SponsorIN

Brandon Tiong

Emanuel Macias

Ege Keser

Table of Contents

Table of Contents2

1. Project Definition3
2. Project Requirements4

2.1 Functional Requirements5

2.2 Usability Requirements5

2.3 System Requirements6

2.4 Subsystem Architecture Design6

a. User Management Subsystem6

b. Communication Subsystem7

c. Admin Control Subsystem7

d. Company Subsystem8

e. Athlete Subsystem8

f. Search & Discovery Subsystem9

g. UI/UX Subsystem9

h. Backend Processing Subsystem10

i. Data Management Subsystem10

j. Security & Compliance Subsystem11

2.5 Subsystem – Functionality Diagrams12

3. Project Specification13

Type chapter level (level 2)5

Type chapter title (level 3)6

**1. Project Definition**

**Background**

Before 2021, the college athletes around the United States were not able to receive any financial support from their supporters or find a sponsorship because of the NIL (Name, Image and Likeness) restrictions of NCAA (National Collegiate Athletic Association). Due to the recent laws passed in many states, college athletes had the chance to get paid by their supporters as well as started to find sponsorships to build their brands using their own name. However, according to our research, there is no platform that would help college athletes to promote themselves and find a possible sponsorship connection. It’s easier for companies to find talented athletes and sponsor them due to their giant budgets. Unfortunately, that is not the case for athletes.

**What is SponsorIN and how will it help?**

SponsorIN will come into play when college athletes want to promote themselves and find sponsorships easily. SponsorIN is a web application to help college athletes and companies to find each other. SponsorIN will provide college athletes and companies an environment to build network, stay connected, and possibly, build a business relationship. College athletes will be able to promote themselves and look out for sponsorship, while companies are looking for athletes with great potential and help them grow their own brand under their sponsorship.

**How will it be developed?**

We will start developing SponsorIN by developing basic functions that most of the web applications would normally have. These functions include user registration, sign in – sign out and basic search. Then, our team will focus on developing the user specific functions such as sending-accepting-negotiating offers, athlete pool viewing-filtering-sorting, user verification and user editing-deletion. Once we have all the functions up and running, we will connect those functions with associated front-end and database sections so that we will have a complete web application that aims to accomplish our goals.

**2. Project Requirements**

**2.1 Functional Requirements:**

* **User Types**: Three different types of users—Admin, Company, and Athlete—with distinct permissions and functionalities.

**Common Features for All Users**

* **Login & Authentication**: Secure login process for Admins, Companies, and Athletes.
* **Profile Creation & Editing**: Customizable profiles to add and edit pertinent details.
* **Messaging System**: In-app secure messaging for communication between all user types.
* **User Search**: Ability to search for other users based on relevant criteria.

**Admin-Specific Features**

* **Profile Verification**: Exclusive rights to verify the profiles of Companies and Athletes.
* **User Editing & Deletion**: Capability to edit or delete any user profiles.

**Company-Specific Features**

* **Send Offers**: The ability to send sponsorship offers to Athletes.
* **Organized Watchlist**: A feature to save and organize a list of Athletes they are interested in.
* **Filtered Athlete Pool**: Advanced search and filters specific to finding suitable Athletes.

**Athlete-Specific Features**

* **Accept/Decline/Counter Offers**: The ability to interact with offers from Companies by accepting, declining, or sending a counteroffer.

A diagram of a software system

Description automatically generated

#### **2.2 Usability Requirements**

##### **User Interface**

* **Responsive Design**: The web app should be accessible and usable on various devices including smartphones, tablets, and desktops.
* **Intuitive Navigation**: Menus and sub-menus should be easy to understand and navigate.
* **Information Architecture**: Efficient categorization of information for easy retrieval and action.

##### **Performance**

* **Load Time**: The web application should load within 3 seconds to provide a seamless user experience.
* **Scalability**: The system should be able to handle a growing number of users and data.
* **Error Handling**: Graceful error messages and solutions should be provided for common issues.

#### **2.3 System Requirements**

##### **Hardware**

* Since the application is web-based, the hardware requirements are minimal; users need a device with internet connectivity. On the server side, robust cloud servers may be used for better scalability and performance.

##### **Software**

* **Back-End**: Python Flask for server-side logic.
* **Front-End**: HTML, CSS, JavaScript for UI/UX.
* **Web Server**: A server capable of running Python, such as Apache or Nginx.

##### **Database**

* **Database Engine**: PostgreSQL
* **Data Backup**: Regular automated backups.
* **Data Integrity**: ACID compliance to ensure data integrity.

#### **Security Requirements**

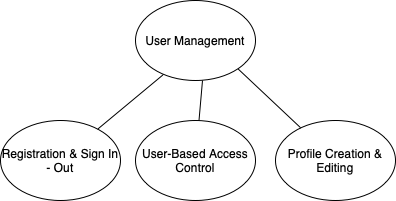
* **Data Encryption**: Data should be encrypted during transit and at rest.
  + This will be done by HTTPS, and SSL certificate
* **Authorization**: Role-based access control for different types of users.
* **Secure APIs**: All APIs should be secure and only accessible through authorized methods.

By addressing these requirements, the project will provide an efficient and secure experience for both athletes and companies.

### ***2.4 Subsystem Architecture Design***

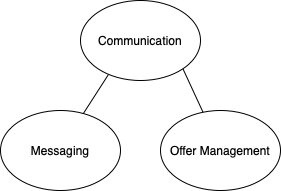
#### **a. User Management Subsystem:**

* **Description**: Handles user registration, authentication, profile management, and Users (Admin, Company, Athlete).
* **Components**:
* Registration & Sign In/Out.
* User-Based Access Control.
* Profile Creation & Editing.



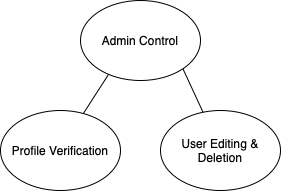
#### **b. Communication Subsystem:**

* **Description**: Allows in-app messaging between companies and athletes, including offer negotiations.
* **Components**:
* Messaging System.
* Offer Management (Create, Accept, Decline, Counter).



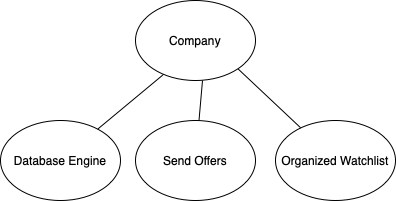
#### **c. Admin Control Subsystem:**

* **Description**: Provides control for administrators to oversee, validate, and intervene when necessary.
* **Components**:
* Profile Verification.
* User Editing & Deletion.



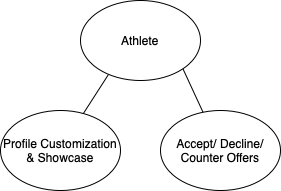
#### **d. Company Subsystem:**

* **Description**: For companies to find, save, and interact with potential athlete profiles.
* **Components**:
* Filtered Athlete Pool.
* Send Offers to Athletes.
* Organized Watchlist.



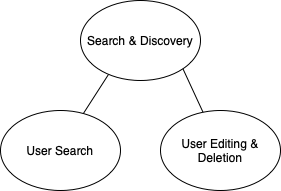
#### **e. Athlete Subsystem:**

* **Description**: Enables athletes to market themselves effectively and interact with companies.
* **Components**:
* Profile Customization & Showcase.
* Accept/Decline/Counter Offers.



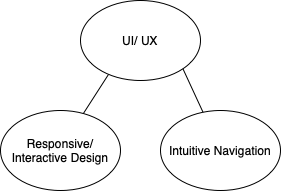
#### **f. Search & Discovery Subsystem:**

* **Description**: Provides a search capability for both companies and athletes.
* **Components**:
* User Search
* Advanced Filtering Options



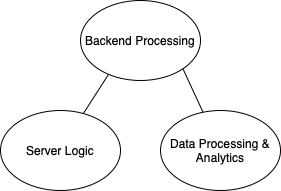
#### **g. UI/UX Subsystem:**

* **Description**: Focuses on delivering an optimal user experience across devices.
* **Components**:
* Responsive - Interactive Design.
* Intuitive Navigation.



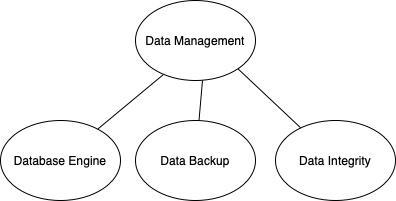
#### **h. Backend Processing Subsystem:**

* **Description**: Deals with the server-side logic, data processing, and other computations.
* **Components**:
* Server Logic (Python Flask).
* Data Processing & Analytics.



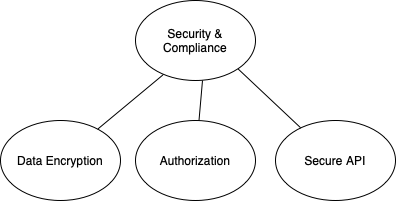
#### **i. Data Management Subsystem:**

* **Description**: Ensures secure, efficient, and reliable data storage and retrieval.
* **Components**:
* Database Engine (PostgreSQL).
* Data Backup & Recovery.
* Data Integrity.



#### **j. Security & Compliance Subsystem:**

* **Description**: Protects user data, ensures secure communication, and enforces platform integrity.
* **Components**:
* Data Encryption (HTTPS, SSL).
* Authorization & Role-based Permissions.
* Secure APIs.



### ***2.5 Subsystem – Functionality Diagrams***

#### **a. User Management Subsystem:**

**A diagram of a person's diagram

Description automatically generated**

#### **b. Communication Subsystem:**

**A diagram of a person's communication process

Description automatically generated**

#### **c. Admin Control Subsystem:**

**A diagram of a person

Description automatically generated**

#### **d. Company Subsystem:**

**A diagram of a company

Description automatically generated**

#### **e. Athlete Subsystem:**

**A diagram of a person with text

Description automatically generated**

**3. Project Specification**

* **Area of Focus**

The primary focus of the application is to create a web application that connects college athletes and sponsoring companies, in compliance with the newly updated NCAA regulations. Our target audience includes college athletes seeking sponsorships and companies interested in sponsoring athletes to promote their products or service.

* **Libraries, Frameworks and Development Environment**

The web application will primarily be developed using Python Flask for the backend, HTML, CSS, and JavaScript for the frontend, and PostgreSQL for the relational database. The framework will include React to enhance the frontend.

* **Platform**

The primary platform will be web-based for desktop browsers.

* **Genre**

The application SponsorIN falls under the Business Networking and Sports Management genre.

* **Responsibilities**

1. Brandon Tiong: Mainly responsible for backend and database.
2. Emanuel Macias: Mainly responsible for frontend.
3. Ege Keser: Mainly responsible for documentation and wherever help is needed, frontend, backend, database.

* **Source Code**

Source code for the SponsorIN project can be accessible through the link below:

<https://github.com/twosize/SponsorIN>

* **Team Members**

1. Brandon Tiong: <https://github.com/twosize>
2. Emanuel Macias: <https://github.com/DarkNeo777>
3. Ege Keser: <https://github.com/egekeser>