|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Software Design Specifications**  ***Event Craft***  **Version: [1.0]**   |  |  | | --- | --- | | Project Code | S24-100 | | Supervisor | Mr. Farooq Zaidi | | Co Supervisor | Fahad Hussain | | Project Team | 20k1672 Zain ul Abdin  20k1038 Qazi Zain  19k1066 Ali Khaf ul wara | | Submission Date | 16th May 2024 | |   **Document History**  *[Revision history will be maintained to keep a track of changes done by anyone in the document.]*   |  |  |  |  | | --- | --- | --- | --- | | Version | Name of Person | Date | Description of change | | 1.0 | Zain | 28 Apr | Started Working on the Document | | 1.1 | Qazi Zain | 1 May | Added Further Details | | 1.2 | Qazi Zain | 3 May | Added Data Dictionaries | | 1.3 | Ali Khaf | 5 May | Added Software Level Architecture | | 1.4 | Zain | 6 May | Added ERD and Sequence Diagram | | 1.5 | Ali Khaf | 8 May | Reviewed the whole document, proofreading and made a few required changes |         **Distribution List**   |  |  | | --- | --- | | **Name** | **Role** | | Farooq Zaidi | Supervisor | | Fahad Hussain | Co Supervisor | |  |  |       **Document Sign-Off**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Version** | **Sign-off Authority** | **Project Role** | **Signature** | **Sign-off Date** | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |

**Document Information**

|  |  |
| --- | --- |
| **Category** | **Information** |
| Customer | FAST-NU |
| Project | Event Craft |
| Document | Software Design Specification |
| Document Version | 1.5 |
| Status | Draft |
| Author(s) | Zain ul Abdin, Qazi Zain, Ali Khaf |
| Approver(s) |  |
| Issue Date | 8 May 2024 |
| Document Location | FAST NUCES, Main Campus, Karachi |
| Distribution | Advisor  Project Coordinator’s Office (through Advisor) |

**Definition of Terms, Acronyms and Abbreviations**

|  |  |
| --- | --- |
| **Term** | **Description** |
| ASP | Active Server Pages |
| DD | Design Specification |
| Teacher | Used Interchangeably with Faculty in the SRS |
| Faculty | Used Interchangeably with Teacher in the SRS |
|  |  |
|  |  |
|  |  |
|  |  |

**Table of Contents**

Table of Contents

[1 Introduction 8](#_Toc154659657)

[1.1 Purpose of Document 8](#_Toc154659658)

[1.2 Intended Audience 8](#_Toc154659659)

[1.3 Document Convention 8](#_Toc154659660)

[1.4 Project Overview 8](#_Toc154659661)

[1.5 Scope 9](#_Toc154659662)

[2 Design Considerations 10](#_Toc154659663)

[2.1 Assumptions and Dependencies 11](#_Toc154659664)

[2.2 Risks and Volatile Areas 11](#_Toc154659665)

[3 System Architecture 12](#_Toc154659666)

[3.1 System Level Architecture 12](#_Toc154659667)

[3.2 Software Architecture 13](#_Toc154659668)

[4 Design Strategy 14](#_Toc154659669)

[5 Detailed System Design 15](#_Toc154659670)

[5.1 GUI Design 15](#_Toc154659671)

[5.2 Database Design 17](#_Toc154659672)

[5.2.1 ER Diagram 17](#_Toc154659673)

[5.2.2 Data Dictionary 18](#_Toc154659674)

[5.3 Application Design 23](#_Toc154659675)

[5.3.1 Sequence Diagram 23](#_Toc154659676)

[5.3.2 State Diagram 29](#_Toc154659677)

[1 References 31](#_Toc154659678)

[2 Appendices 32](#_Toc154659679)

# Introduction

## Purpose of Document

The purpose of this Software Design Specification (SDS) document is to give detailed description and outline the functional and nonfunctional requirement of “Event Craft”. Furthermore, this document includes goal, scope, objective, and system functionalities of system.

## Intended Audience

The intended audience for this document includes Developers, Manager, Quality Assurance, Service Providers , users and anyone who involved in the development, quality assurance and deployment process of this system.

## Document Convention

Fonts Styles: Arial for heading and subheadings, Calibri content table.

Font Size: 12 for sub-headings,10 for description, and 16 headings.

## Project Overview

Event Craft is a comprehensive event management system designed to simplify the process of planning and organizing events, ranging from personal celebrations to professional gatherings. With a user-friendly interface and a wide range of features, Event Craft aims to revolutionize the event planning industry by offering a seamless platform that enhances the experience for both event organizers and participants.

scams and monitor the environment.

## Scope

The scope of our project is to encompasses the development of an extensive event management system, our aim to streamline the process which will be held during the organizing an events. It can be birthday or wedding reception. The main idea develops this system is to create use friendly HCI base application which

can manage all the aspect of event. Users will be able to login to their accounts and utilize features such as time scheduling and checklist management to effectively plan their events. This system will also facilitate collaboration by allowing users to designate helpers who is helping to organized an event, by assign tasks, ensuring coordination throughout the planning process and organizing process.Furthermore, this system will offer recommendations for event planning based on budget constraints, time and depending on area. System will allow user to guest management features which will listed guests, and send them invitations. In Addition, this system will also provide AI-generated e-cards for event invitations. Furthermore, users will have access to a wide range of event components such as venues, catering, and decoration etc. On the other hand, Service providers will be able to customize their services within a framework. Service provider will receive notifications for payments and orders. Users and Service providers both can monitor their progress of an event.

# Design Considerations

For purposes of development and use, the system must always be dependable, efficient, adaptable, and flexible. In order to address the key problems that existed with the system's design on a more granular level, we spent a lot of time working toward a better design, UI, and most time for requirement gathering of project. Some of the basic design issues that we came up with are:

1. How might we interact with the database?
2. How can we control our different views between Service Provider and Users?
3. What kind of architecture do we want to base our front-end and back-end development ?
4. How we build User friendly application?
5. How can our users successfully interact with our system?
6. How do we authenticate users and service providers?
7. How we can make customized frameworks for service providers ?

There are other design considerations, like:

1. Since it is a web application, we care about compatibility on different browsers enough so that it does not break for our users.
2. Extensibility - The project is created in a way that makes it simple to add new modules with additional functionality without affecting or changing the contents of existing modules.
3. Modularity - Modules are designed to be well-defined, independent components which would lead to better maintainability. Dependencies are clearly defined where needed and are well-intentioned and well-contained.
4. Fault-tolerance - this factor is not really relevant because the program is not yet intended to extend outside local development environments.
5. Reusability – System will be developed through classes and modules which can be reused later on.
6. Performance- By developing a light weighted web app the performance of software will be enhanced.
7. Security - The program can withstand and fend against harmful influences and actions.
8. Usability – By implying the UI/US standards the user will be able to understand and learn the software more easily.
9. Portability - The program will work on many mobile, desktop, and laptop platforms.

In such a scenario, our initial steps were to,

1. Complete and choose the primary project objectives that we were going to work on user requirements analysis and engineering.
2. Work breakdown structure will be used to check what will be the distribution of the tasks.
3. Review the work as necessary.

For our project, we will be using the waterfall methodology. As our milestones are defined, requirements, and scope are well understood, there will be little to no change in our requirements. Hence, this method will enable us to handle design issues in accordance with our workflow and design and the required documentation that is necessary and help with features.

## Assumptions and Dependencies

Following are the assumption and dependences.

* We assume that users and service providers will have a proper internet connection for this event management system.
* Service providers are willing to participate and collaborate within the system.
* Users and Service Provider, both are responsible for providing accurate and up-to-date information to application.
* This system will be developed using Human Computer Interaction (HCI) rules and practices.
* Dependence with API of map, users contact list, gallery and notification turn on.

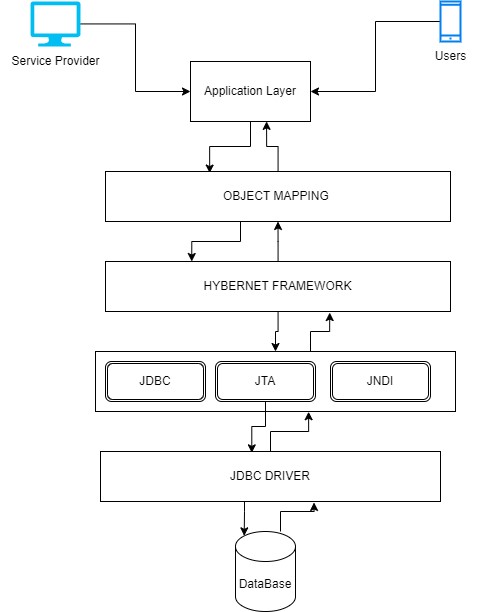
## Risks and Volatile Areas

The platform deals with sensitive user information such as personal details, event plans, and financial data. Risks associated with data breaches, unauthorized access, and data loss must be addressed through robust security measures, encryption protocols, and access control mechanisms.

Integrating with third-party services, such communication APIs, and mapping services.

Poor user experience, usability issues, and navigation difficulties can be occur . This problem can be resolved by using HCI principles

# System Architecture



## Software Architecture

The architecture on which the project will be created is Client-Server architecture where a proper web server will be created to address the request made through the application. Also, the project will be implemented by creating an application using the following technologies:

1. React, React Native (Front-End)
2. Spring Boot(Environment)
3. MySql (relatoional database)

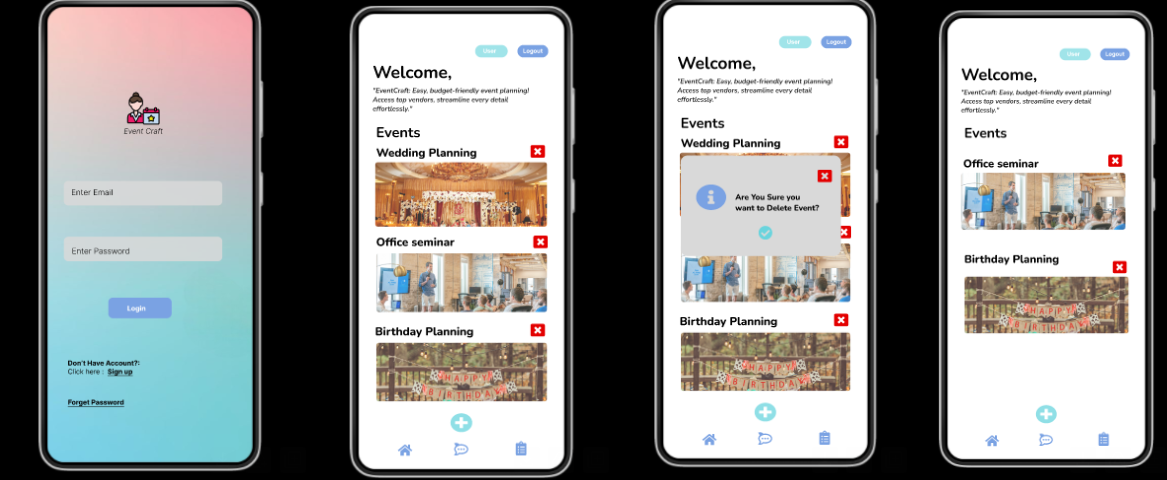
# Design Strategy

Our plan is to create a web app that’s easy to use and can grow with our users’ needs. We’re putting users first, making sure we really understand what they need through research and feedback. This helps us keep improving the app, solving real problems, and making it easier to use.

We are working in short Iteration of agile approach ,our approach based on planning , Analysis the problem , development or finding solutions and evaluation. This develop it easier to adapt as we go. Our design will work well on different devices and screens. Meanwhile, a scalable architecture our design, facilitating future growth and evolution without compromising performance or reliability. By designing modular, loosely coupled components, we minimize dependencies and enable seamless scalability as the application evolves.

# Detailed System Design

## GUI Design



A screenshot of a phone

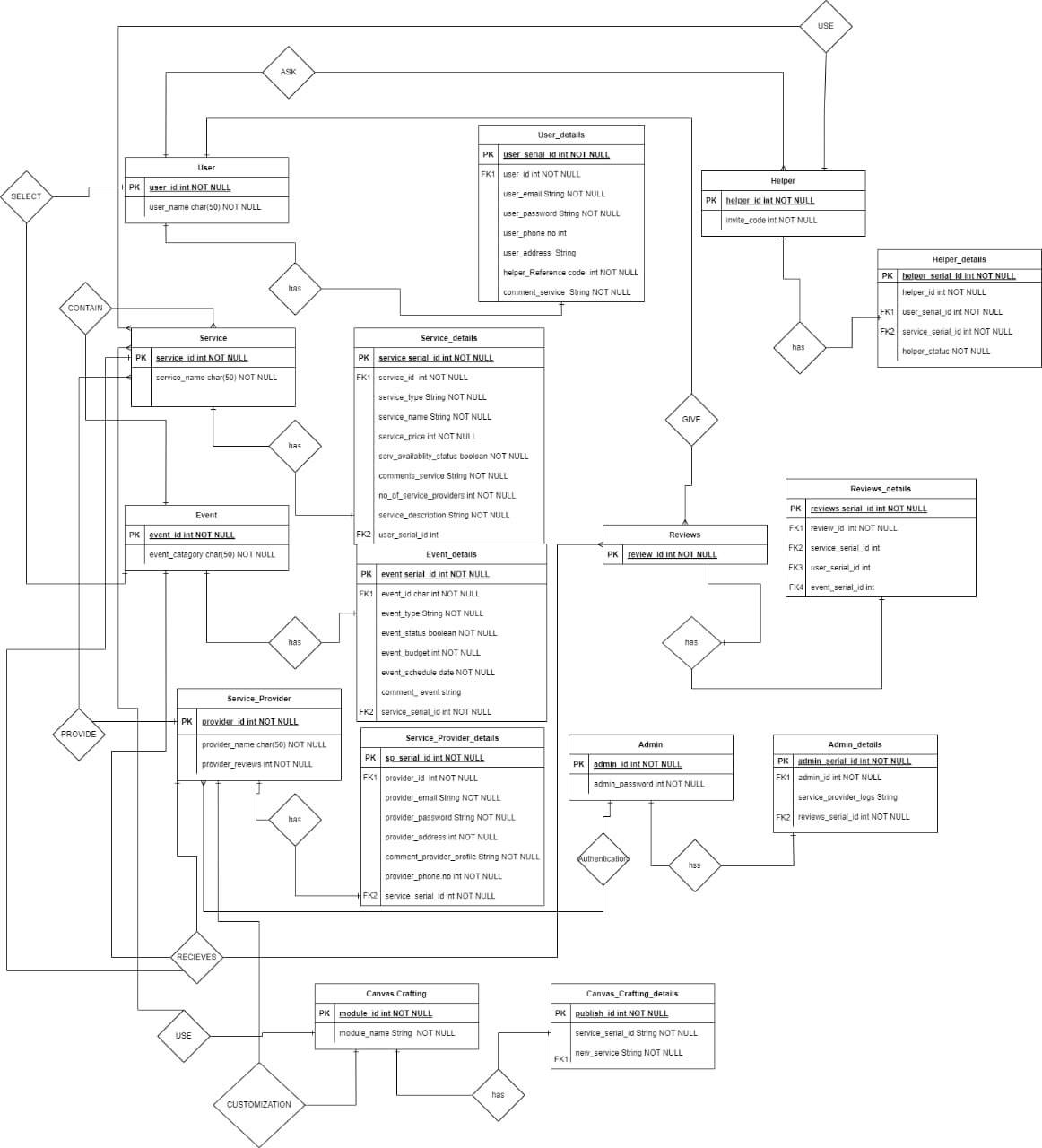
Description automatically generated

A screenshot of a phone

Description automatically generated

## Database Design

### ER Diagram



## Application Design

### Sequence Diagram

Sequence diagram is a type of UML diagram that illustrates how objects interact with each other in a specific sequence of events. It shows the flow of messages between objects over time, highlighting the order of interactions within a system.

#### Login

A diagram of a login screen

Description automatically generated

#### SignUp

A diagram of a software process

Description automatically generated

#### Forget Password

A diagram of a computer program

Description automatically generated

#### System Sequence Diagram

A diagram of a user flow

Description automatically generated

#### System Sequence Diagram of Service Provider

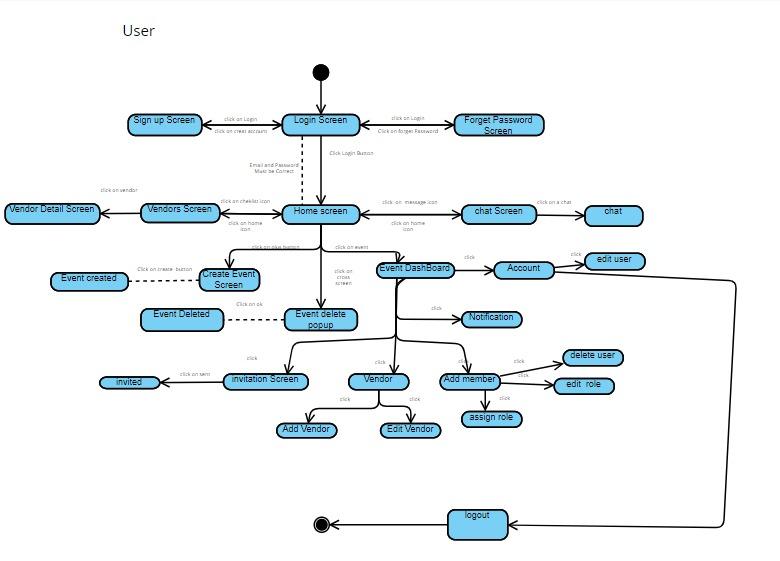
A screenshot of a computer screen

Description automatically generated

### State Diagram

The State machine diagram is a UML diagram that depicts the states an object can be in and the transitions between those states. It models the dynamic behavior of a system by showing how an object responds to various events and changes its state accordingly.

* + - 1. **User State Machine Diagram**



* + - 1. **Service Provide State Machine Diagram**

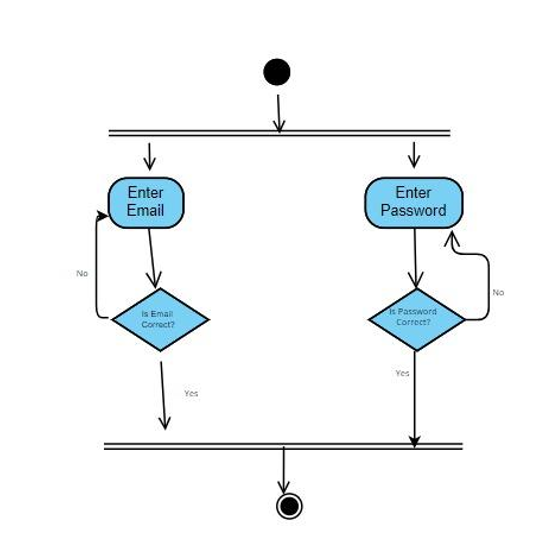
A diagram of a software company

Description automatically generated

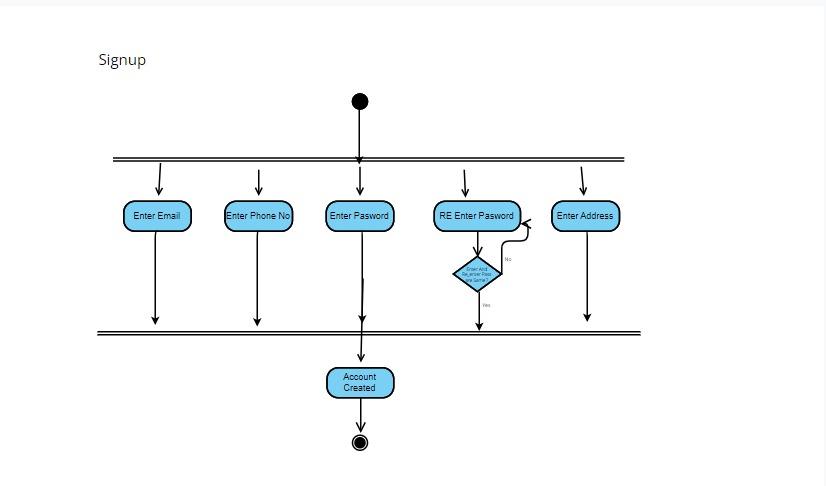
* + 1. **Activity Diagram**

Activity diagram is a type of UML diagram that represents the flow of activities or actions in a system or process. It highlights the sequence and conditions of tasks, illustrating dynamic aspects of the system or workflow.

* + - 1. **Login**

****

* + - 1. **Signup**

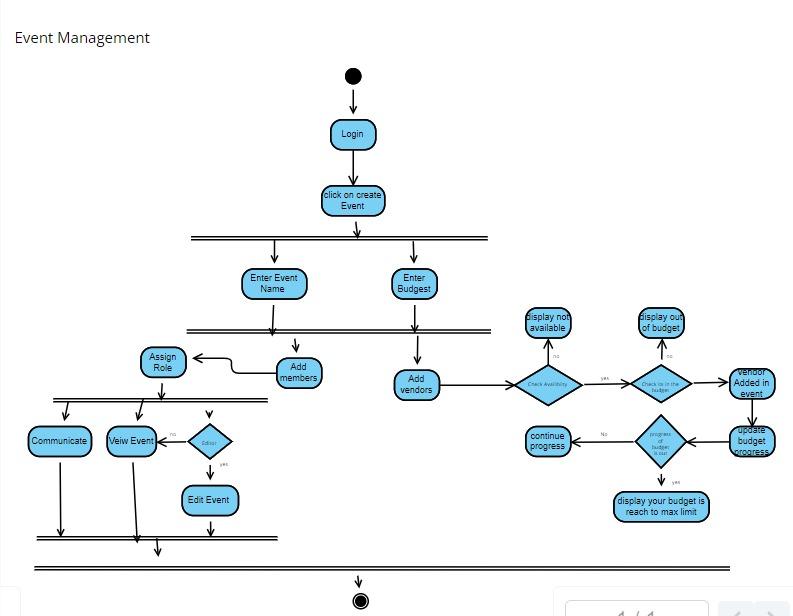


* + - 1. **Forget Password**

**A diagram of a computer process

Description automatically generated**

* + - 1. **Event Management Users**

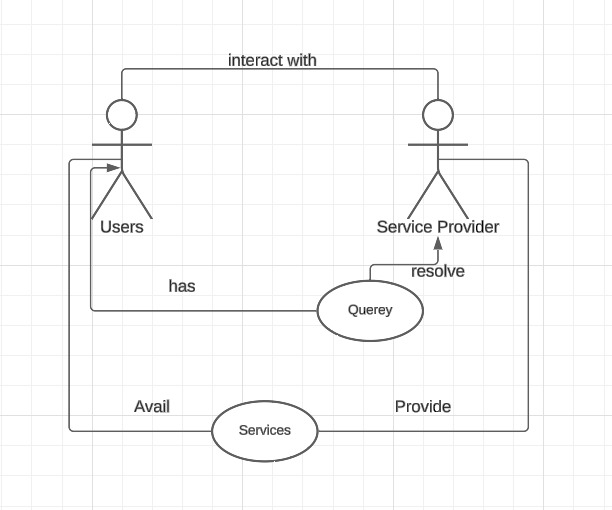


* + - 1. **Event Management Service Provider**

A diagram of a software development

Description automatically generated

# Conceptual Diagram



# References

[1] D. A. Norman, The design of everyday things. Basic Books, 2002.

# Appendices

*Not Applicable*