parameter

1.3.2 Project Cost Estimation by Using COCOMO 81 (Constructive Cost Model)

In our crowdfunding app project, we employ Boehm's COCOMO 81 (Constructive Cost Model) for project cost estimation. COCOMO is a well-established model for estimating the cost, effort, and schedule when planning a new software development activity. It's based on software development practices prevalent in the 1970s through the 1980s and is available in three distinct levels, each offering varying degrees of detail and accuracy in project planning and design.

Three COCOMO Levels:

Basic: The Basic level is utilised for initial, rough estimates. It provides a quick overview of the project's cost, effort, and schedule based on the lines of code (KLOC) and duration (TD).

Intermediate: The Intermediate level, the most commonly used version, offers more detailed insights. It incorporates 15 different factors, known as cost drivers, to account for various project attributes, including personnel capability, tool usage, hardware constraints, and other relevant considerations.

Detailed: At the Detailed level, COCOMO delves into the influence of different factors on specific project phases, such as design, coding/testing, and integration/testing. While it provides a high level of granularity, Detailed COCOMO is not frequently employed.

Software Development Types:

COCOMO takes into account three distinct software development types:

Organic: This type is applicable when relatively small software teams develop familiar types of software in an in-house environment. Most team members possess experience working with related systems.

Embedded: The Embedded type is suitable when the project may require new technology, unfamiliar algorithms, or innovative new methods, making it a more challenging development endeavour.

Semi-detached: Semi-detached is an intermediate stage, lying between the Organic and Embedded types, combining elements of both.

Formulas for Effort and Schedule:

The following formulas are used for estimating effort (person-months, PM) and schedule (duration, TD) for each software development type at the Basic and Intermediate levels:

Basic COCOMO:

Organic: $PM = 2.4 (KLOC)^{1.05}$, $TD = 2.5(PM)^{0.38}$

Semi-Detached: $PM = 3.0 (KLOC)^{1.12}$, $TD = 2.5(PM)^{0.35}$

Embedded: $PM = 2.4 (KLOC)^{1.20}$, $TD = 2.5(PM)^{0.32}$

PM= person-month (effort)

KLOC= lines of code, in thousands

TD= number of months estimated for software development (duration)

Intermediate COCOMO:

Organic: $PM = 2.4 (KLOC)^{1.05} \times M$

Semi-Detached: $PM = 3.0 (KLOC)^{1.12} \times M$

Embedded: $PM = 2.4 (KLOC)^{1.20} \times M$

PM= person-month

KLOC= lines of code, in thousands

M.- reflects 15 predictor variables, called cost drivers

The schedule is determined using the Basic COCOMO schedule equations.

People Required: The number of people required for the project can be calculated as $\text{People Required} = \text{Effort} / \text{Duration}$.

1.4 CPM - Critical Path Method

CPM provides the following benefits:

- Provides a graphical view of the project.
- Predicts the time required to complete the project.
- Shows which activities are critical to maintaining the schedule and which are not.

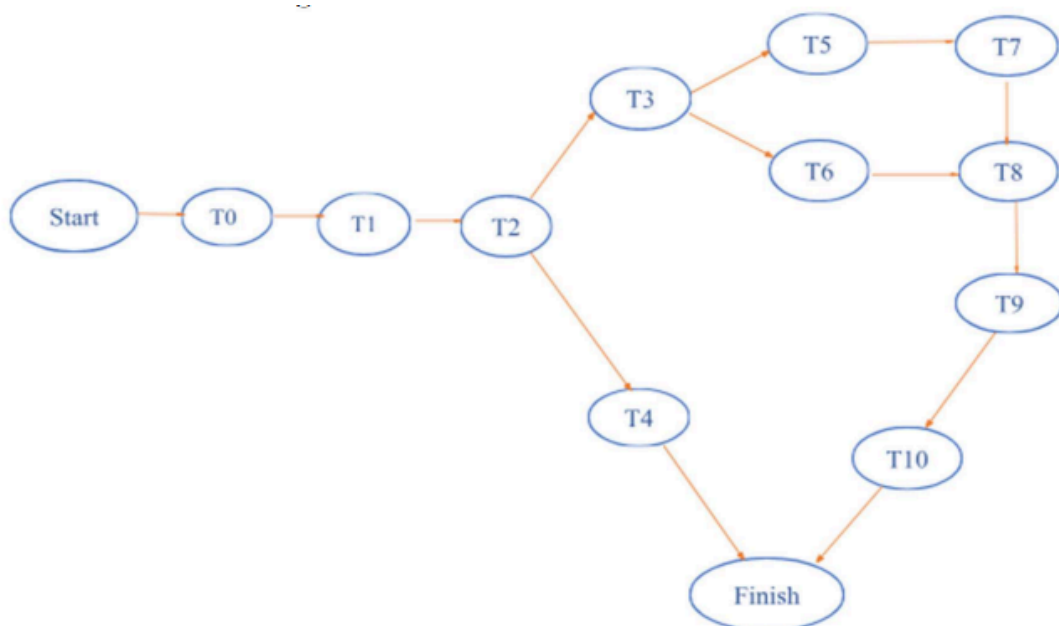
1. Specify the Individual Activities

Activity	Naming Convention
Proposal	T0
Requirement Gathering	T1
Requirement Analysis	T2
Modelling	T3
Identifying Stakeholders	T4
Designing	T5
GUI Designing	T6
Designing Prototypes	T7
Implementation	T8
Testing	T9
Deployment	T10

2. Determine the Sequence of the Activities

T0	none	3
Activity	Immediate Predecessor	Duration (Days)
T1	T0	7
T2	T1	3
T3	T2	7
T4	T2	10
T5	T3	7
T6	T3,T5	14
T7	T5	7
T8	T6,T7	50
T9	T8	15
T10	T9	3

3. Draw the Network Diagram



4. Estimate Activity Completion Time

The time required to complete each activity can be estimated using past experience or the estimates of knowledgeable persons. CPM is a deterministic model that does not take into account variation in the completion time, so only one number is used for an activity's time estimate.

5. Identify the Critical Path

Determining the following six parameters for each activity which can identify the critical path:

ES: earliest start time: the earliest time at which the activity can start given that its precedent activities must be completed first.

$$ES(K) = \max [EF(J) : J \text{ is an immediate predecessor of } K]$$

EF: earliest finish time: equal to the earliest start time for the activity plus the time required to complete the activity.

$$EF(K) = ES(K) + Dur(K)$$

LF: latest finish time: the latest time at which the activity can be completed without delaying the project.

$$LF(K) = \min [LS(J) : J \text{ is a successor of } K]$$

LS: latest start time: equal to the latest finish time minus the time required to complete the activity.

$$LS(K) = LF(K) - Dur(K)$$

TS: Total Slack: the time that the completion of an activity can be delayed without delaying the end of the project

$$TS(K) = LS(K) - ES(K)$$

FS: Free Slack: the time that an activity can be delayed without delaying both the start of any succeeding activity and the end of the project.

$$FS(K) = \min [ES(J) : J \text{ is successor of } K] - EF(K)$$

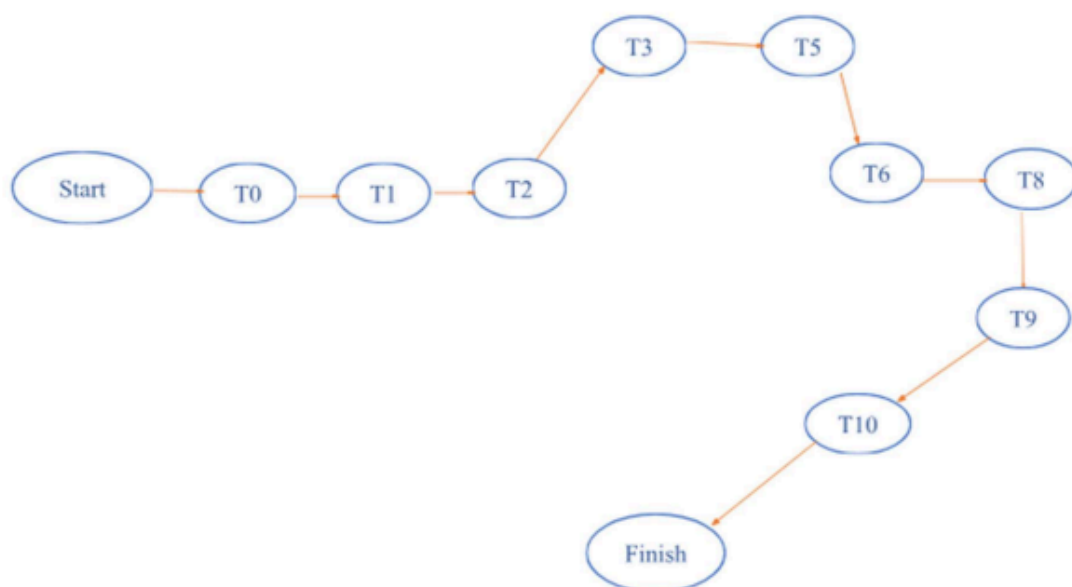
The slack time for an activity is the time between its earliest and latest start time, or between its earliest and latest finish time. Slack is the amount of time that an activity can be delayed past its earliest start or earliest finish without delaying the project.

The critical path is the path through the project network in which none of the activities have slack, that is, the path for which $ES=LS$ and $EF=LF$ for all

activities in the path. A delay in the critical path delays the project. Similarly, to accelerate the project it is necessary to reduce the total time required for the activities in the critical path.

Activity	Duration	ES	EF	LS	LF	Slack
T0	3	0	3	0	3	0
T1	7	3	10	3	10	0
T2	3	10	13	10	13	0
T3	7	13	20	13	20	0
T4	10	13	23	99	109	86
T5	7	20	27	20	27	0
T6	14	27	41	27	41	0
T7	7	27	34	34	41	7
T8	50	40	91	41	91	0
T9	15	91	106	91	106	0
T10	3	106	109	106	109	0

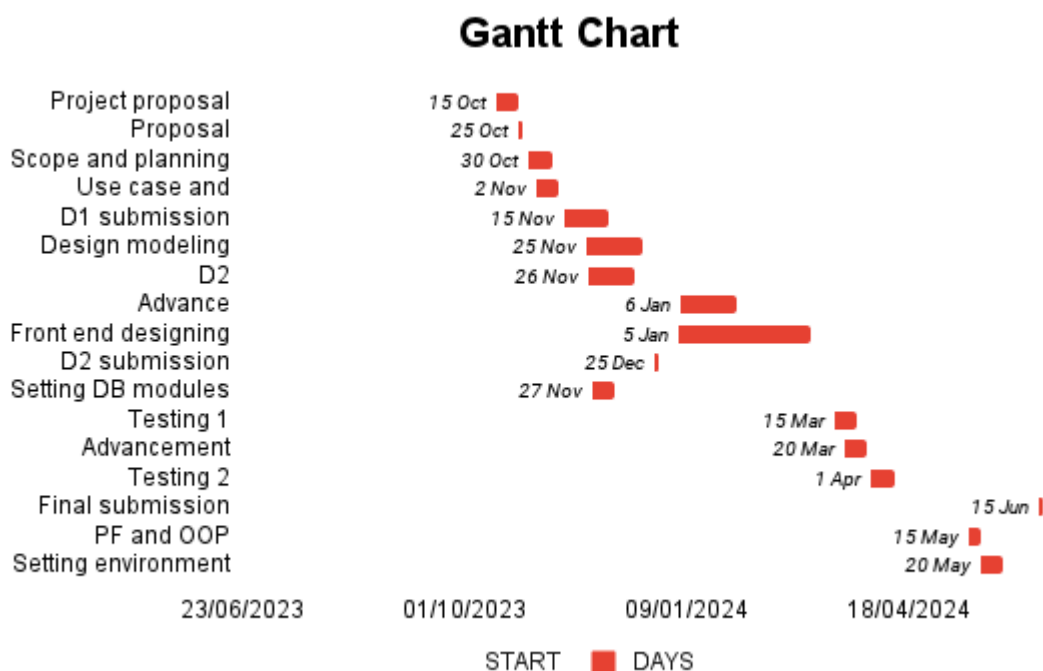
6. Update CPM Diagram



1.5 Gantt chart

The Gantt chart enumerates the activities to be performed on the vertical axis and their corresponding duration on the horizontal axis. In the early start approach, each activity is initiated as early as possible without violating the precedence relations. In the late start approach, each activity is delayed as much as possible as long as the earliest finish time of the project is not compromised.

Based on the Work Breakdown Structure (WBS), a timeline or Gantt chart showing the allocation of time to the project phases or iterations should be developed. This Gantt chart would identify major milestones with their achievement criteria. **Use Google sheet to develop gantt chart.**



1.6 Introduction to Team member and their skill set

Roll No.	Names	Skills
BCSF20A501	Zain Amir	<ul style="list-style-type: none"> • Coding • Logic Building • Deployment • Database Management • API's Configuration

BCSF20A506	Ali Asad	<ul style="list-style-type: none"> • Coding • Logic Building • Deployment • Database Management • API's Configuration
BCSF20A508	Haris Bin Athar	<ul style="list-style-type: none"> • Coding • Logic Building • Deployment • Database Management • API's Configuration
BCSF20A511	Hamna Suri	<ul style="list-style-type: none"> • Coding • Logic Building • Deployment • Database Management • API's Configuration
BCSF20A540	Sadia Rahman	<ul style="list-style-type: none"> • Coding • Logic Building • Deployment • Database Management • API's Configuration

1.7 Tools and Technology with reasoning

- Development Platform: React Framework
- Target Platform: Windows
- Programming Language: React JS, React Native

Working Tools:

Tools	Usage	Work level
Visual Studio Code	IDE	Backend
ReactJS	Front-End Interface	Front End interaction
Google sheet	WBS	Gantt Chart Maker
Google Docs	Text Editor	Documentation
Mongo Atlas	DB Editor	Database

React, Tailwind	Front End	Front End Outputs
MongoDB Atlas	DB storages	Database
GitHub	Sharing of code	Collaboration

1.8 Vision Document

A vision document for our crowdfunding app project outlines the overarching goals, objectives, and aspirations for the app. This document serves as a guiding compass that captures the desired outcomes and the overall vision of our project. Below are key elements to be included in this comprehensive vision document:

1. Introduction:

Provide an overview of our crowdfunding app and its purpose.

Explain the need for a platform that connects passionate project creators with enthusiastic backers and the benefits it can bring to both individuals and the creative community.

2. Vision Statement:

Clearly articulate the vision for our crowdfunding app. For example: "To create a cutting-edge crowdfunding platform that empowers creators to bring their ideas to life while connecting backers with meaningful and innovative projects."

3. Objectives:

Identify the key objectives that our app aims to achieve, which may include:

- Facilitating a seamless and user-friendly crowdfunding experience.
- Fostering creativity and innovation by supporting a wide range of project categories.
- Empowering project creators to access resources and funding for their ventures.
- Offering backers a diverse and rewarding selection of projects to support.

4. Target Users:

Define the target user segments for the app. These could include project creators, backers, entrepreneurs, artists, innovators, and individuals passionate about supporting creative projects.

5. Key Features and Functionality:

Outline the core features and functionalities that the app will offer, such as:

- Project creation and management tools for creators.
- User-friendly and secure payment options for backers.
- Project discovery and recommendation algorithms.
- Backer rewards and engagement mechanisms.
- A transparent and trust-based rating and review system.
- Real-time project updates and communication channels.

6. Technology and Infrastructure:

Discuss the technology stack and infrastructure required for the development and maintenance of the app, including considerations for data security, scalability, and efficient payment processing.

7. Innovation and Impact:

Highlight our commitment to fostering innovation and the app's potential impact on creators, backers, and the broader creative community. This could include driving entrepreneurship, supporting creative expression, and promoting economic growth.

8. Legal and Ethical Considerations:

Address the legal and ethical considerations that the app will adhere to, including user data privacy, terms of service, intellectual property rights, and ethical project representation.

9. Stakeholders and Partnerships:

Identify the key stakeholders, including project creators, backers, regulatory bodies, and potential partners. Discuss the importance of collaboration and partnerships to achieve the app's goals.

10. Future Roadmap:

Provide a glimpse into the future plans for the app, such as potential enhancements, expansion into new markets, and integration with emerging technologies to further empower project creators and backers.

1.9 Risk List

Here's a list of potential risks associated with this app:

1. Technical Risks:

- **Integration Challenges:** The difficulties in seamlessly connecting the platform with various payment gateways, project management tools, and third-party services.
- **Performance Issues:** The problems with the app's speed or responsiveness, and scalability when handling a large volume of user interactions, project updates, and payment transactions.
- **Data Security:** Risks related to data breaches, unauthorized access, or vulnerabilities. A data breach can lead to legal and financial consequences.

2. Regulatory and Legal Risks:

- **Compliance:** Failure to adhere to local, regional, or national regulations governing financial transactions, privacy and intellectual property rights.
- **Liability:** Legal challenges arising from disputes, accidents, or incidents related to crowdfunding projects hosted on the platform.
- **Privacy Concerns:** Privacy risks associated with collecting, storing, and processing user and project data. Failing to follow data protection regulations or secure user data properly, potentially leading to privacy breaches or non-compliance with data protection regulations.

3. Market Risks:

- **Competitive Landscape:** This risk involves the presence of already established crowdfunding platforms and other fundraising methods that compete for both backers and creators.
- **Market Adoption:** The risk related to the adoption of this app by potential users and creators. Factors such as user demand, marketing strategies, and market trends will influence how quickly the platform gains traction.
- **Changing Consumer Behaviour:** Market risks can be influenced by shifts in consumer behaviour and preferences from time to time.

4. Financial Risks:

- **Funding Challenges:** The challenge to secure the initial capital required for development and ongoing operational expenses. Insufficient funding can lead to project delays or the inability to address critical platform needs.
- **Revenue Model Uncertainty:** The uncertainties regarding the platform's revenue model, such as fees charged to creators or backers.
- **Economic Downturn:** Economic downturns can affect the platform's

financial health by reducing user spending and available investment capital, which can impact both the number of projects and the willingness of backers to contribute.

5. Operational Risks:

- **Service Disruptions:** Technical issues, server downtime, or network outages that affect the availability and reliability of the app.
- **Supply Chain Interruptions:** The risks may involve disruptions in the delivery of services or resources that are critical for the platform's operations, leading to delays and affecting the overall user experience.
- **Resource Management:** The inefficient resource management, including human resources and technology. It may result in inefficiencies, increased costs, and delays in project management.

6. User Risks:

- **User Data Protection:** Failure to adequately secure user data. Risks related to data breaches, unauthorised access, regulatory issues, and reputational damage.
- **User Trust and Reputation:** The potential loss of user confidence in the platform due to issues like security breaches, poorly managed projects, or unfulfilled promises.
- **User Engagement:** Low user engagement levels can lead to decreased user contributions and potentially drive users away from the platform.

7. Scalability Risks:

- **Infrastructure Scaling:** This includes the risk of increased user traffic and activity during the peak hours, leading to inconsistent user experience and service disruptions.
- **Performance under Load:** The platform may perform inefficiently under high loads and traffic spikes, affecting the smooth user experience with this app.

8. Technology Risks:

- **Technological Obsolescence:** The risk involves the possibility that the platform's technology stack becomes outdated and less competitive in the near future.

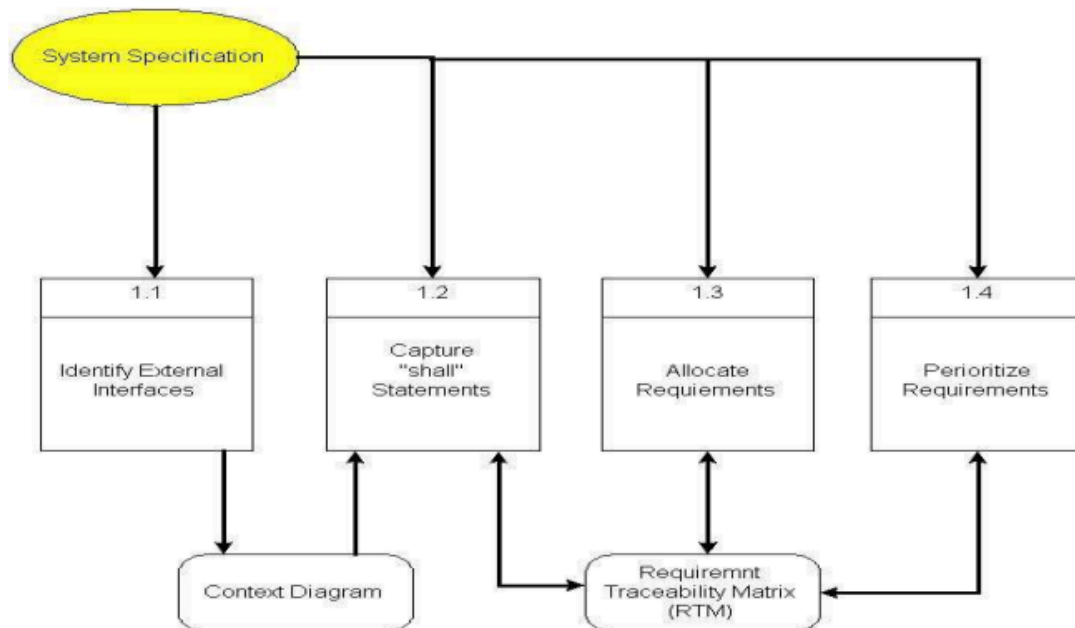
- **Software Bugs and Glitches:** The presence of software bugs can disrupt the functionality of this app, resulting in user frustration, loss of trust, and potentially causing users to seek alternatives.
- **Third-Party Service Dependencies:** The dependency on external services introduce risks related to service outages, changes in terms, or unexpected disruptions that can affect the platform's functionality

*****REQUIREMENTS ENGINEERING*****

1 Introduction

Requirements engineering process provides the appropriate mechanism for understanding what the customer wants, analysing need, assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification and managing the requirements as they are transformed into an operational system. The task of capturing, structuring, and accurately representing the user's requirements so that they can be correctly embodied in systems which meet those requirements (i.e. are of good quality).

- Requirements elicitation
- Requirements analysis and negotiation
- Requirements specification
- System modelling
- Requirements validation
- Requirements management



Here, requirements specification is to be discussed. Requirements specification would lead to the following four steps:

- Identify external interfaces
- Development of context diagram
- Capture “shall statements
- Allocate requirements
- Prioritise requirements
- Development of requirements traceability matrix

1.1 Systems Specifications

The following are the clauses that must be included while describing the system specifications.

Introduction

Introducing our Crowdfunding app, designed to serve as a web-based hub for creators to launch campaigns, showcase their projects, and receive financial support from investors, having the rights to remove or update their project

details from time to time. The investors can browse through a diverse range of projects from various categories, fund campaigns and can even leave a comment or query regarding the project. The platform caters to a diverse audience, including project creators, backers, and administrators responsible for managing platform operations.

Organisational Chart

There is not any specific organisational chart as we are doing this project on our own without involving any organisation.

Scope of the System

The system is divided into 3 phases:

Phase I: This phase involves the project planning, cost estimation, use-cases, project description and requirement gathering.

Phase II: The second phase involves the implementation of the web application.

Phase III: This phase involves the deployment of a web application and testing of it.

Summary of Requirements (Initial Requirements)

It should include search and filtering tools to find campaigns of interest. Users must be able to create accounts, providing essential details. Secure data handling must be implemented to protect user data and financial transactions because it is a top most priority. Detailed project descriptions and multimedia content should be accessible. This abstract will act as a foundation for the future analysis of the system.

1.2 Identifying External Entities

The identification of the external entities will be based on the information contained in your Abstract. This identification is done after two phases. We will map the “Green wood” case study to make things more comprehensible.

The Identification of External Entities is done in two phases

a. Over Specify Entities from Abstract

On the basis of the Abstract, one might identify the entities from the problem.

b. Perform Refinement

After over specifying the entities, you have to refine them on the basis of your business logic.

1.3 Context Level Data Flow Diagram

Context level data flow diagram contains only one process, representing the entire system. The process is given the number zero and all external entities are shown on the context diagram as well as major data flow to and from them. The diagram does not contain any data stores.

1.4 Capture "shall" Statements

Identify "shall" statements, as they would be all functional requirements.

Para No.	External Entity	Initial Requirements
1.0	Owner	An owner 'shall' have an easy-to-use interface to create and manage campaigns.
1.0	Owner	An owner 'shall' have a unique profile displaying the previous projects and communication history.
1.0	Owner	Owner 'shall' have the ability to set clear funding goals.
1.0	Owner	Owner 'shall' have a dashboard to overview the campaign's progress.
1.0	Owner	An owner 'shall' be able to share projects on social media platforms.
1.0	Owner	A owner 'shall' be able to receive funds cleared by backer
1.0	Backer	The backer 'shall' have a wide overlook of the ongoing campaigns
1.0	Backer	A backer 'shall' be able to contribute funds through a secure payment system.
1.0	Backer	A backer 'shall' be able to communicate with the project creator.
1.0	Backer	A backer 'shall' be able to discover a diverse range of projects
1.0	Backer	A backer 'shall' be able to provide feedback.
1.0	Backer	A backer 'shall' be able to overview the progress and details of the campaigns they have backed.

1.5 Allocate Requirements

Allocate the requirements in the use cases.

Para No.	Initial Requirements	Use Case Names
1.0	An owner 'shall' have an easy-to-use interface to create and manage campaigns.	UC_Interface
1.0	An owner 'shall' have a unique profile displaying the previous projects and communication history.	UC_ProjectCreation
1.0	Owners 'shall' have the ability to set clear funding goals.	UC_Funding_Plan
1.0	Owners 'shall' have a dashboard to overview campaign's progress.	UC_Owner_Dashboard
1.0	An owner 'shall' be able to share projects on social media platforms.	UC_Publish
1.0	The backer 'shall' have a wide overlook of the ongoing campaigns	UC_Overlook
1.0	A backer 'shall' be able to contribute funds through a secure payment system.	UC_Secure_Payment
1.0	A backer 'shall' be able to communicate with the project creator.	UC_Contact
1.0	A backer 'shall' be able to discover a diverse range of projects	UC_Search
1.0	A backer 'shall' be able to provide feedback.	UC_Feedback
1.0	A user 'shall' be able to login into his account.	UC_Login
1.0	A project owner 'shall' be able to receive funds from the backers	UC_Receive_Payment
1.0	A backer 'shall' be able to overview the progress and details of the campaigns they have backed.	UC_Backer_Dashboard

1.6 Prioritise Requirements

Requirements must be prioritised as this will help achieve tasks easily. Rank them as “highest, medium, and lowest”.

Par	Rank	Initial Requirements	UseCase ID	Use Case
-----	------	----------------------	------------	----------

a No.				Names
1.0	high	An owner 'shall' have an easy-to-use interface to create and manage campaigns.	UC_1	UC_Interface
1.0	high	An owner 'shall' have a unique profile displaying the previous projects and communication history.	UC_2	UC_ProjectCr eation
1.0	high	Owners 'shall' have the ability to set clear funding goals.	UC_3	UC_Funding_ Plan
1.0	medium	Owners 'shall' have a dashboard to overview campaign's progress.	UC_4	UC_Owner_D ashboard
1.0	low	An owner 'shall' be able to share projects on social media platforms.	UC_5	UC_Publish
1.0	high	The backer 'shall' have a wide overlook of the ongoing campaigns	UC_6	UC_Overlook
1.0	high	A backer 'shall' be able to contribute funds through a secure payment system.	UC_7	UC_Secure_P ayment
1.0	medium	A backer 'shall' be able to communicate with the project creator.	UC_8	UC_Contact
1.0	high	A backer 'shall' be able to discover a diverse range of projects	UC_9	UC_Search
1.0	low	A backer 'shall' be able to provide feedback.	UC_10	UC_Feedback
1.0	high	A user 'shall' be able to login into his account	UC_11	UC_Login
1.0	high	A project owner 'shall' be able to receive funds from the backers	UC_12	UC_Receive_ Payment
1.0	high	A backer 'shall' be able to overview the progress and details of the campaigns they have backed.	UC_13	UC_Backer_D ashboard

1.7 Requirements Trace-ability Matrix

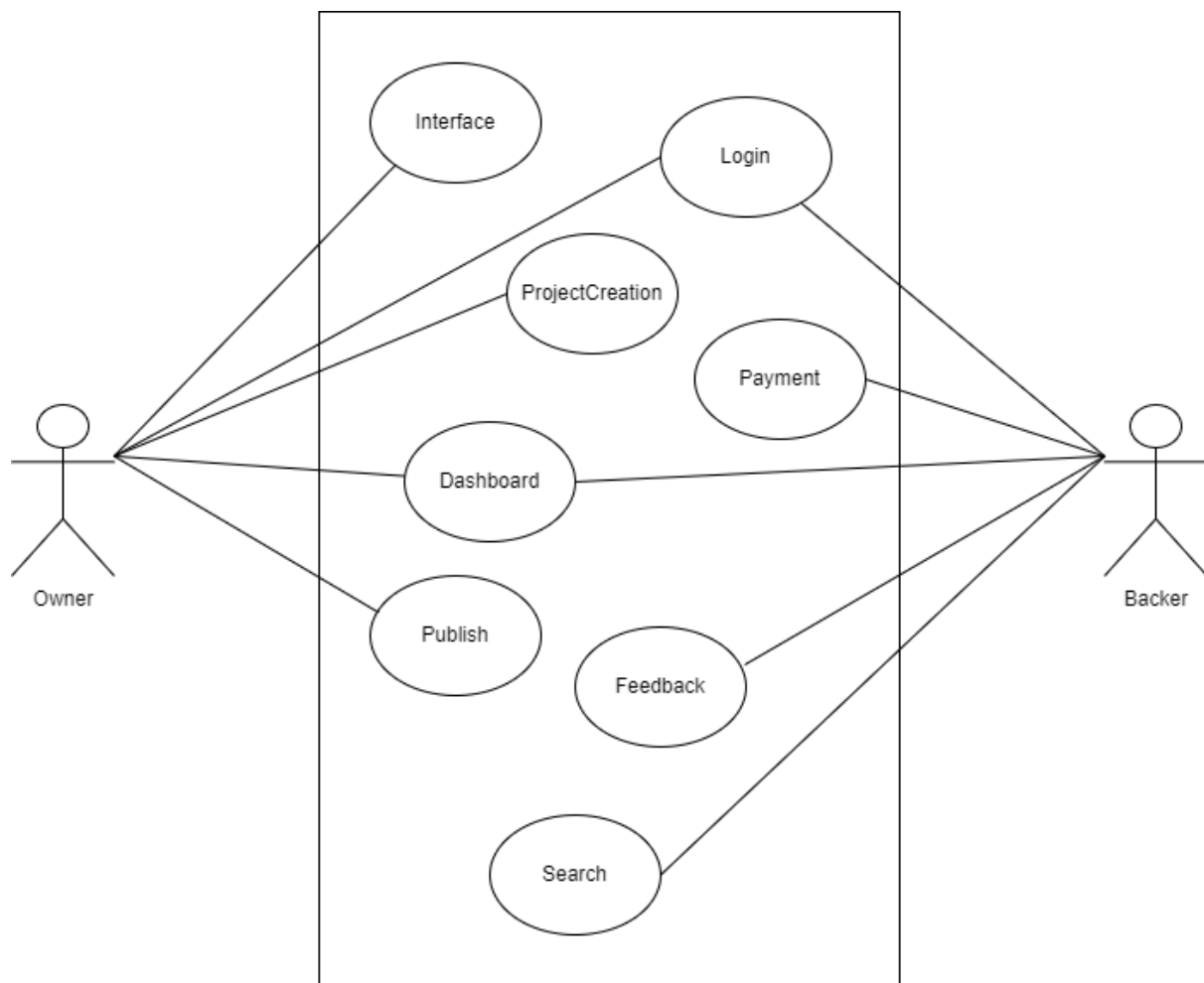
The requirements trace-ability matrix is a table used to trace project life cycle activities and work products to the project requirements. The matrix establishes a thread that traces requirements from identification through implementation.

Sr.	Para No.	Rank	Initial Requirements	Built	Use Case Names	Category
1	1.0	high	An owner 'shall' have an easy-to-use interface to create and manage campaigns.	B1	UC_Interface	Business
2	1.0	high	An owner 'shall' be able to create and display the projects.	B1	UC_ProjectCreation	Business
3	1.0	high	Owners 'shall' have the ability to set clear funding goals.	B1	UC_Funding_Plan	Business
4	1.0	medium	Owners 'shall' have a dashboard to overview campaign's progress.	B1	UC_Owner_Dashboard	Business
5	1.0	low	An owner 'shall' be able to share projects on social media platforms.	B1	UC_Publish	Business
6	1.0	high	The backer 'shall' have a wide overlook of the ongoing campaigns	B1	UC_Overlook	Business
7	1.0	high	A backer 'shall' be able to contribute funds through a secure payment system.	B1	UC_Secure_Payment	Business
8	1.0	medium	A backer 'shall' be able to communicate with the project creator.	B1	UC_Contact	Business
9	1.0	high	A backer 'shall' be able to discover a diverse range of projects	B1	UC_Search	Business
10	1.0	low	A backer 'shall' be able to provide feedback.	B1	UC_Feedback	Business
11	1.0	high	A user 'shall' be able to login	B1	UC_Login	Business
12	1.0	high	A owner "shall" be able to receive funds cleared by the backer.	B1	UC_Receive_Payment	Business
13	1.0	medium	A backer "shall" have a dashboard to overview campaign's progress.	B1	UC_Backer_Dashboard	Business

1.9 High Level Use Case Diagram

A use case scenario is a visual description, typically written in structured English or point form, of a potential business situation that a system may or

may not be able to handle. A use case defines a goal-oriented set of interactions between external actors and the system under consideration.





Second Deliverable for Object Oriented Approach

Second Deliverable:

Introduction:

Our crowdfunding platform aims to revolutionise the way individuals and businesses bring their ideas to life by fostering a vibrant community of supporters and innovators. Designed as a catalyst for dreams and initiatives, our website serves as a dynamic hub where creators can transform their visions into reality through the collective support of backers and enthusiasts. At the core of our platform is a commitment to empowering creators from diverse backgrounds. We strive to offer a space that transcends geographical boundaries, enabling creators to showcase their projects and connect with a global audience. From aspiring artists and entrepreneurs to social innovators and tech enthusiasts, our platform embraces a multitude of creative endeavours.

Our platform facilitates a seamless and user-friendly experience, providing creators with intuitive tools to craft compelling campaigns. Through engaging storytelling, multimedia elements, and interactive features, creators can effectively communicate their ideas, missions, and goals, inspiring backers to join their journey.

In addition to providing a launchpad for projects, our platform ensures transparency and trust between creators and backers. Robust support mechanisms, secure payment systems, and clear communication channels foster a safe and conducive environment for all participants. Backers have the opportunity to explore a diverse range of projects, make contributions, and engage directly with creators, thus becoming integral parts of the projects they support.

Beyond funding, our platform nurtures a sense of community and collaboration. It encourages meaningful interactions, feedback sharing, and the celebration of milestones, creating a supportive ecosystem where ideas flourish and innovations thrive.

Ultimately, our crowdfunding website is more than a funding platform; it is a catalyst for creativity, a conduit for innovation, and a global stage where

dreams take flight, driven by the collective power of passionate creators and dedicated backers.

Use Case Description:

1. UC_Interface:

Use Case Description:

This use case allows the owner to have an easy-to-use interface to create and manage campaigns.

Pre-Condition:

- The website is accessible and functional without any errors or technical failures.
- The user must have login.

Basic Flow:

- The owner accesses the website dashboard.
- The owner navigates to the campaign creation section.
- The owner fills in campaign details, including title, description, funding goal, etc.
- The owner manages campaigns through the dashboard (edit, delete, update).

Alternate Flows:

If there's a system failure, an error message prompts the user to retry.

Post Condition:

The owner successfully creates and manages campaigns using an intuitive interface.

2. UC_ProjectCreation:

Use Case Description:

This use case enables owners to create and display projects.

Pre-Condition:

The website is operational without any system issues.

Basic Flow:

- The owner logs in to the platform.
- The owner navigates to the project creation section.
- The owner enters project details: title, description, images, videos, etc.
- The owner publishes the project.

Alternate Flows:

If the owner doesn't complete mandatory fields, an alert prompts them to fill in all required information.

Post Condition:

A new project is successfully created and displayed on the platform.

3. UC_Funding_Plan:

Use Case Description:

This use case enables owners to set clear funding goals for their projects

.

Pre-Condition:

The website is operational and accessible to project owners.

Basic Flow:

- The owner accesses the project details section.
- The owner sets a clear funding goal (financial target) for the project.

Alternate Flows:

If the funding goal isn't set or is invalid, the system prompts the owner to input a valid target.

Post Condition:

The funding goal is successfully defined for the project.

4. UC_Owner_Dashboard:

Use Case Description:

This use case provides owners with a dashboard to overview campaign progress.

Pre-Condition:

The website is functional without any technical issues.

Basic Flow:

- The owner logs in to their account.
- The owner accesses the dashboard section.
- The owner views an overview of campaign progress, including funding status, backer statistics, etc.

Alternate Flows:

If there's a system failure, the owner receives an error message and can retry.

Post Condition:

The owner can effectively monitor and track campaign progress through the dashboard.

5. UC_Publish:**Use Case Description:**

This use case facilitates project owners to share their projects on various social media platforms.

Pre-Condition:

The website is operational and accessible to project owners.

Basic Flow:

- The project owner logs into their account.
- The owner navigates to the project details or management section.
- The owner selects the project they wish to share.
- The system presents the owner with social media sharing options (e.g., Facebook, Twitter, LinkedIn).
- The owner clicks on the preferred social media platform for sharing.
- The system redirects the owner to the chosen platform's sharing interface.
- The owner customises the shared content (optional) and publishes the project on the selected platform.

Alternate Flows:

If there's an issue with social media platform integration, the system notifies the owner to try again later or choose another platform.

Post Condition:

The project owner successfully shares their project on the selected social media platform(s), enhancing visibility and outreach for the crowdfunding campaign.

6. UC_Overlook:

Use Case Description:

This use case enables backers to have a comprehensive overview of ongoing crowdfunding campaigns available on the platform.

Pre-Condition:

The website is operational and accessible to backers.

Basic Flow:

- The backer logs into their account.
- The backer navigates to the section displaying ongoing campaigns or a specific "Overlook" dashboard.
- The system presents a list or grid view of various ongoing campaigns, including details like campaign name, description, funding status, progress, and deadline.
- The backer can filter, sort, or search for specific campaigns based on preferences.
- By clicking on a campaign, the backer accesses more detailed information regarding the campaign goals, funding tiers, creator information, and perks for contributors.

Alternate Flows:

If there are no ongoing campaigns or if there's an error in loading the campaigns, the system displays a message indicating that no campaigns are available or prompts the user to try again later.

Post Condition:

The backer gains a comprehensive overview of the ongoing crowdfunding campaigns, allowing them to explore and select campaigns they might be interested in supporting.

7. UC_Secure_Payment:

Use Case Description:

This use case enables backers to contribute funds securely to the selected crowdfunding campaigns.

Pre-Condition:

The backer is logged into their account on the crowdfunding platform.

Basic Flow:

- The backer selects a particular campaign they wish to support.
- The backer reads through the campaign details, funding goals, and contribution tiers available.
- The backer selects the desired funding tier or enters a custom amount.
- The system prompts the backer to confirm the contribution amount and details.
- The backer chooses a secure payment method (e.g., credit/debit card, PayPal, etc.).
- The system processes the payment securely through an encrypted and reliable payment gateway.
- Upon successful payment processing, the system generates a confirmation receipt for the backer.

Alternate Flows:

If there's an issue with payment processing, the system prompts the backer to retry or choose an alternative payment method.

Post Condition:

The backer successfully contributes funds to the selected campaign through a secure payment system, supporting the campaign's funding goals.

8. UC_Contact:

Use Case Description:

This use case enables backers to communicate directly with the project creator or campaign owner.

Pre-Condition:

The backer is logged into their account on the crowdfunding platform.

Basic Flow:

- The backer navigates to the campaign or project they wish to communicate about.

- The system provides an option or button to contact the project creator.
- The backer clicks on the "Contact" button, triggering a communication interface (e.g., messaging system, email form) to initiate contact with the project creator.
- The backer writes their message or query and sends it to the project creator.
- The system notifies the project creator about the received message and facilitates communication between the backer and project creator.

Alternate Flows:

If the project creator has disabled communications or is unavailable, the system informs the backer about the unavailability of direct contact options.

Post Condition:

The backer initiates communication with the project creator, allowing inquiries, clarifications, or discussions related to the project.

9. UC_Search:

Use Case Description:

This use case allows backers to discover a diverse range of crowdfunding projects available on the platform.

Pre-Condition:

The backer is logged into their account on the crowdfunding platform.

Basic Flow:

- The backer accesses the search functionality or search bar provided on the platform.
- The backer enters specific keywords, categories, or filters to search for projects of interest.
- The system retrieves and displays a list of projects matching the entered search criteria.
- The backer can browse through the search results, view project details, and select projects to explore further.

Alternate Flows:

If the search query returns no results or if there are connectivity issues, the system informs the backer that no projects match the search criteria or prompts the user to refine the search.

Post Condition:

The backer discovers a diverse range of projects based on their search criteria, enabling exploration and selection of interesting projects.

10. UC_Feedback:**Use Case Description:**

This use case allows backers to provide feedback regarding their experiences or opinions about the crowdfunding platform or specific campaigns.

Pre-Condition:

The backer is logged into their account on the crowdfunding platform.

Basic Flow:

- The backer navigates to the feedback section or clicks on the feedback option provided by the platform.
- The system presents a form or interface where the backer can input their feedback, suggestions, or comments.
- The backer writes their feedback and submits the form or message.
- The system acknowledges the feedback submission and stores it for further analysis or improvements.

Alternate Flows:

If the backer does not submit any feedback or leaves the feedback field empty, the system prompts the backer to enter feedback.

Post Condition:

The backer successfully provides feedback, contributing to the platform's enhancement or addressing concerns related to campaigns or the platform itself.

11.UC_Login:**Use Case Description:**

This use case enables users to access the platform by logging into their respective accounts.

Pre-Condition:

The user navigates to the login page of the crowdfunding platform.

Basic Flow:

- The user enters their credentials (username/email and password) in the provided login fields.
- The system verifies the entered credentials against the database records.
- If the credentials are valid and match the system records, the user is granted access to their account.
- The system redirects the user to the homepage or user-specific dashboard upon successful login.

Alternate Flows:

If the entered credentials are incorrect or not found in the system records, the system prompts the user to re-enter the correct credentials or reset their password.

Post Condition:

The user successfully logs in to their account, gaining access to the platform's features and functionalities.

12. UC_Receive_Payment:**Use Case Description:**

This use case enables project owners or campaign creators to receive funds that have been cleared by backers.

Pre-Condition:

The campaign reaches its funding goal, and the backer's payments are processed and cleared.

Basic Flow:

- Once the campaign funding goal is met or exceeded, the system initiates the process to transfer the cleared funds to the project owner.
- The system processes the transaction and transfers the funds to the designated account of the project owner or campaign creator.
- The system generates a confirmation notification to inform the project owner about the successful transfer of funds.

Alternate Flows:

If there are delays or issues in processing the funds, the system notifies the project owner about the delay and takes necessary actions to resolve the issue.

Post Condition:

The project owner receives the cleared funds from the backers, facilitating the implementation or execution of the funded project or campaign.

13. UC_Backer_Dashboard:

Use Case Description:

This use case provides a dashboard interface for backers to overview the progress and details of the campaigns they have backed.

Pre-Condition:

The backer is logged into their account on the crowdfunding platform.

Basic Flow:

- The backer accesses the "Dashboard" section or feature available in their account.
- The system displays a summary or list of campaigns/projects backed by the user along with their respective progress details, funding status, and updates.
- The backer can view details such as funding amounts, campaign statuses, project updates, and expected delivery information for each backed campaign.

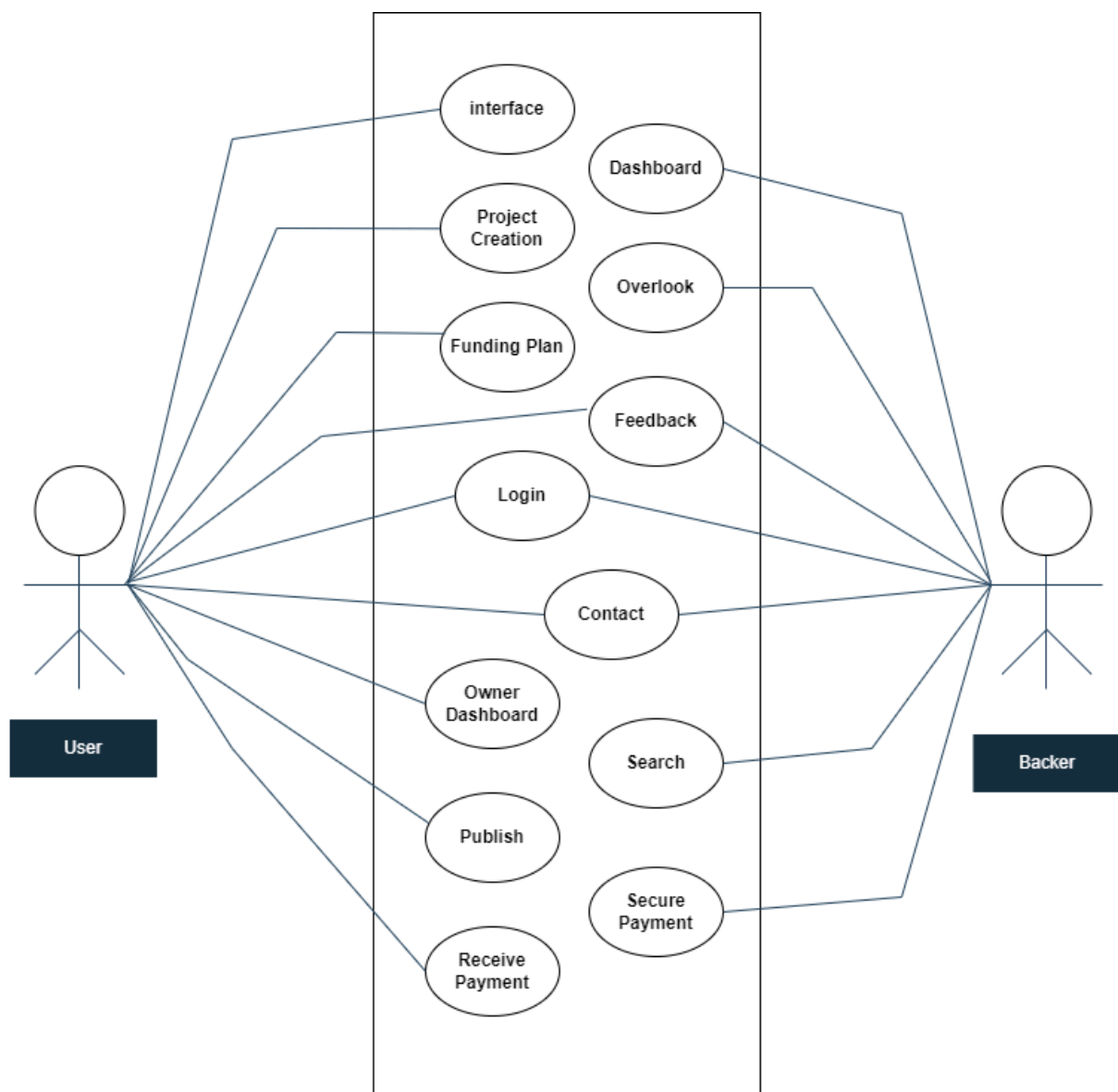
Alternate Flows:

If there are connectivity issues or delays in retrieving campaign details, the system informs the backer about the temporary unavailability of the dashboard or prompts the user to try again later.

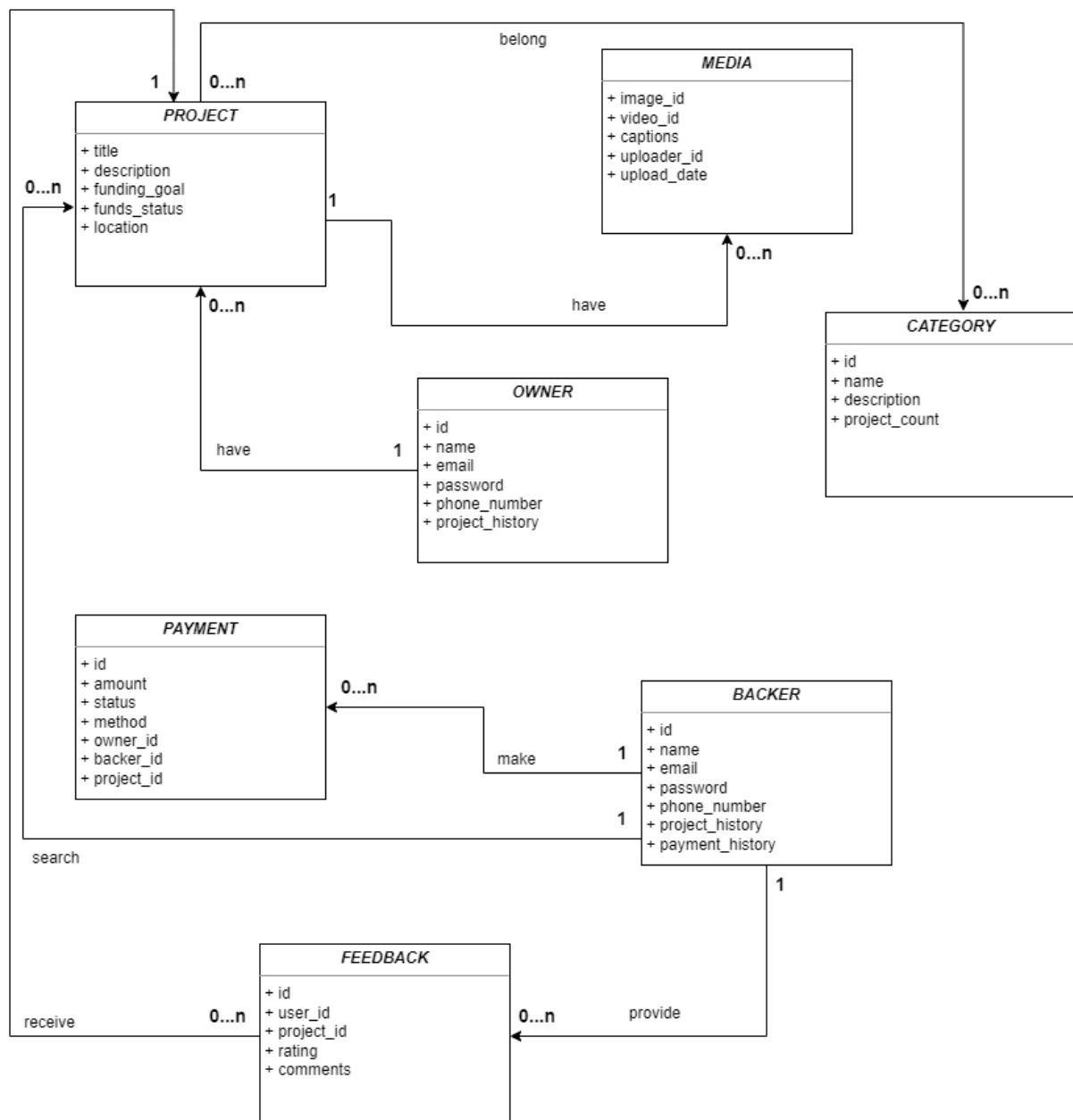
Post Condition:

The backer successfully views an overview of the campaigns they have backed through the dashboard, enabling them to track the progress and status of their supported projects.

Use Case Diagram (refined and updated):

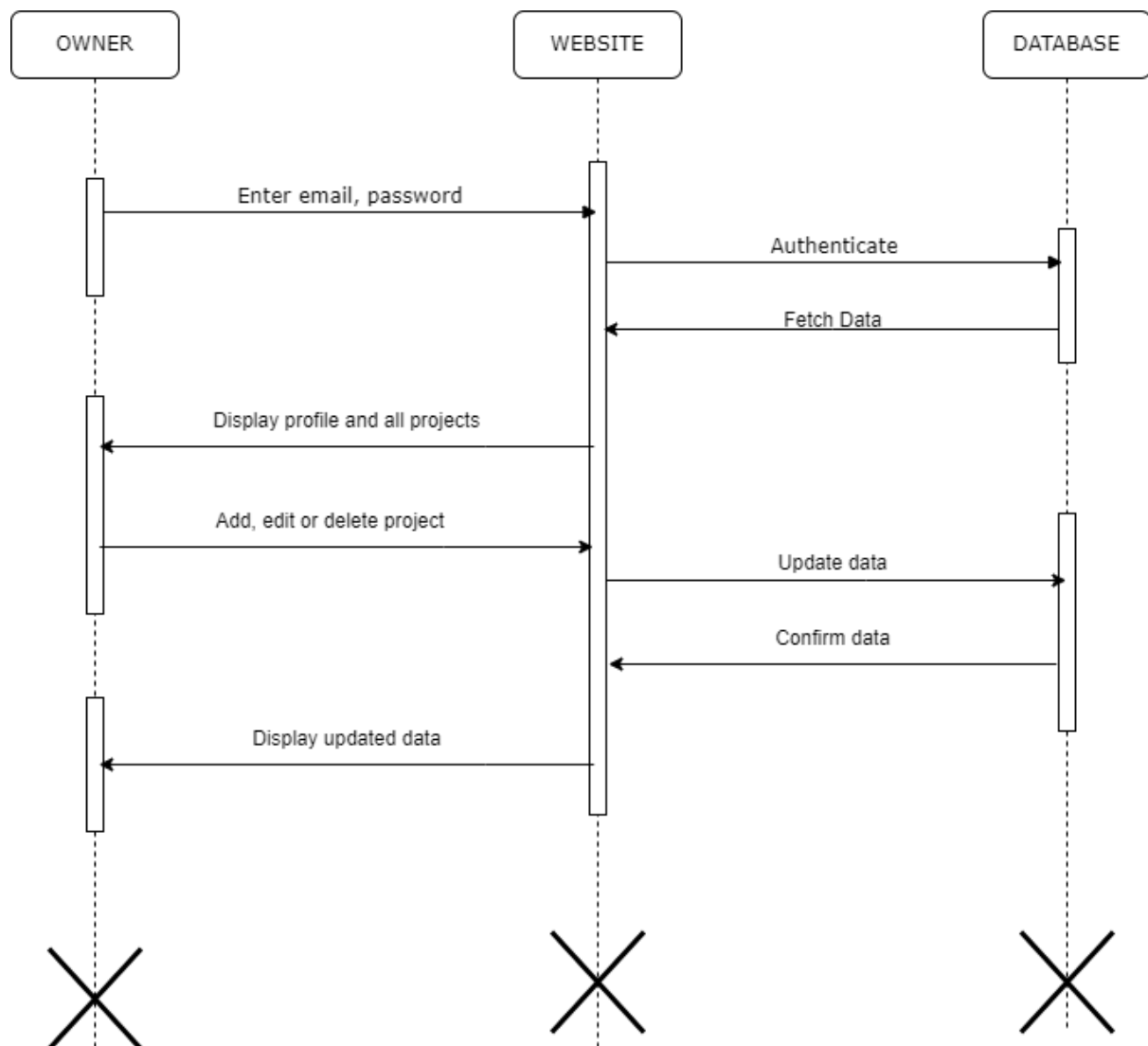


Domain Model:

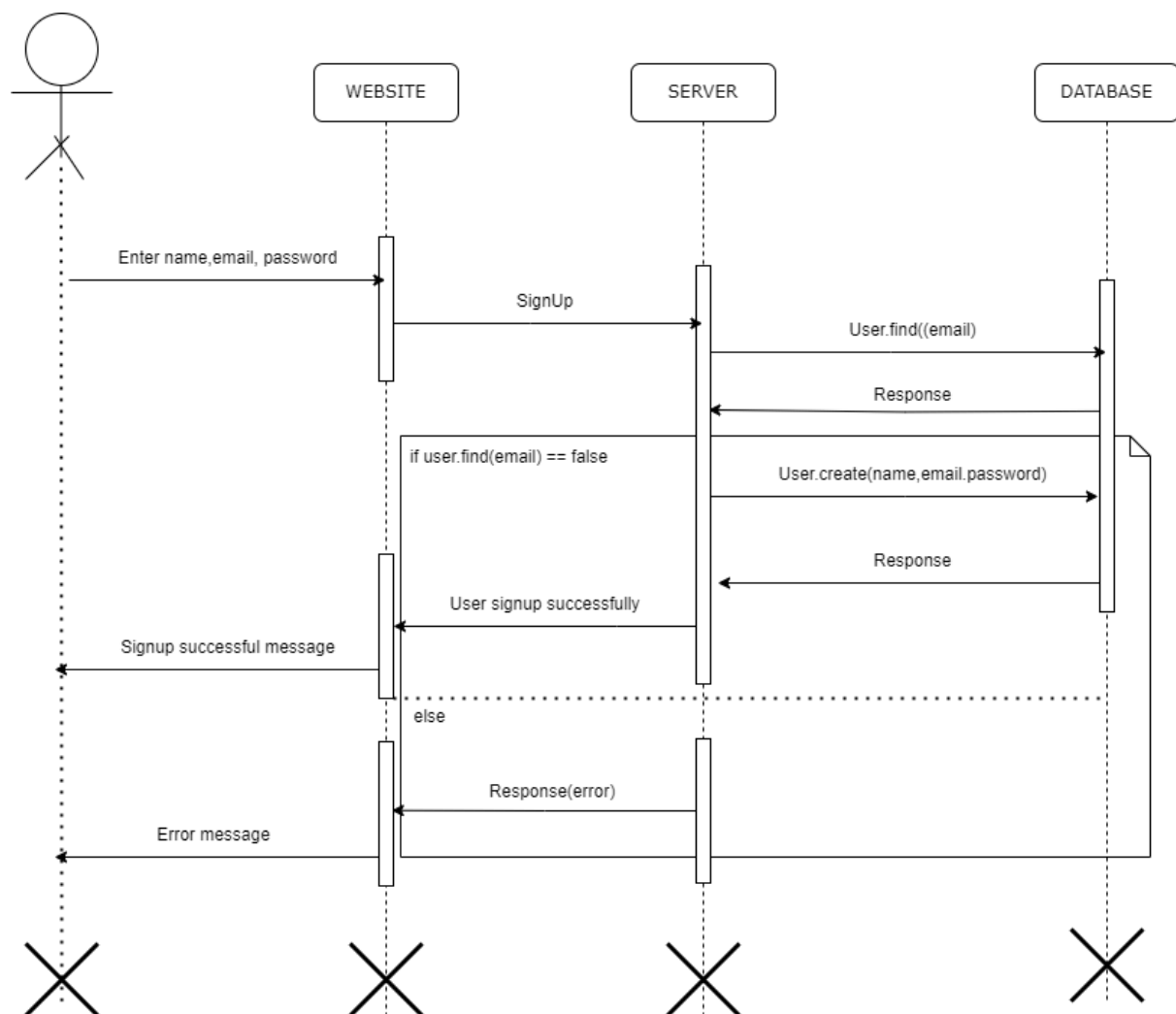


Sequence Diagram

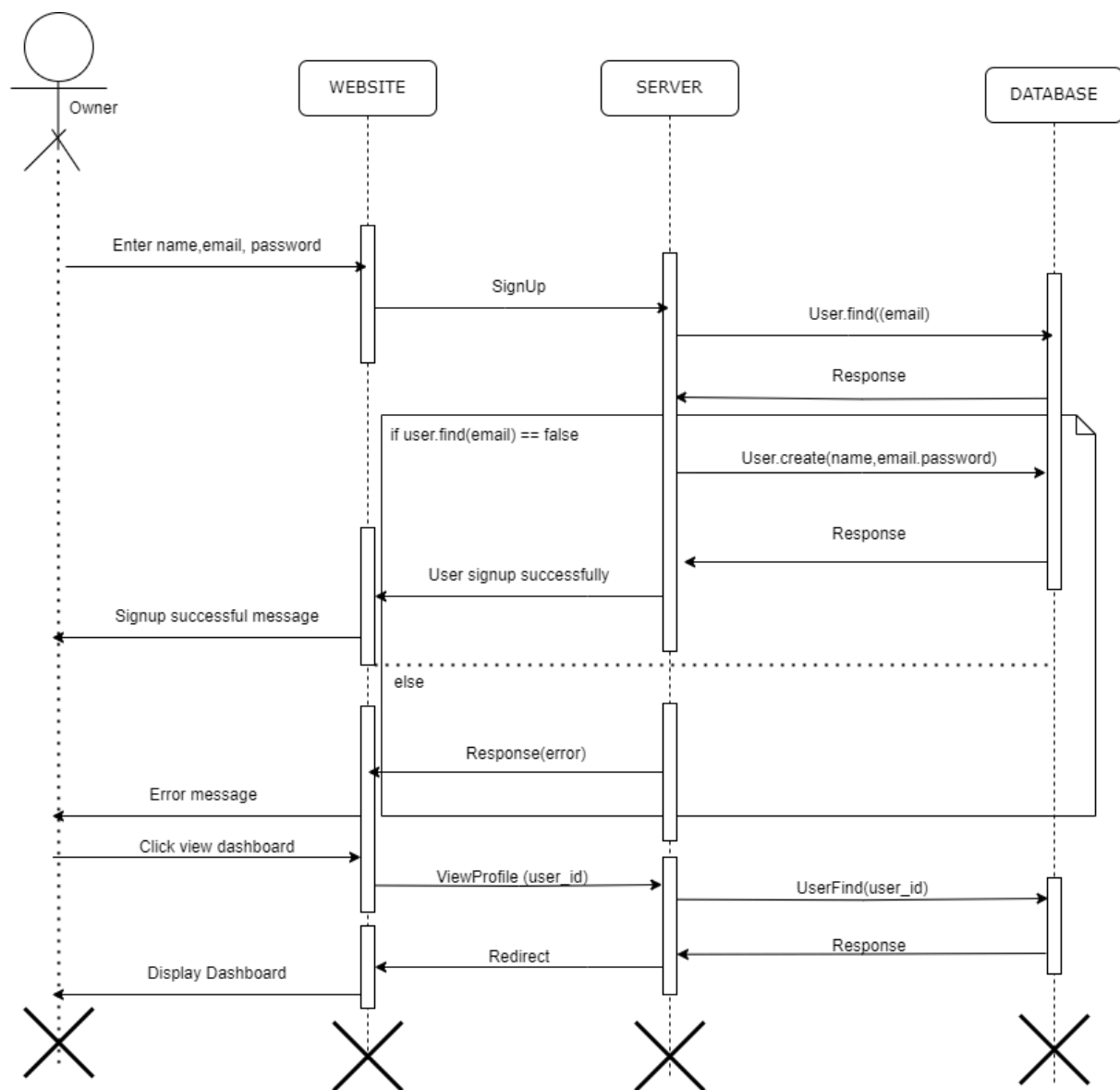
1. UC_Interface:



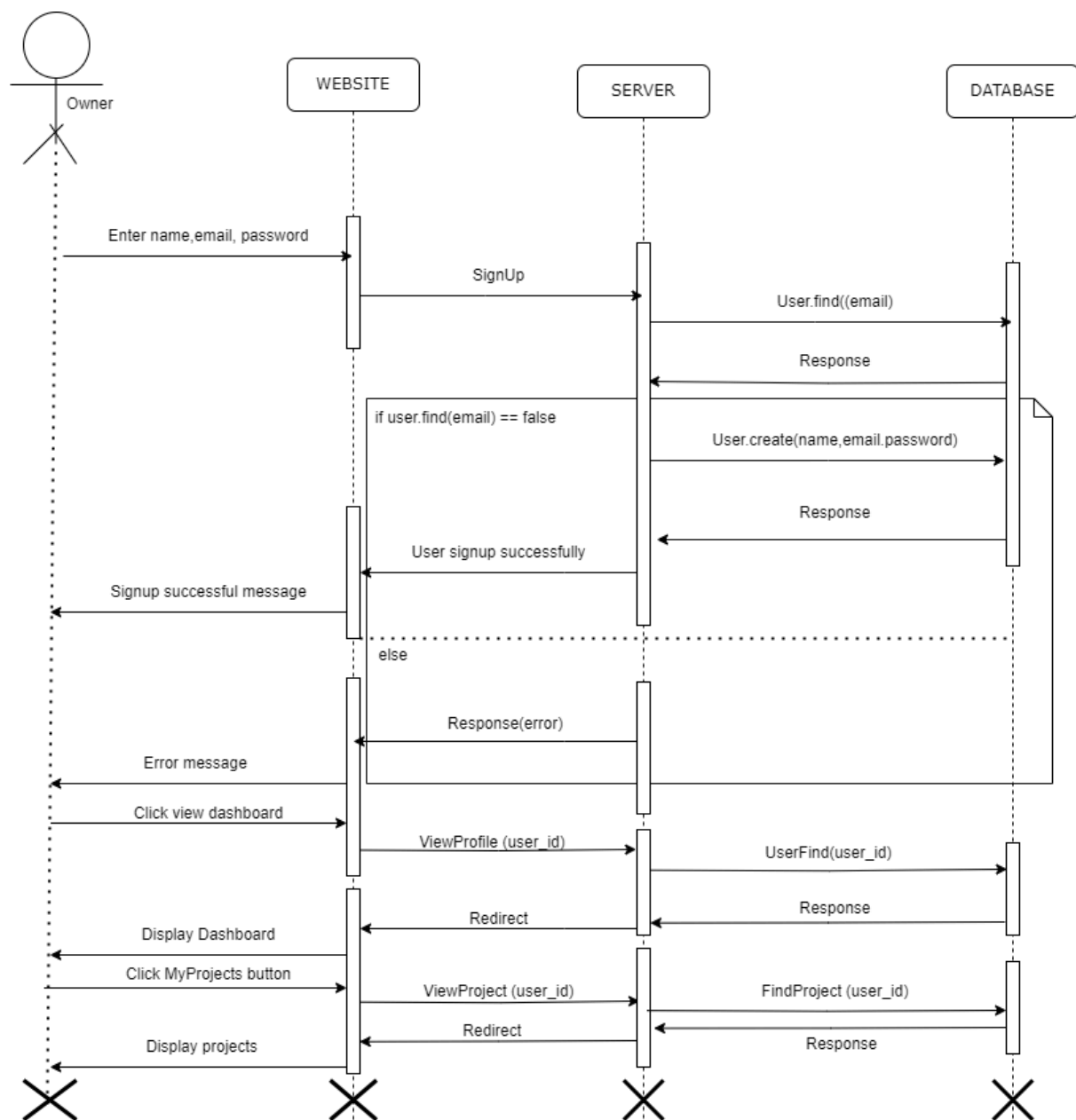
2. UC_Login:



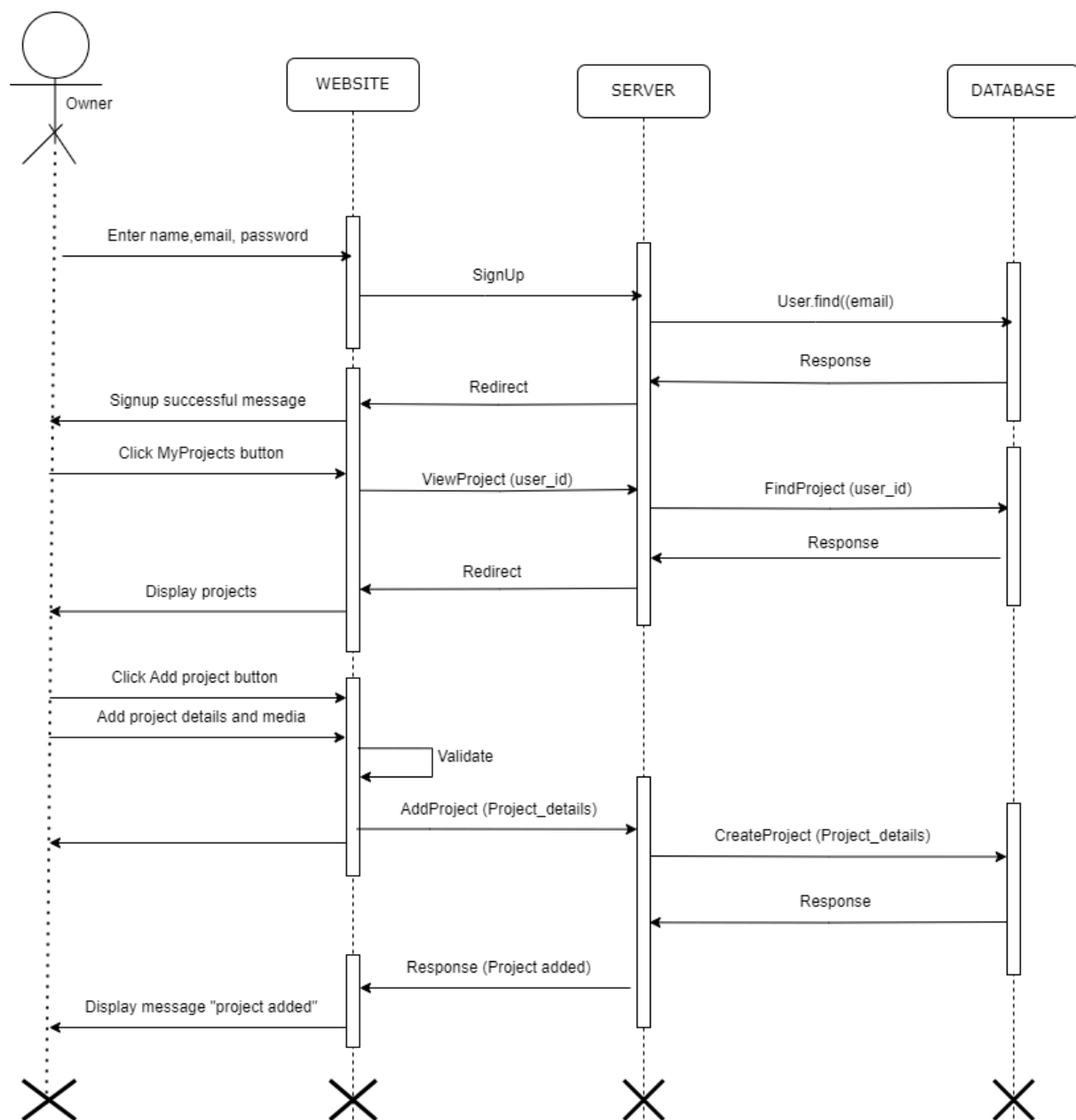
3. UC_Owner_Dashboard:



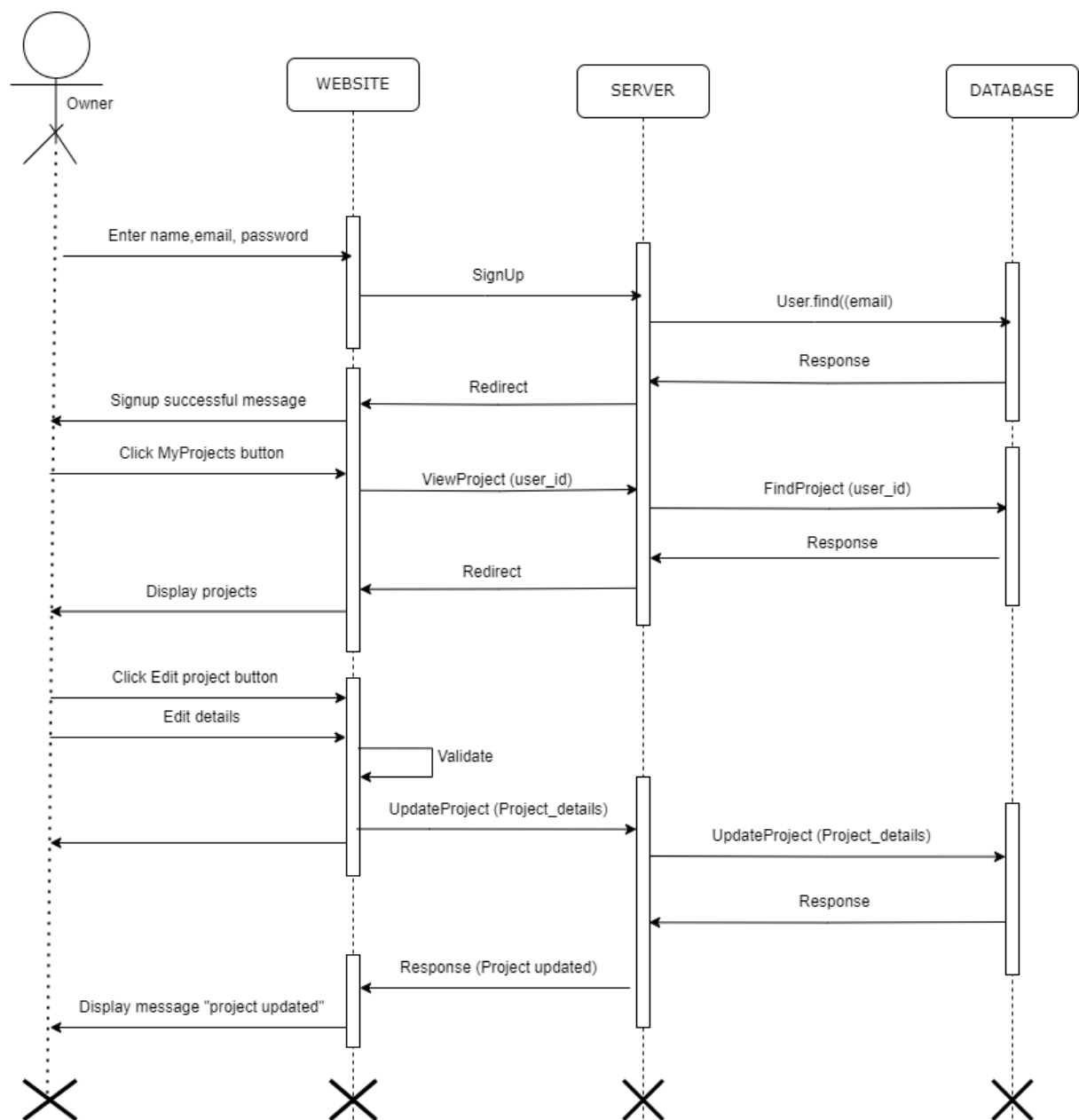
4. UC_Owner_Projects:



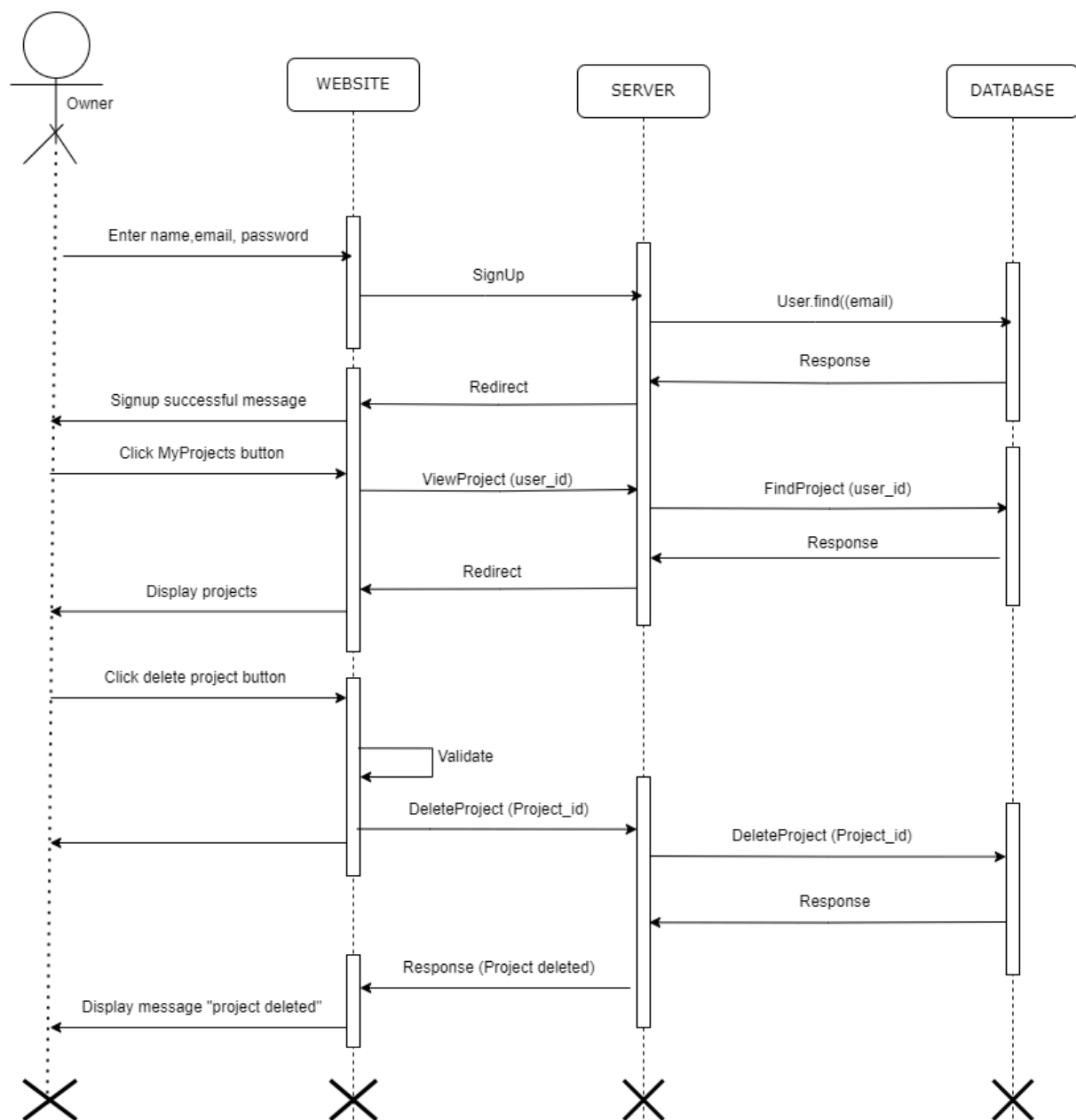
5. UC_ProjectCreation:



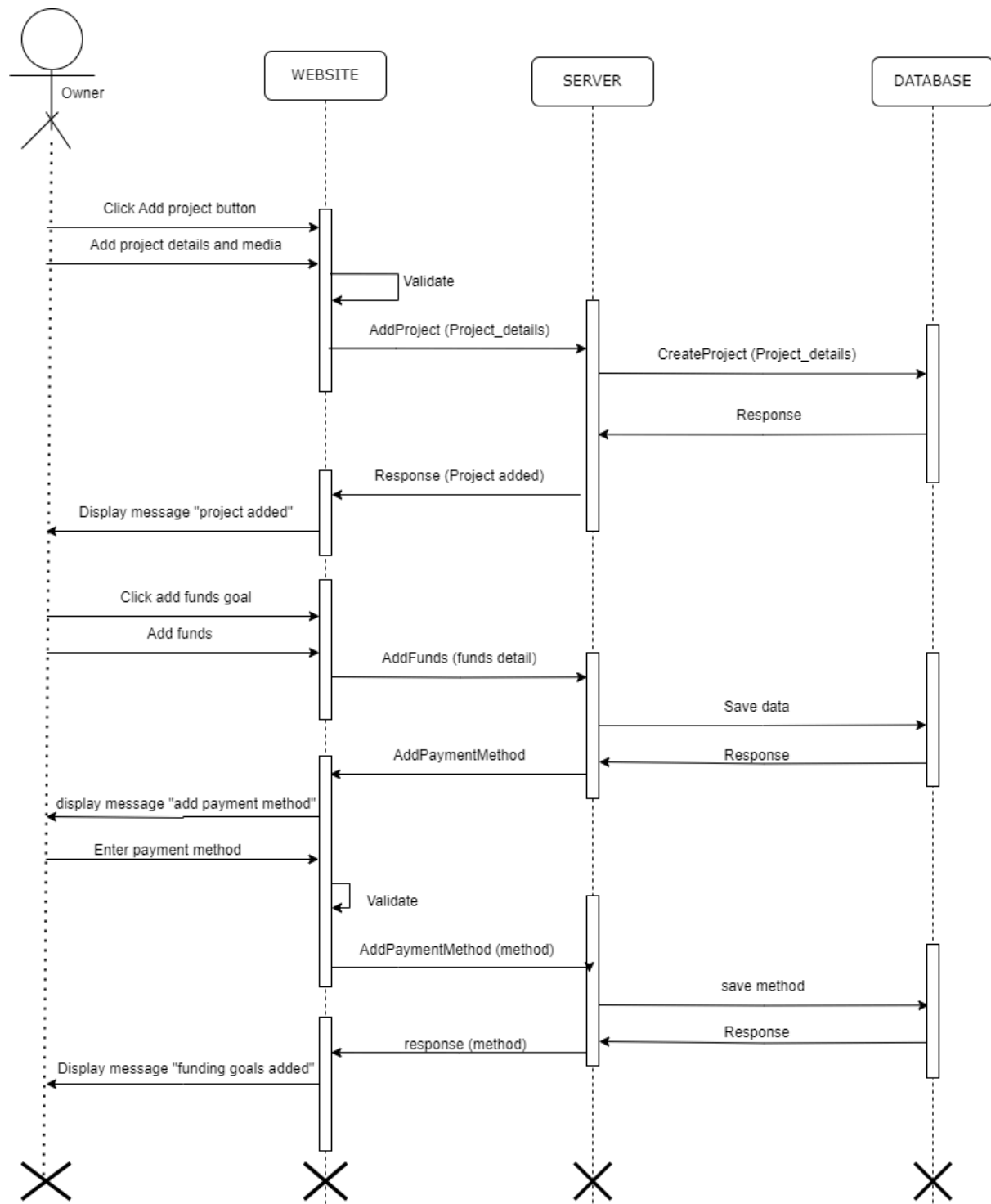
6. UC_UpdateProject:



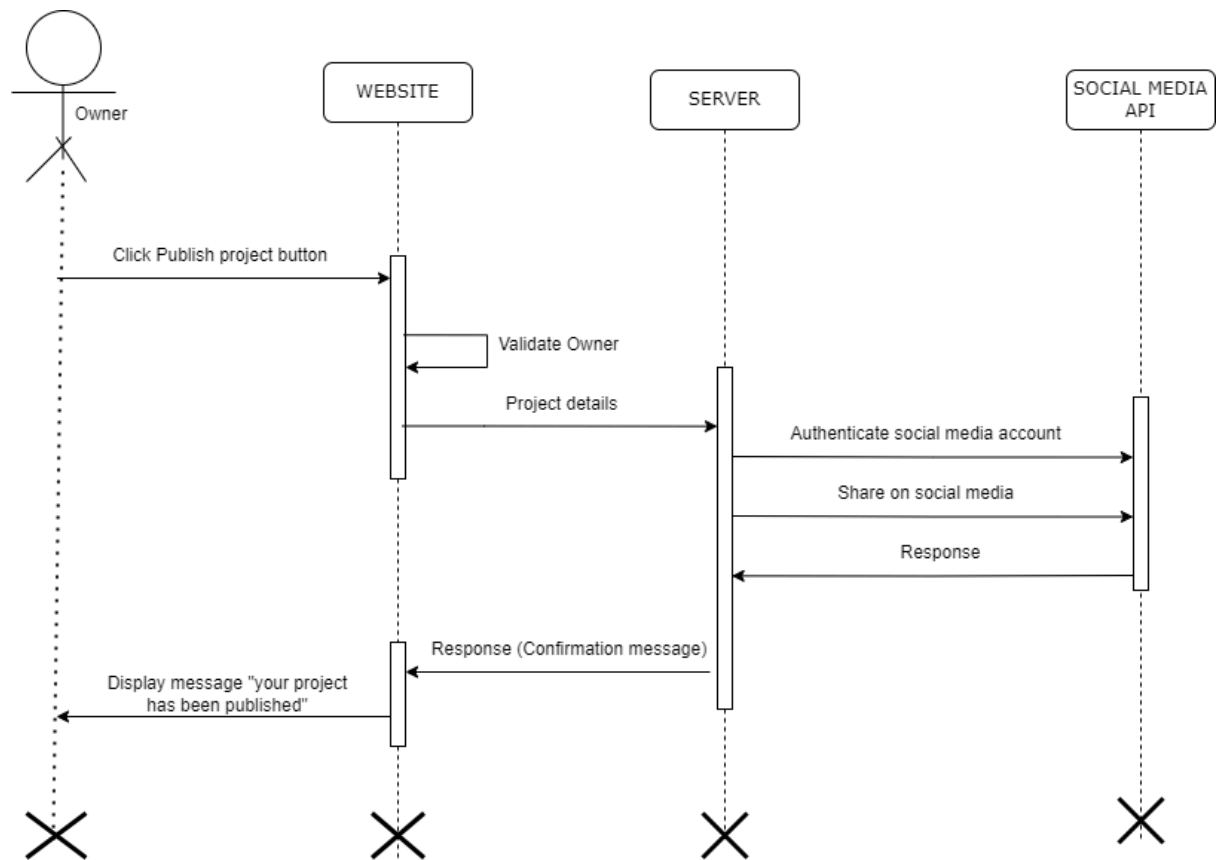
7. UC_DeleteProject:



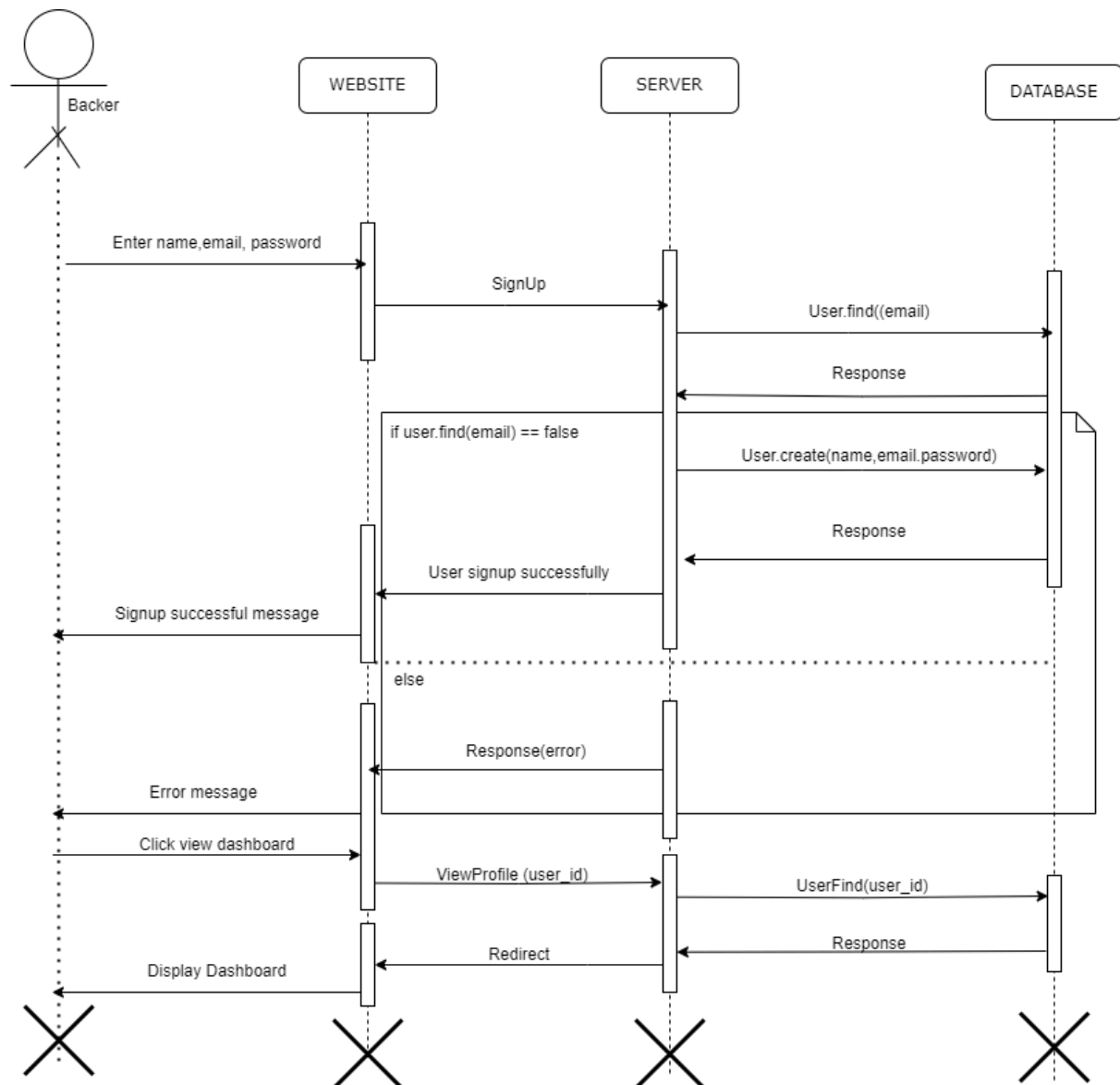
8. UC_Funding_Plan:



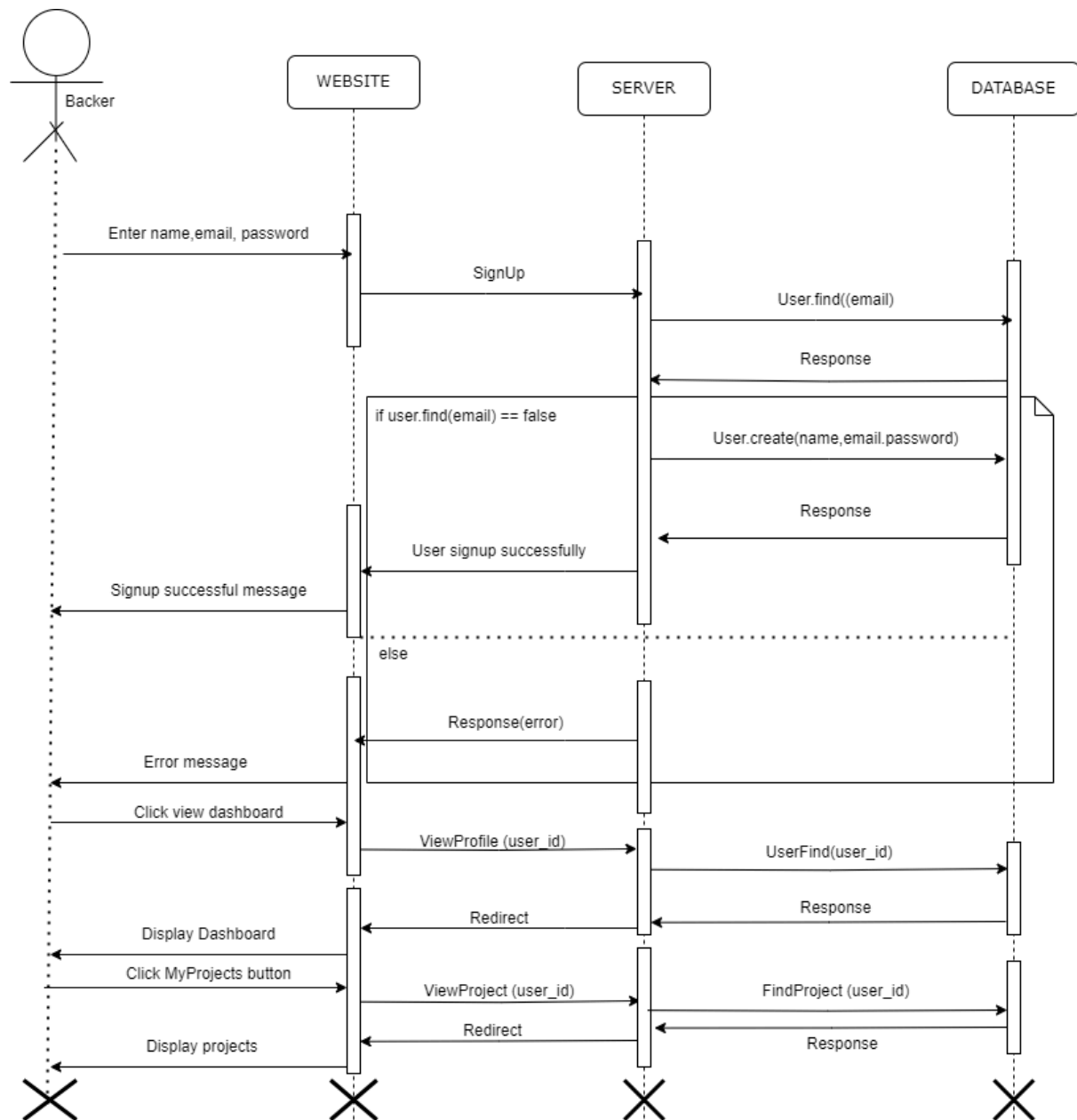
9. UC_Publish:



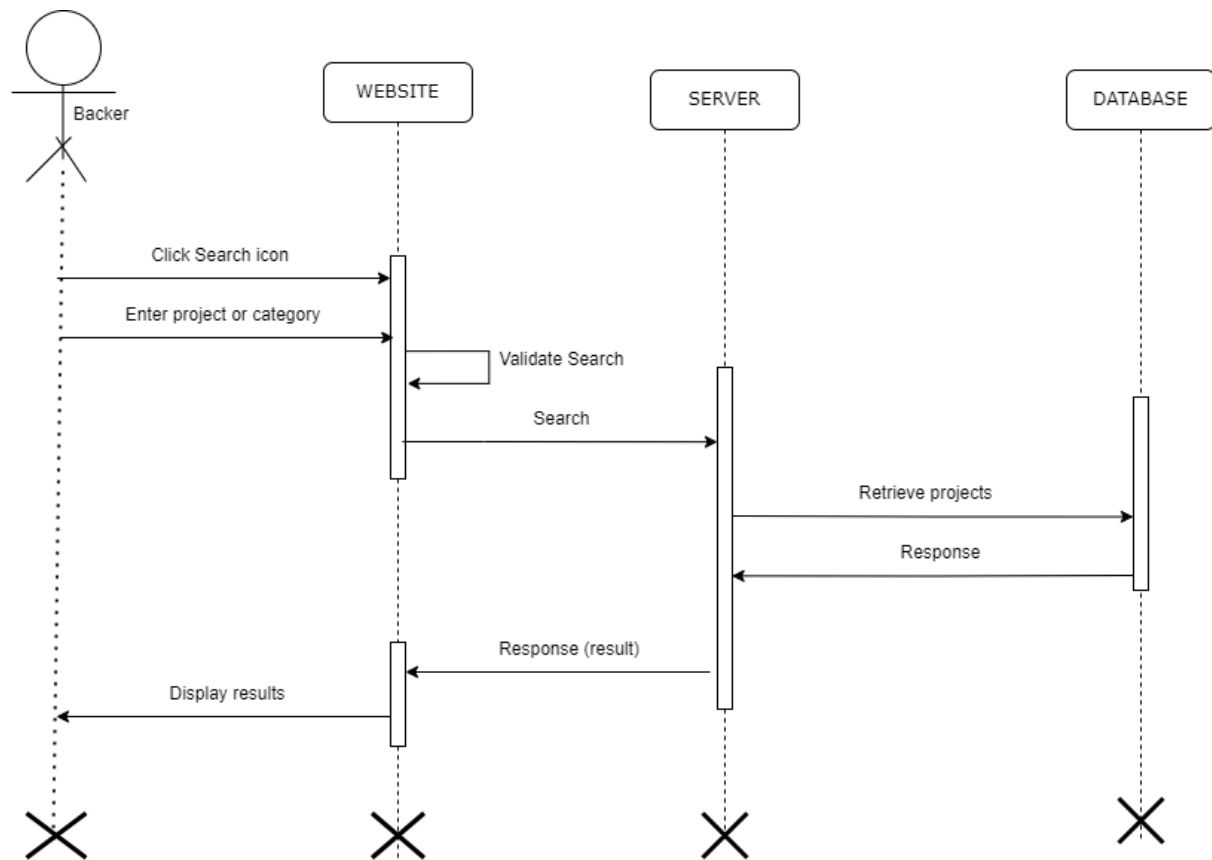
10. UC_Backer_Dashboard:



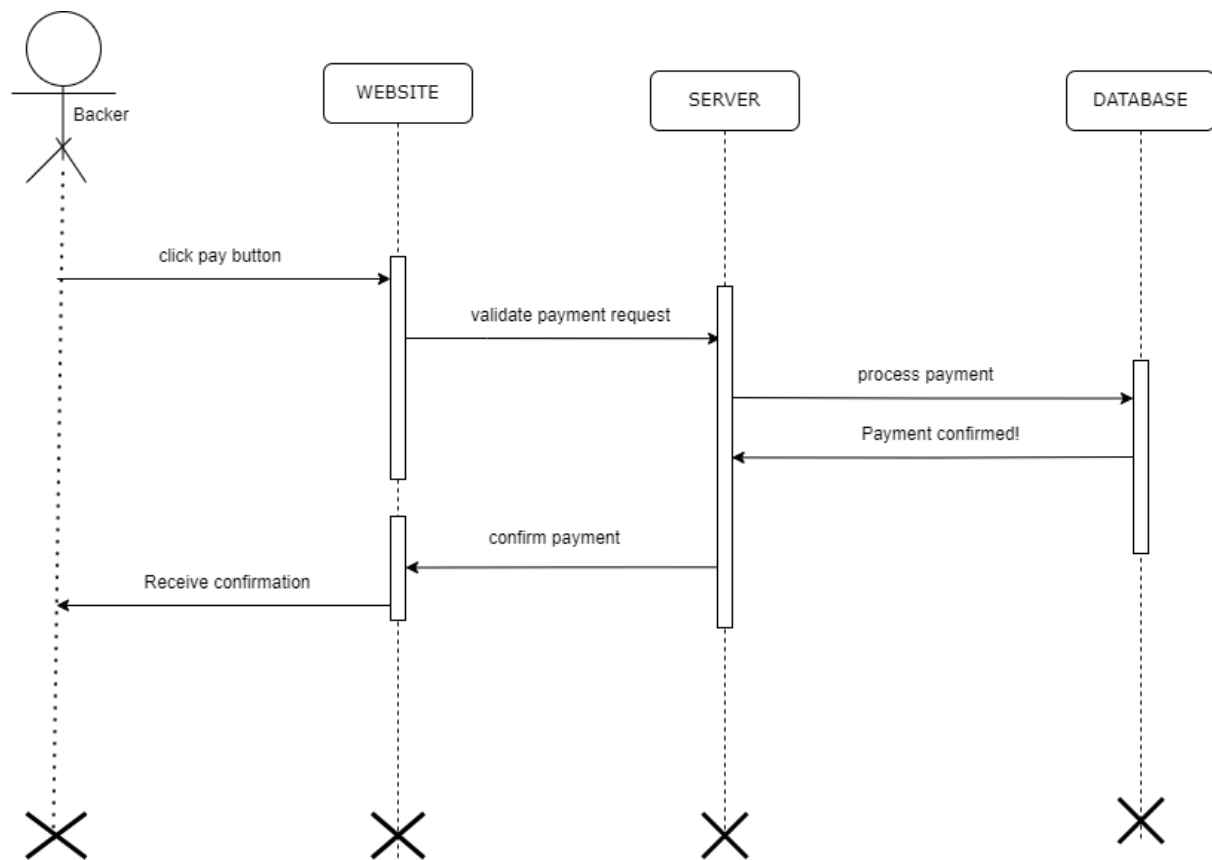
11. UC_Overlook:



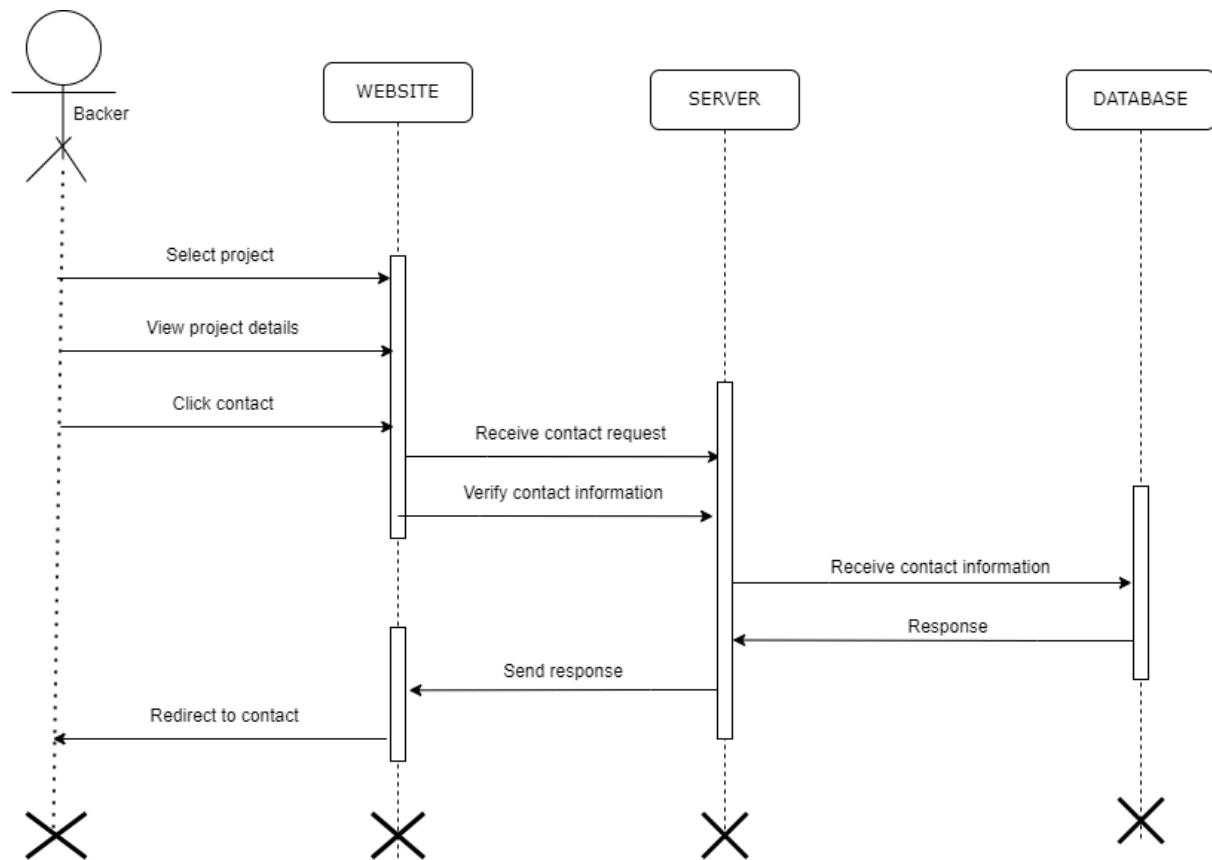
12. UC_Search:



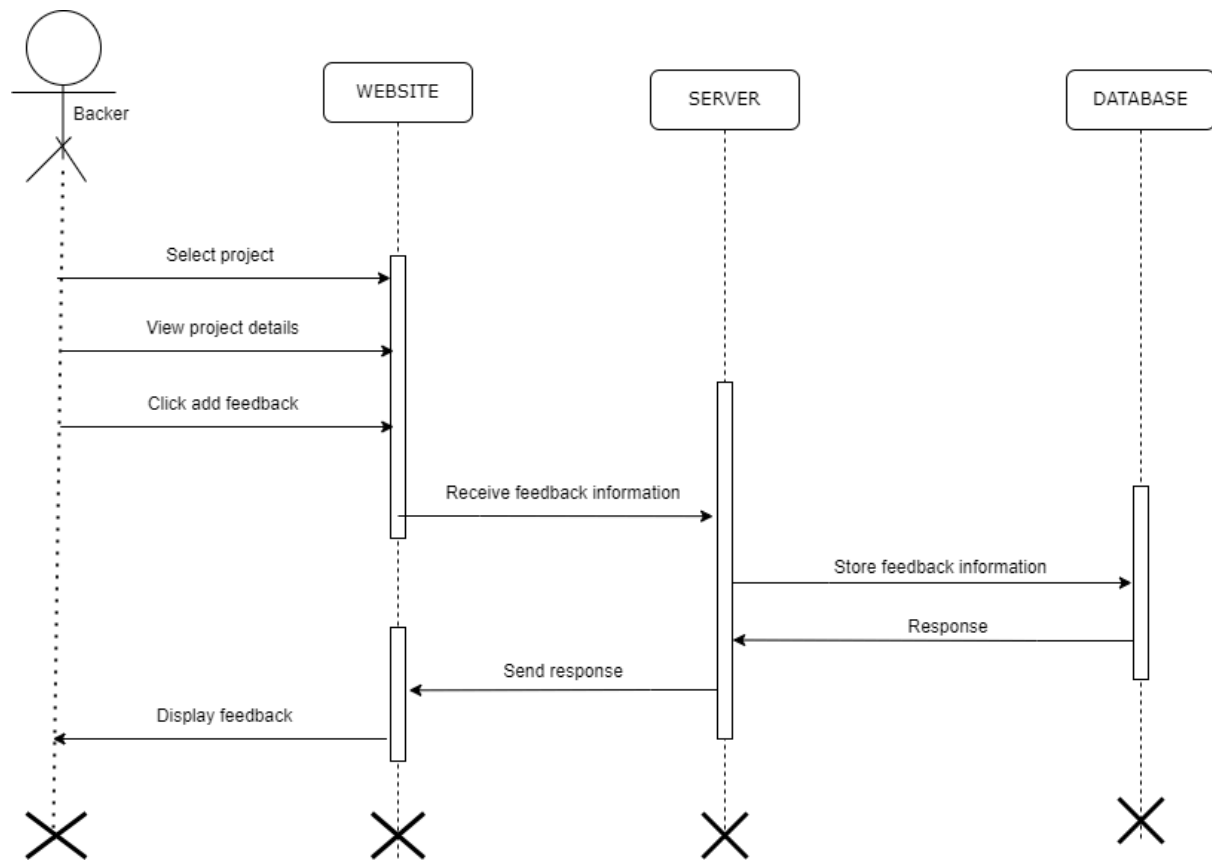
13. Secure_Payment:



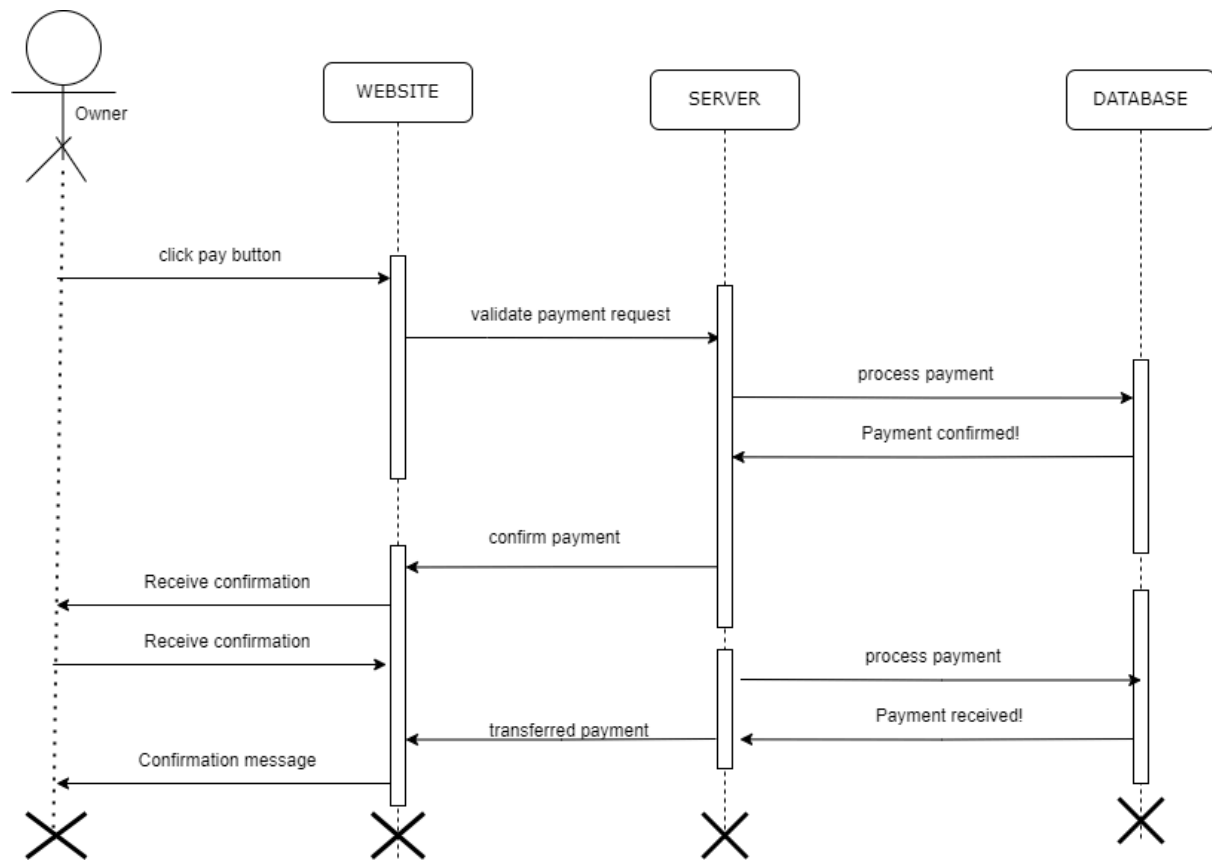
14. UC_Contact:



15. UC_Feedback:

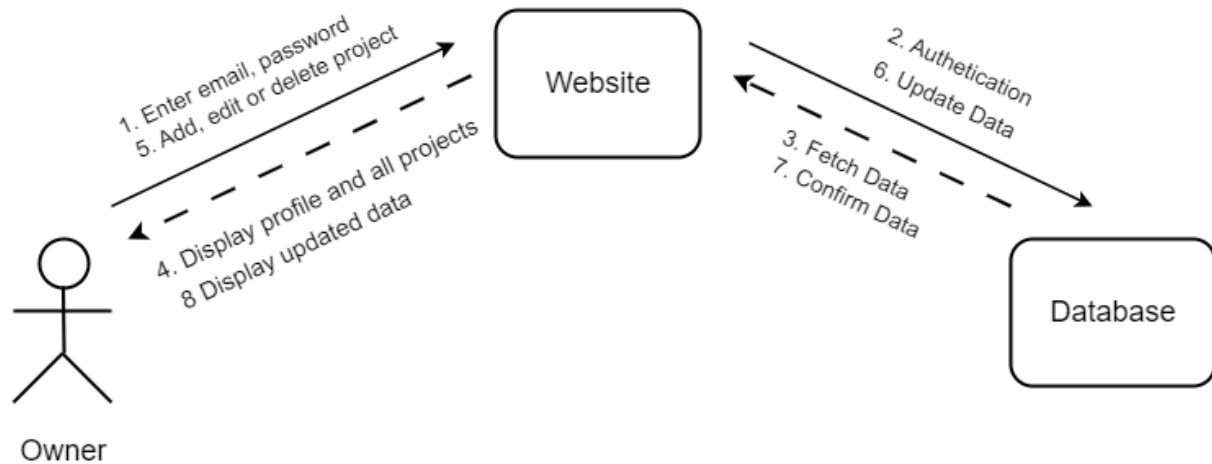


16. UC_Receive_Payment:

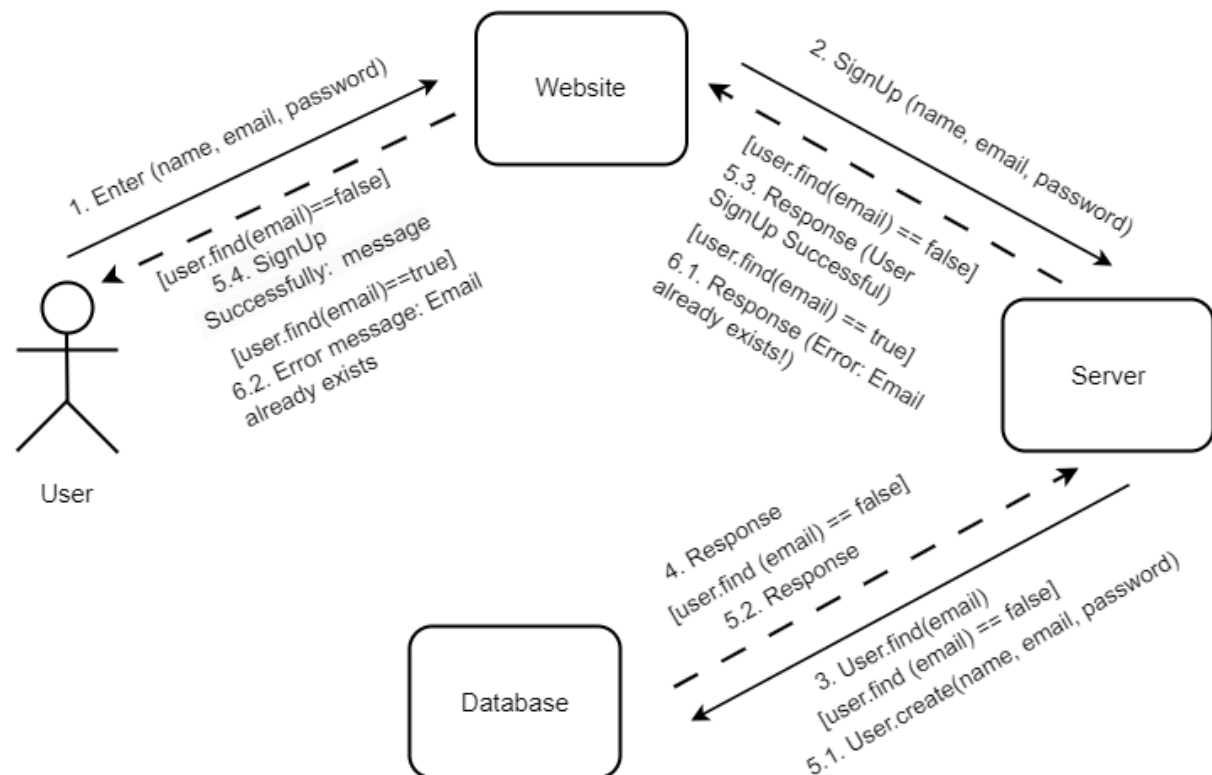


Collaboration Diagram

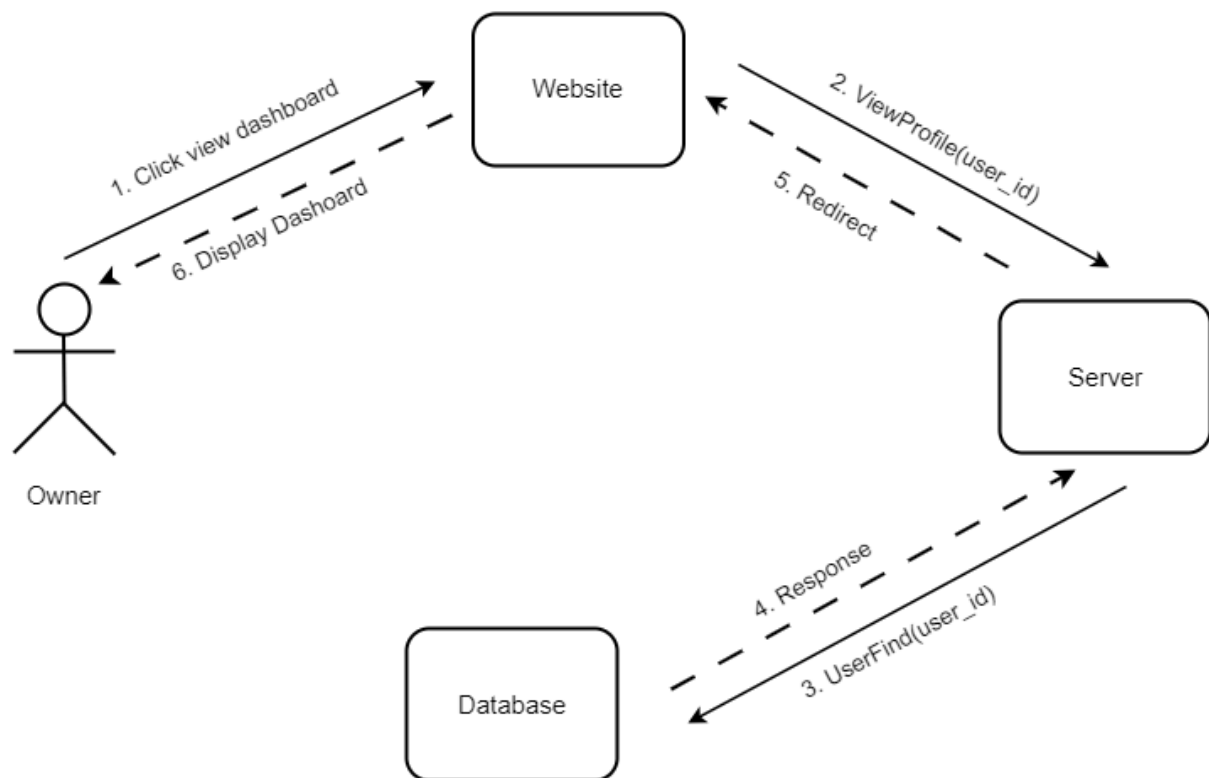
1. UC_Interface:



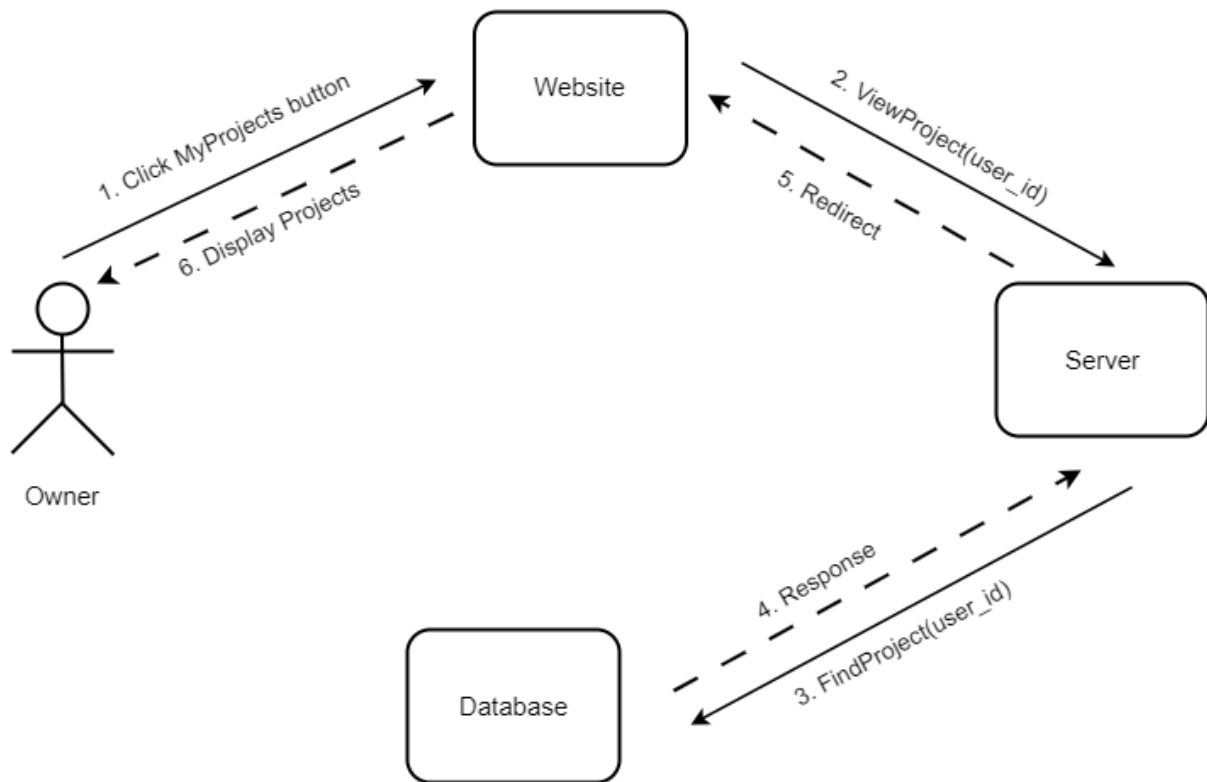
2. UC_Login:



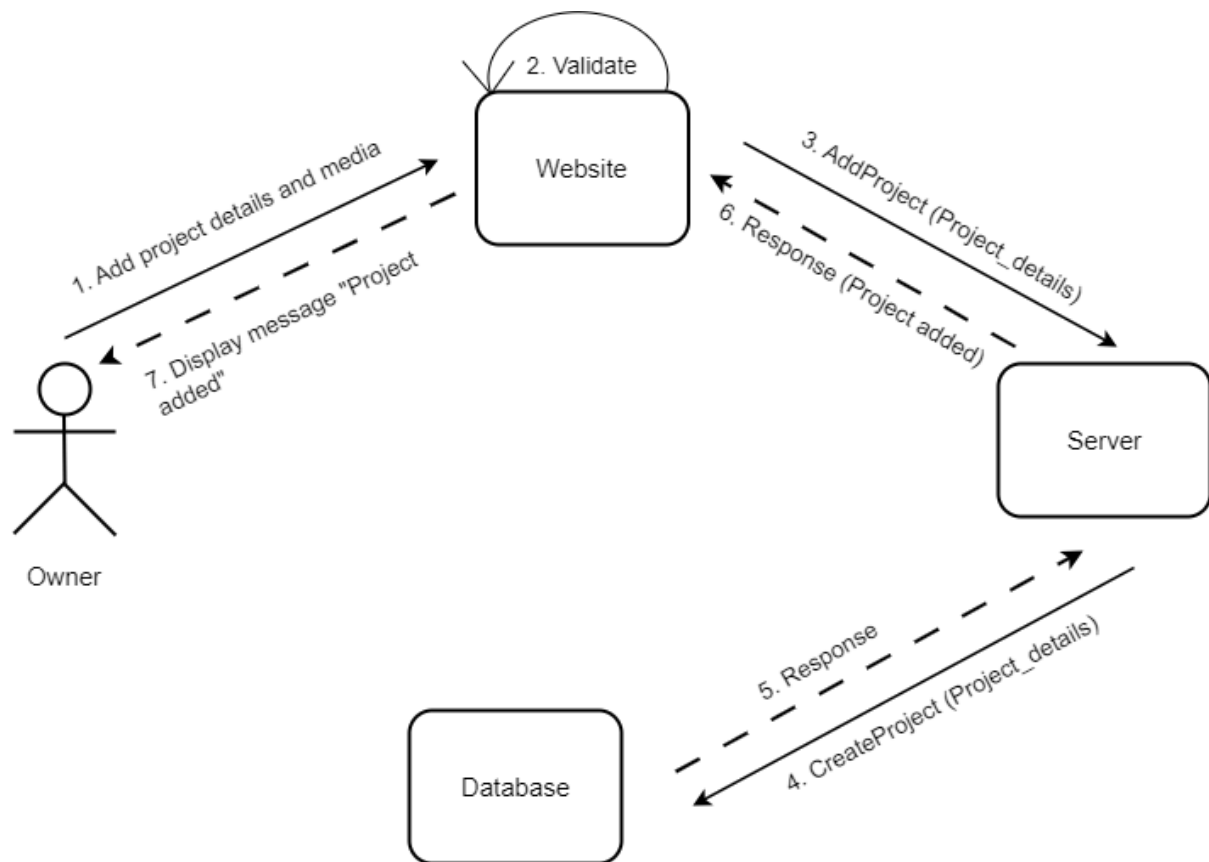
3. UC_Owner_Dashboard:



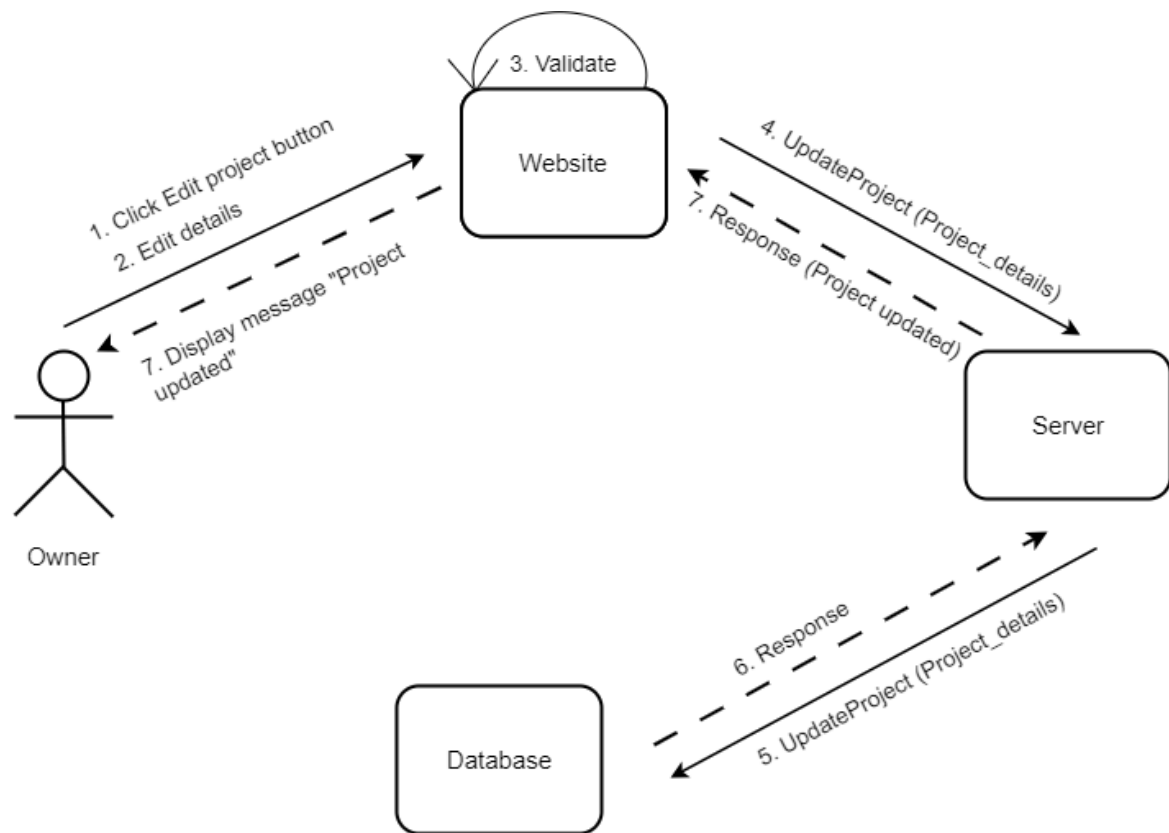
4. UC_Owner_Projects:



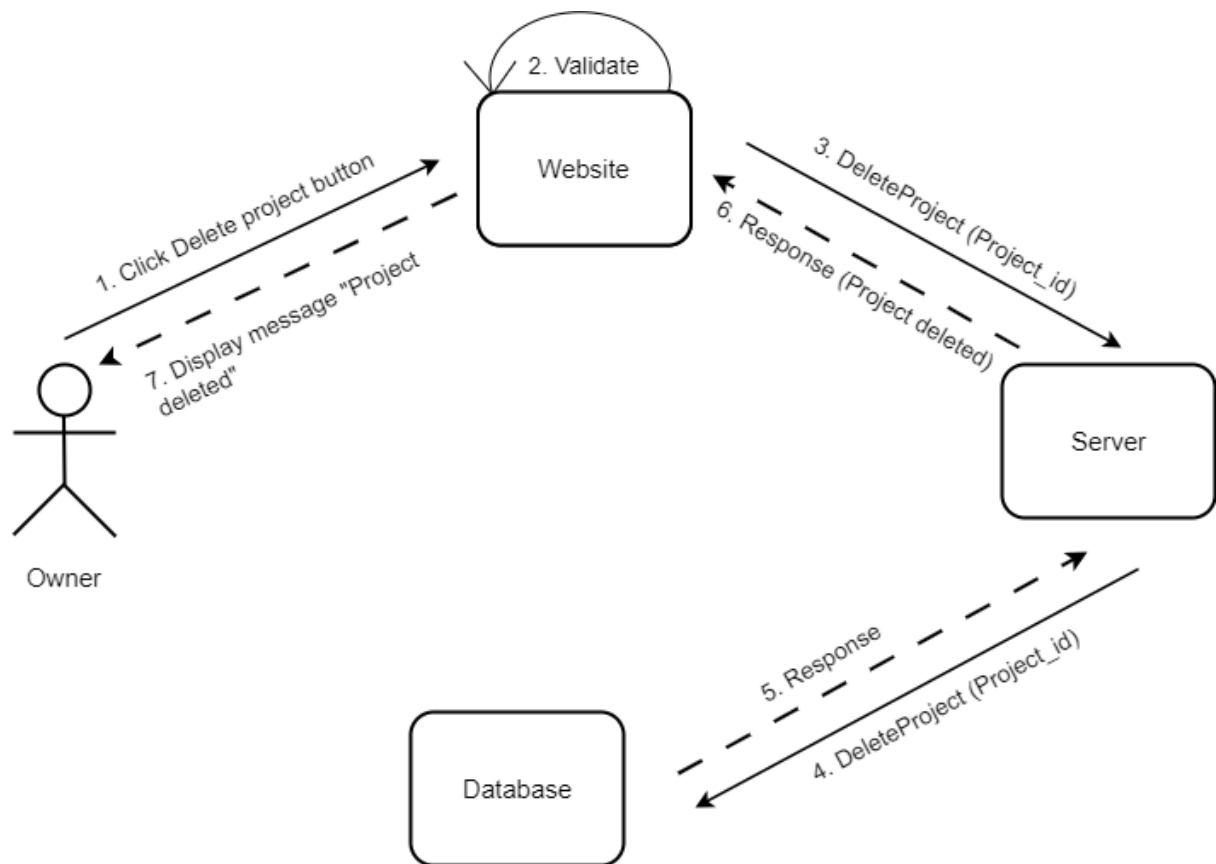
5. UC_ProjectCreation:



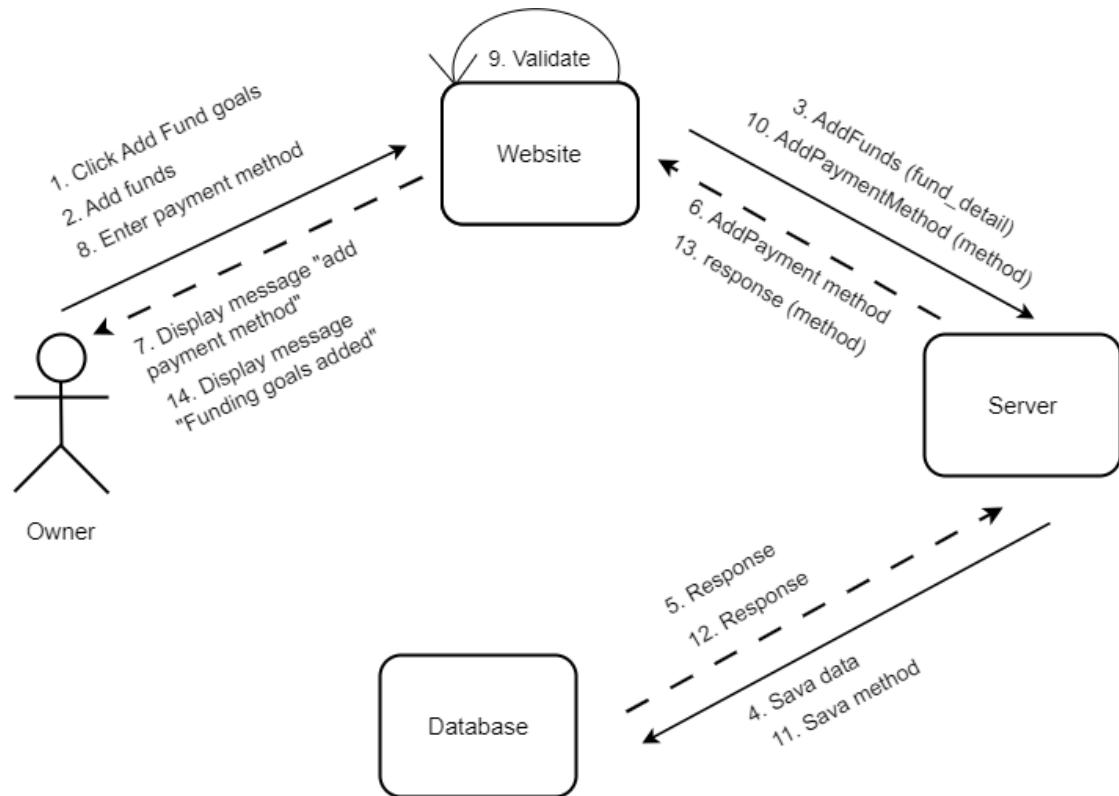
6. UC_UpdateProject:



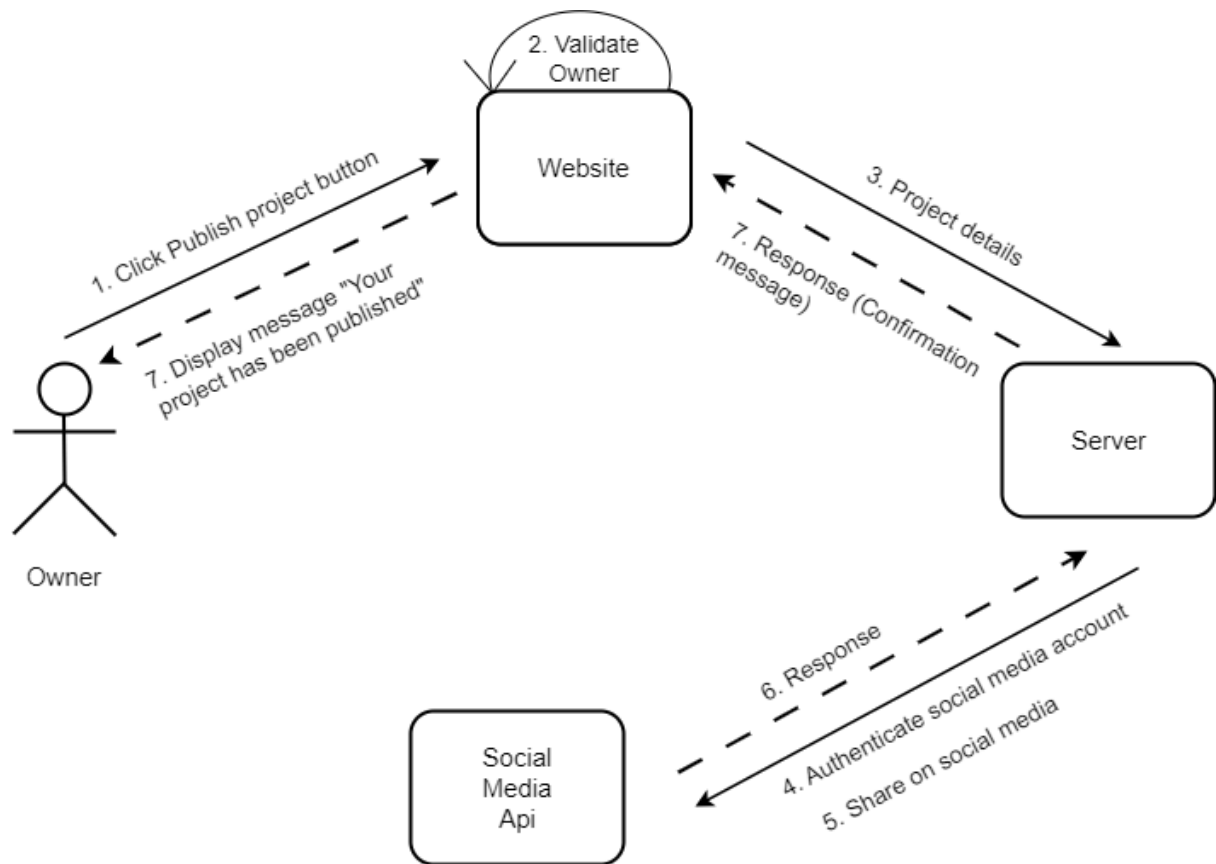
7. UC_DeleteProject:



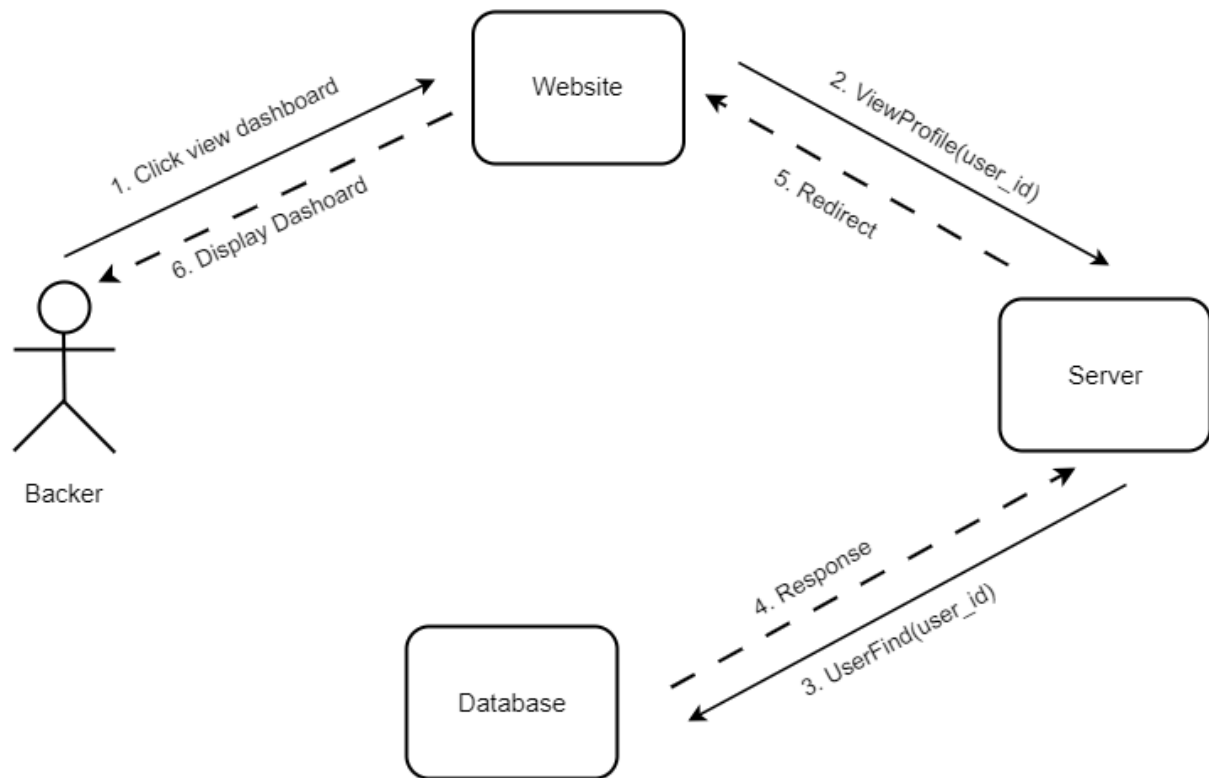
8. UC_Funding_Plan:



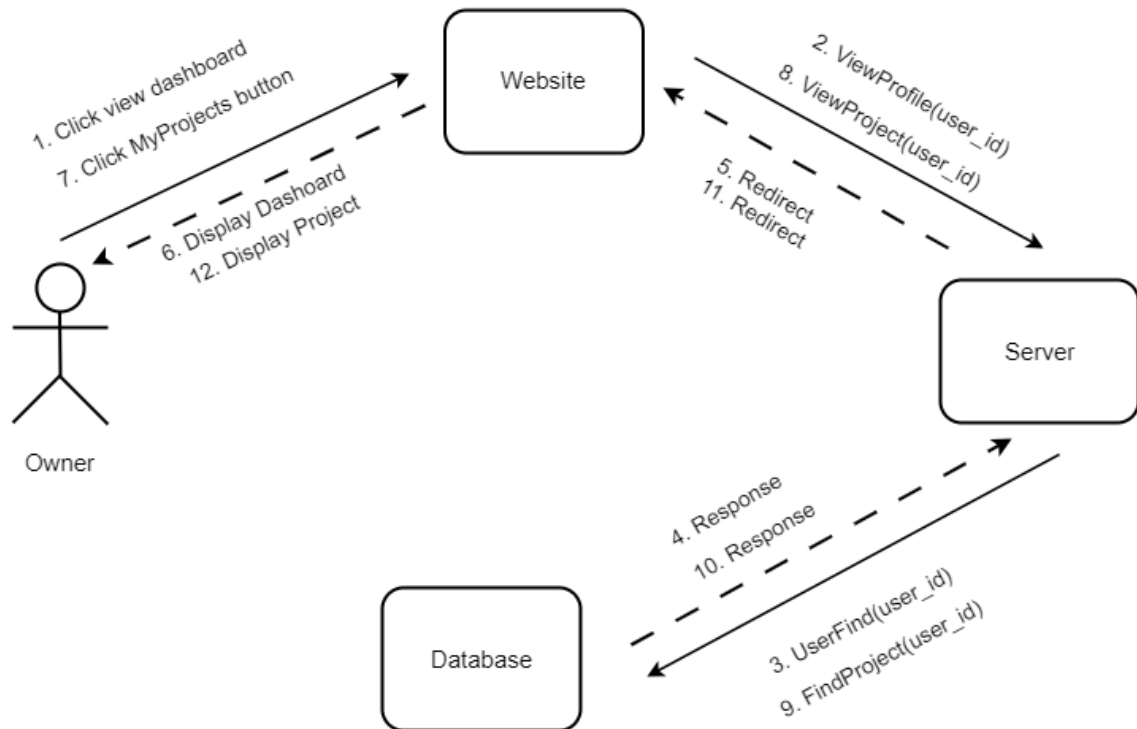
9. UC_Publish:



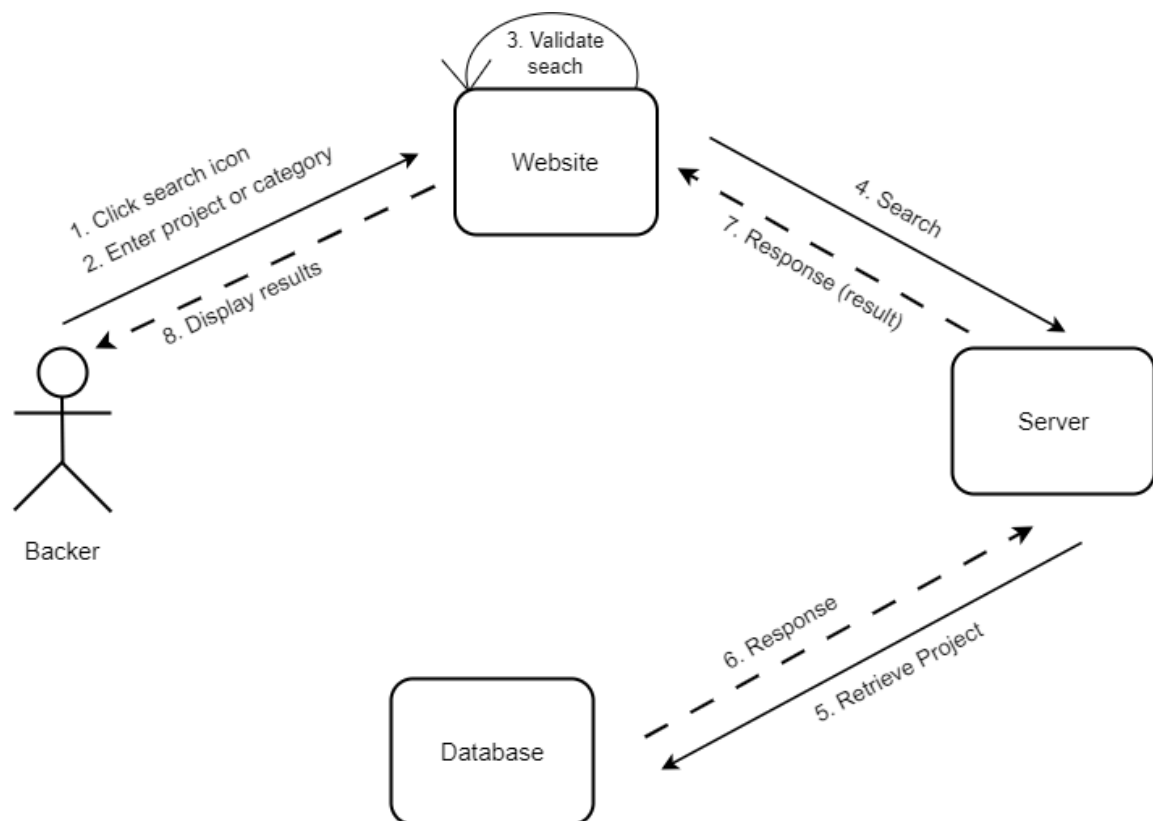
10. UC_Backter_Dashboard:



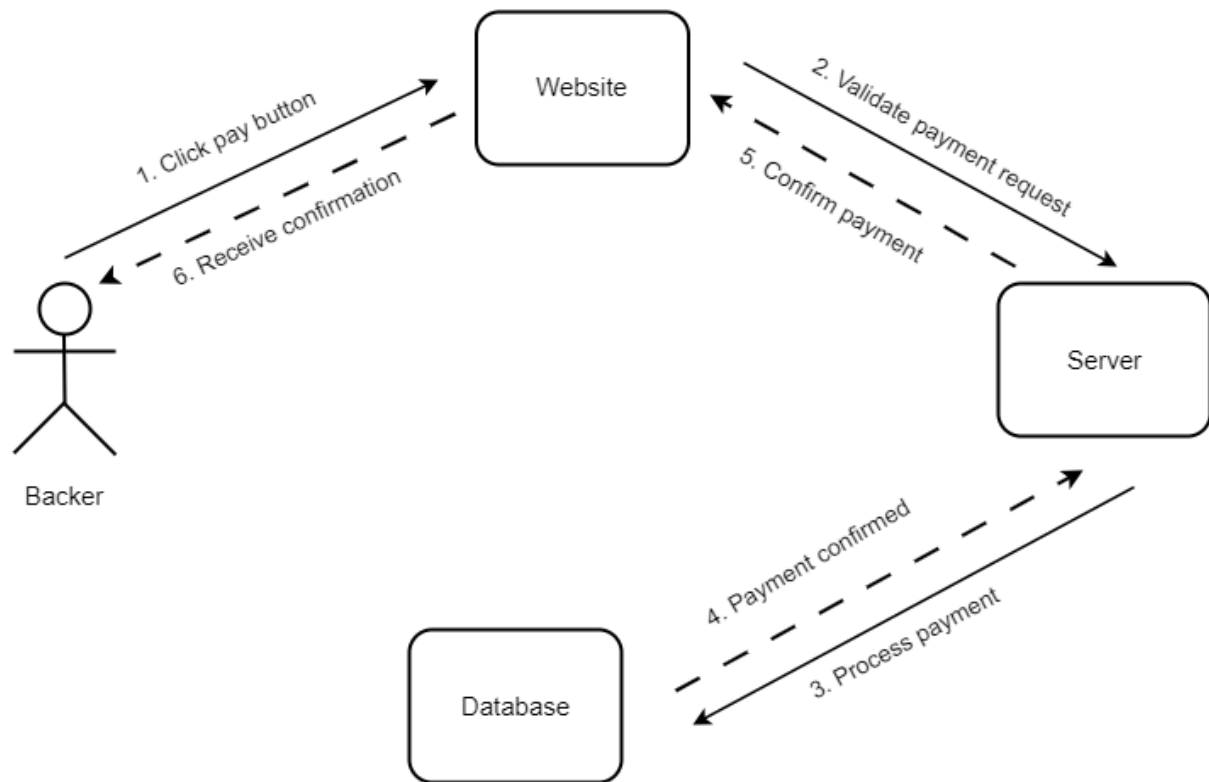
11. UC_Overlook:



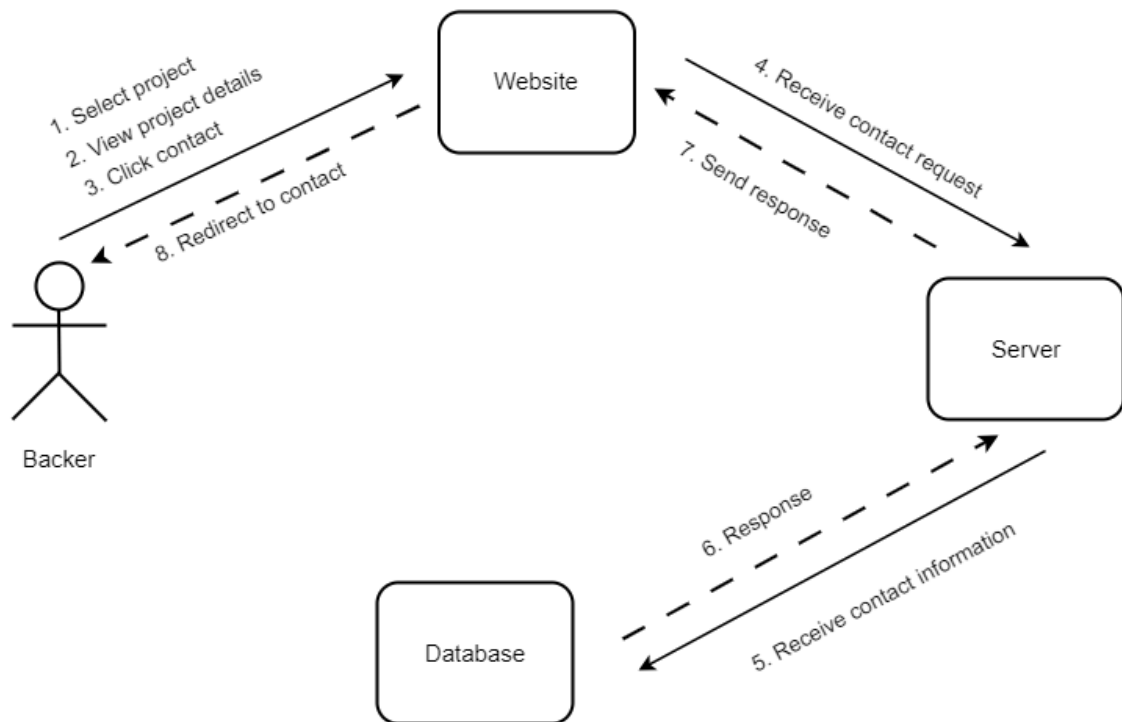
12. Search:



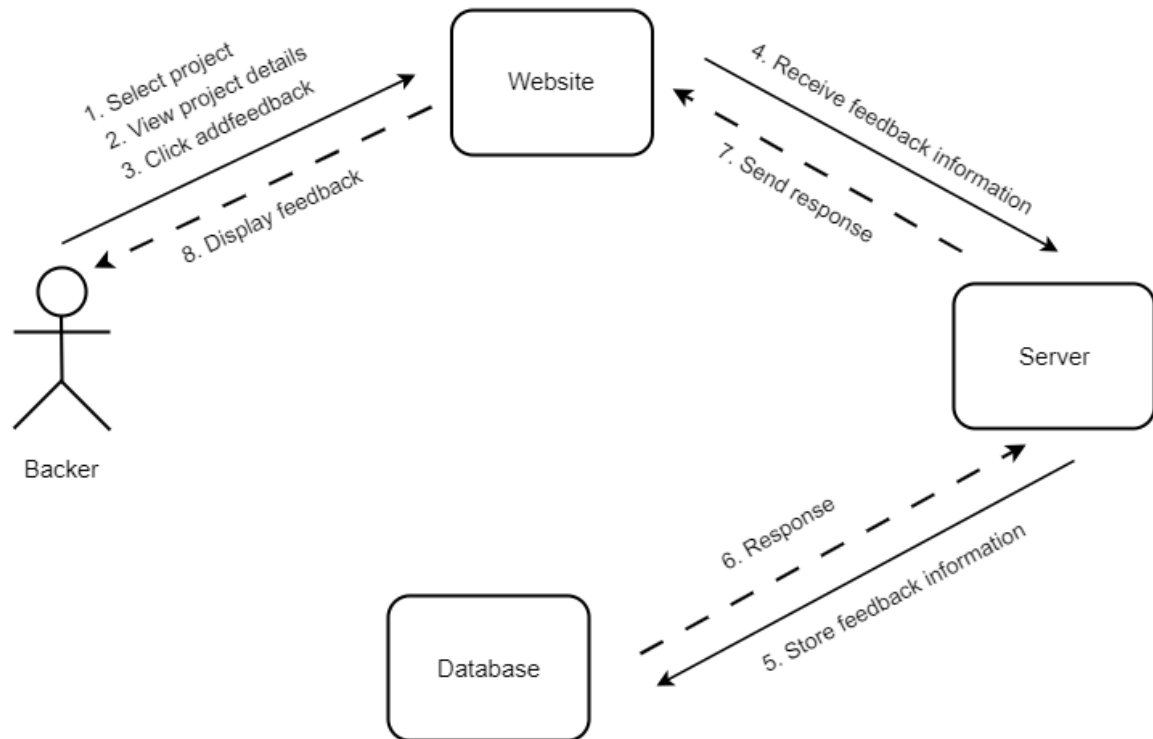
13. Secure_Payment:



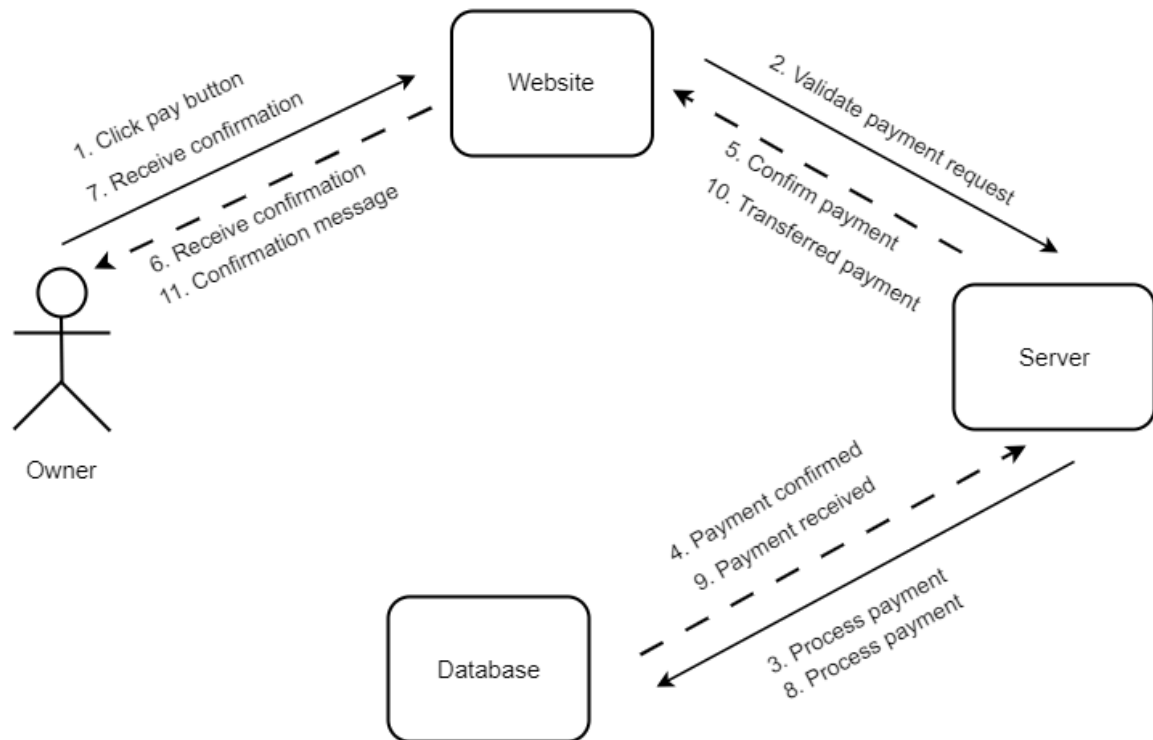
14. UC_Contact:



15. UC_Feedback:



16. UC_Receive_Payment:



Operation Contracts

Operation Contract no: 1

Name:

Interface

Responsibilities:

The owner has an easy-to-use interface to create and manage campaigns.

Cross-Reference:

UC_Interface

Exceptions:

- The user faces network-related problems while interacting with the interface.
- The server hosting the interface experiences high traffic or overload.

Pre-Conditions:

- The website is accessible and functional without any errors or technical failures.
- The user must have login.

Post Condition:

- The owner successfully creates and manages campaigns using an intuitive interface.

Operation Contract no: 2**Name:**

Project Creation

Responsibilities:

This enables owners to create and display projects.

Cross-Reference:

UC_ProjectCreation

Exceptions:

- The owner provides incomplete or incorrect information during the campaign creation process.
- The website experiences technical errors or failures during the campaign creation.

Pre-Conditions:

- The website is operational without any system issues.

Post Condition:

- A new project is successfully created and displayed on the platform.

Operation Contract no: 3

Name:

Funding Plan

Responsibilities:

This enables owners to set clear funding goals for their projects

Cross-Reference:

UC_Funding_Plan

Exceptions:

- The owner attempts to set an invalid funding goal (e.g., negative amount, non-numeric input)
- The user experiences network-related issues that prevent the successful submission of the funding goal.

Pre-Conditions:

- The website is operational without any system issues.

Post Condition:

- The funding goal is successfully defined for the project.

Operation Contract no: 4

Name:

Owner Dashboard

Responsibilities:

It provides owners with a dashboard to overview campaign progress.

Cross-Reference:

UC_Owner_Dashboard

Exceptions:

- Owners experience network connectivity issues that prevent them from accessing the dashboard.
- The system encounters difficulties in retrieving campaign progress data for display on the dashboard.

Pre-Conditions:

- The website is functional without any technical issues.

Post Condition:

- The owner can effectively monitor and track campaign progress through the dashboard.

Operation Contract no: 5

Name:

Publish

Responsibilities:

It facilitates project owners to share their projects on various social media platforms.

Cross-Reference:

UC_Publish

Exceptions:

- The user encounters authentication issues when attempting to connect or authenticate with a selected social media platform.
- The selected social media platform is temporarily unavailable or undergoing maintenance.
- The user decides to cancel the publishing process before completion.

Pre-Conditions:

- The website is operational and accessible to project owners.

Post Condition:

- The project owner successfully shares their project on the selected social media platform(s), enhancing visibility and outreach for the crowdfunding campaign.

Operation Contract no: 6

Name:

Overlook

Responsibilities:

It enables backers to have a comprehensive overview of ongoing crowdfunding campaigns available on the platform.

Cross-Reference:

UC_Overlook

Exceptions:

- The system encounters difficulties in retrieving information about ongoing crowdfunding campaigns.
- Backers experience network-related issues that prevent them from accessing the crowdfunding campaign overview.

Pre-Conditions:

- The website is operational and accessible to backers.

Post Condition:

- The backer gains a comprehensive overview of the ongoing crowdfunding campaigns, allowing them to explore and select campaigns they might be interested in supporting.

Operation Contract no: 7

Name:

Secure Payment

Responsibilities:

It enables backers to contribute funds securely to the selected crowdfunding campaigns.

Cross-Reference:

UC_Secure_Payment

Exceptions:

- The payment authorization process fails due to issues with the backer's payment method or insufficient funds.
- Unauthorised users attempt to make payments on behalf of a backer.
- Backers experience network-related issues that prevent them from completing the secure payment process.

Pre-Conditions:

- The backer is logged into their account on the crowdfunding platform.

Post Condition:

- The backer successfully contributes funds to the selected campaign through a secure payment system, supporting the campaign's funding goals.

Operation Contract no: 8**Name:**

Contact

Responsibilities:

It enables backers to communicate directly with the project creator or campaign owner.

Cross-Reference:

UC_Contact

Exceptions:

- The payment authorization process fails due to issues with the backer's payment method or insufficient funds.
- Unauthorised users attempt to make payments on behalf of a backer.
- Backers experience network-related issues that prevent them from completing the secure payment process.

Pre-Conditions:

- The backer is logged into their account on the crowdfunding platform.

Post Condition:

- The backer initiates communication with the project creator, allowing inquiries, clarifications, or discussions related to the project.

Operation Contract no: 9**Name:**

Search

Responsibilities:

It allows backers to discover a diverse range of crowdfunding projects available on the platform.

Cross-Reference:

UC_Search

Exceptions:

- The search returns no results based on the backer's criteria.
- An unexpected technical issue occurs while processing the search request.

Pre-Conditions:

- The backer is logged into their account on the crowdfunding platform.

Post Condition:

- The backer discovers a diverse range of projects based on their search criteria, enabling exploration and selection of interesting projects.

Operation Contract no: 10**Name:**

Feedback

Responsibilities:

It allows backers to provide feedback regarding their experiences or opinions about the crowdfunding platform or specific campaigns.

Cross-Reference:

UC_Feedback

Exceptions:

- Backers experience network-related issues preventing them from submitting feedback.
- An unexpected technical issue occurs while processing the feedback submission.

Pre-Conditions:

- The backer is logged into their account on the crowdfunding platform.

Post Condition:

- The backer successfully provides feedback, contributing to the platform's enhancement or addressing concerns related to campaigns or the platform itself.

Operation Contract no: 11

Name:

Login

Responsibilities:

It enables users to access the platform by logging into their respective accounts.

Cross-Reference:

UC_Login

Exceptions:

- The user enters incorrect login credentials (username or password).
- After multiple unsuccessful login attempts, the user's account is temporarily locked for security reasons.

Pre-Conditions:

- The user navigates to the login page of the crowdfunding platform.

Post Condition:

- The user successfully logs in to their account, gaining access to the platform's features and functionalities.

Operation Contract no: 12

Name:

Receive Payment

Responsibilities:

It enables project owners or campaign creators to receive funds that have been cleared by backers.

Cross-Reference:

UC_Receive_Payment

Exceptions:

- There is a delay in processing payments, and funds are not immediately available to the project owner.
- The payment authorization process fails due to issues with the backer's payment method or insufficient funds.

Pre-Conditions:

- The campaign reaches its funding goal, and the backer's payments are processed and cleared.

Post Condition:

- The project owner receives the cleared funds from the backers, facilitating the implementation or execution of the funded project or campaign.

Operation Contract no: 13

Name:

Backer Dashboard

Responsibilities:

It provides a dashboard interface for backers to overview the progress and details of the campaigns they have backed.

Cross-Reference:

UC_Backer_Dashboard

Exceptions:

- The backer experiences network-related issues preventing the successful loading of the dashboard.
- There is a delay in updating the status of backed campaigns in the dashboard (e.g., funding progress, project updates).

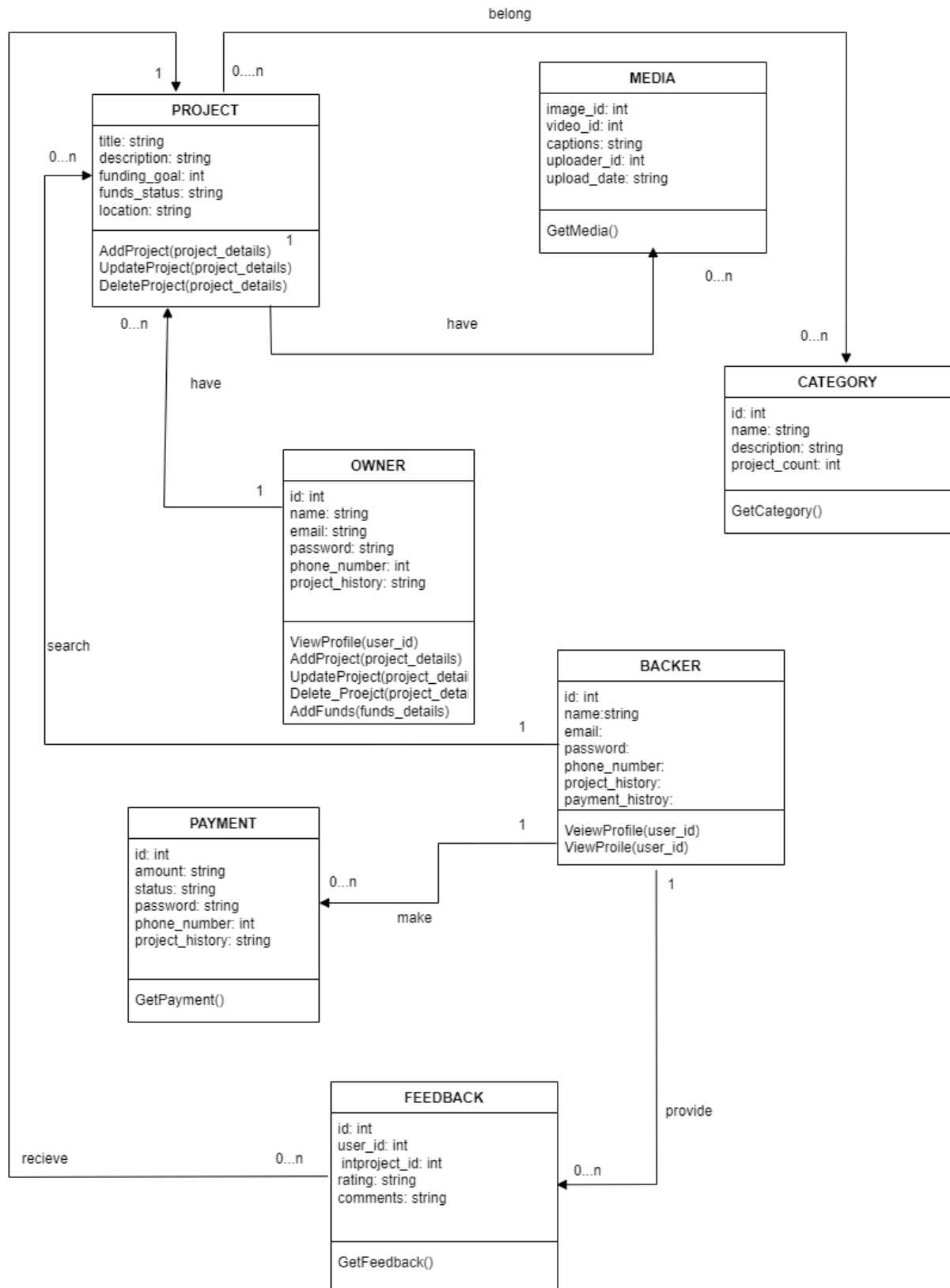
Pre-Conditions:

- The backer is logged into their account on the crowdfunding platform.

Post Condition:

- The backer successfully views an overview of the campaigns they have backed through the dashboard, enabling them to track the progress and status of their supported projects.

1.7 Design Class Diagram



1.8 Data Model

