**Project Title : Connectify**

Prepared by:

Abdul Munim (22i-1080)

Talha Azim (22i-1243)

Hadi Saleemi (22i-1043)

Software Requirments and Design Documents

Table of Contents

Table of Contents ii

1. Introduction 1

1.1 Purpose 1

1.2 Product Scope 1

1.3 Title 1

1.4 Objectives 1

1.5 Problem Statement 1

2. Overall Description 1

2.1 Product Perspective 1

2.2 Product Functions 2

2.3 List of Use Cases 2

2.4 Extended Use Cases 2

2.5 Use Case Diagram 2

3. Other Nonfunctional Requirements 2

3.1 Performance Requirements 2

3.2 Safety Requirements 2

3.3 Security Requirements 2

3.4 Software Quality Attributes 2

3.5 Business Rules 3

3.6 Operating Environment 3

3.7 User Interfaces 3

4. Domain Model 3

5. System Sequence Diagram 3

6. Sequence Diagram 3

7. Class Diagram 4

8. Package Diagram 4

9. Deployment Diagram 4

# Introduction

## Purpose

The purpose of this document is to outline the software requirements for the Connectify application, specifically version 1.0. Connectify aims to provide a platform for startups to connect, grow, and succeed in their business endeavors. It describes the core functionalities, including networking, marketing support, idea refinement, and event participation. This document focuses on the entirety of the system and its subsystems, providing a detailed understanding of how it addresses the challenges faced by startups globally.

## Product Scope

**Connectify** is a global platform tailored for startups at the early stages of their business journeys. Its primary purpose is to simplify the process of networking, marketing, and collaboration for startups, providing them with tools and opportunities to thrive. The application offers a cohesive ecosystem where startups can:

* Discover and connect with similar startups.
* Establish meaningful connections with larger companies.
* Participate in global events to showcase products and network.

## Title

Connectify: Build Tomorrow’s Company

## Objectives

The primary objectives of Connectify are:

1. **Facilitate Networking**: Enable startups to connect with each other and larger companies.
2. **Marketing Support**: Provide affordable tools to create marketing assets and strategies.
3. **Event Participation**: Organize events for startups to present their products to key stakeholders.
4. **Goal-Setting Assistance**: Offer guidance to set and achieve realistic business goals.
5. **Encourage Innovation**: Promote creativity and collaboration within a competitive market.

## Problem Statement

Startups often face significant challenges during their initial stages:

* Difficulty connecting with similar startups and larger companies due to limited networks.
* Lack of resources to market their ideas effectively, and strategic planning.
* Time constraints and limited access to research about other startups in their region or industry.
* Limited platforms for showcasing their work to influential stakeholders.

**Connectify** was created to eliminate these obstacles. The app provides an integrated platform where startups can:

* Build meaningful connections with other startups and influential companies.
* Simplify marketing efforts using in-built tools.
* Showcase their products and network at high-profile events.
* Collaboratively set goals, and grow their business efficiently.

# Overall Description

## Product Perspective

Connectify is a **new, self-contained platform** designed to be a comprehensive tool for startup growth. It does not rely on any existing systems but aims to integrate smoothly with social media platforms, business databases, and event management systems. Its features are structured to cater to the lifecycle of startups, from ideation to scaling.

Key system components include:

* **Startup Networking Module**: Facilitates connections based on industry, and mutual interests.
* **Event Manager**: Schedules and manages participation in global events.

## Product Functions

Connectify’s major functions include:

* User registration and profile creation.
* Intelligent networking recommendations.
* Event notifications and participation management.
* Collaborative workspaces for business idea improvement.
* Analytics for goal tracking and progress measurement.

## List of Use Cases

## Extended Use Cases

## Use Case Diagram

A diagram of a person's structure

Description automatically generated

A diagram of a diagram

Description automatically generated

# 

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## 

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

## Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

# Domain Model

# System Sequence Diagram

# Sequence Diagram

# 

# Class Diagram

# Component Diagram

# Package Diagram

# Deployment Diagram

**User**

**Attributes**: user\_ID (unique identifier), name, email, password.

**Methods**: ReleaseNotification() (sends notifications).

**Relationships**: One-to-many with ChatBox, Meeting, Notification. Interacts with StartUp and Events.

**StartUp**

**Attributes**: industry (domain), status (e.g., active).

**Methods**: AddAchievement(), AddFinancialData(), AddService() (manage startup data), Search() (find connections/resources).

**Relationships**: One-to-one with Financial Data and Goals. Uses Service and Achievement classes.

**Company**

**Attributes**: name, industry.

**Methods**:Announce() (public announcements), CreateNewEvent() (schedule events).

**Relationships**:Linked to Events.

**Financial Data**

**Attributes**:current\_income, current\_expenses, profit/loss, progress\_graph.

**Methods**:ShowGraph(), SaveData(), GenerateReport() (insights and summaries).

**Relationships**:Linked to multiple Financial Entry records.

**Financial Entry**

**Attributes**: entry\_ID, amount, type, reason.

**Methods**: AddData() (track transactions).

**Relationships**: Many-to-one with Financial Data.

**ChatBox**

**Attributes**: chatter (participants).

**Methods**: SpecifyChat(), SendMessage() (handle chats).

**Relationships**: One-to-many with Conversation.

**Conversation**

**Attributes**: conversation\_ID.

**Methods**: AddMessage() (manage messages).

**Relationships**: One-to-many with Message.

**Message**

**Attributes**: message\_ID, content, time\_stamp.

**Methods**: AddMsg() (add messages).

**Goals**

**Attributes**: goal\_ID, description, status.

**Methods**: SetGoal() , (manage goals).

**Relationships**: Linked to StartUp.

**Events**

**Attributes**: event\_ID, name, date, location.

**Methods**: ScheduleEvent() (manage event details).

**Relationships**: Connected to User and Company.

**Job**

**Attributes**: job\_ID, title, category, salary, date\_posted.

**Methods**: AddApplication() (add applications).

**Relationships**: Linked to Job Application.

**Job Application**

**Attributes**: jobApp\_ID, user\_ID, job\_ID.

**Methods**: AddApplication() (process applications).

**Achievement**

**Attributes**: achievement\_ID, description, date.

**Methods**: AddAchievement() (log achievements).

**Relationships**: Linked to StartUp.

**Service**

**Attributes**: service\_ID, name, category, base\_price.

**Methods**: GetDetails() (fetch service info), FindServices() (search services).

**Notification**

**Attributes**: notification\_ID, message, type.

**Methods**: AddNotification() (send notifications).

**Relationships**: Linked to User.

This summary condenses the relationships, attributes, and methods of the classes while maintaining clarity.