



**VIT<sup>®</sup>**  
**CHENNAI**  
Vellore Institute of Technology  
(Deemed to be University under section 3 of UGC Act, 1956)

**SOFTWARE ENGINEERING**  
**DA -1**  
**COLLAGE SELECTION APP**

**NAME :- ANKIT**

**22BPS1023**

## 1. Introduction

### 1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to detail the requirements for the "**College Selector**" mobile application. This application will assist high school students in finding colleges that match their academic profile, interests, and financial situation.

### 1.2 Document Conventions

This document follows the IEEE 830-1998 standard for an SRS. The following conventions are used throughout this document:

- **Bold text** is used for section titles and key terms.
- *Italicized text* is used for emphasis.
- Monospace font is used for code snippets or system commands.
- **[Requirement ID]** is used to uniquely identify each requirement.

### 1.3 Intended Audience and Reading Suggestions

This document is intended for the following stakeholders:

- **Project Managers:** To understand the project scope, timeline, and resource allocation.
- **Developers:** To understand the technical requirements and architecture of the application.
- **Quality Assurance (QA) Testers:** To create test plans and validate that the final product meets the specified requirements.
- **Clients/Stakeholders:** To review and approve the project requirements.

Readers are advised to read the "Project Scope" section first to get a high-level overview before diving into the detailed functional and non-functional requirements.

### 1.4 Project Scope

The initial release of the "College Selector" app will focus on the following core functionalities:

- User registration and profile creation.
- A comprehensive college search engine with multiple filtering options.
- The ability to view detailed college profiles.
- A side-by-side college comparison tool.
- A feature to save and manage a list of favorite colleges.

The following features are considered out of scope for the initial release and are planned for future versions:

- Integration with college application portals.
- Scholarship and financial aid search functionality.
- A social networking feature for connecting with other students.

## 1.5 References

- College Board (2024). *College Data and Statistics*.
- U.S. Department of Education (2024). *National Center for Education Statistics (NCES) Database*.
- IEEE (1998). *IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998)*.

## 2. Overall Description

### 2.1 Product Perspective

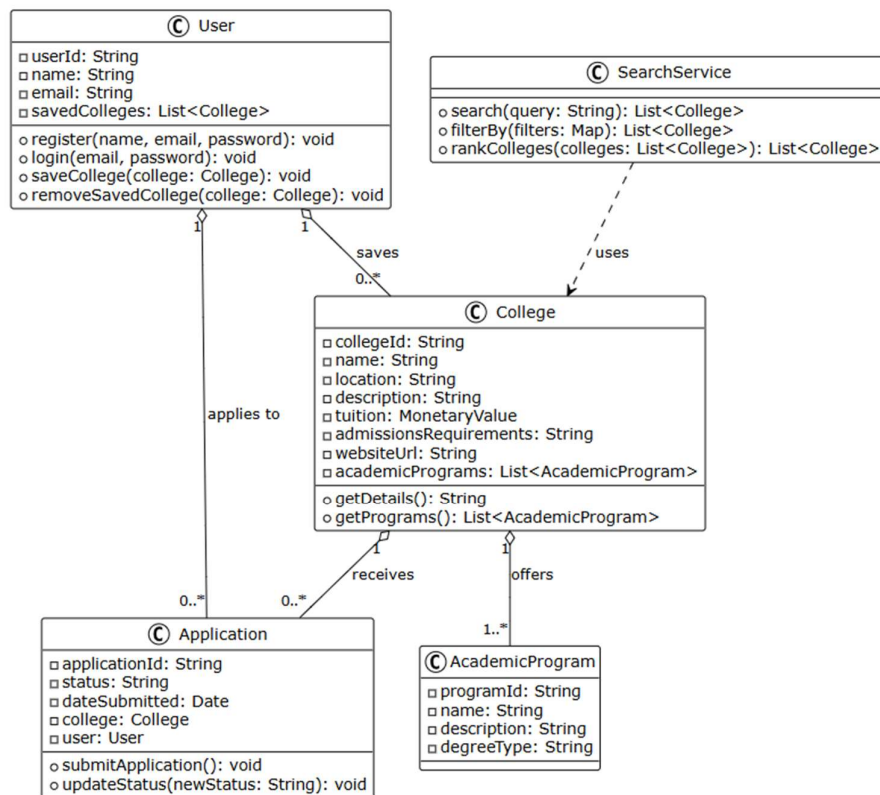
The **College Selector** app is a standalone mobile application, which means it is not a component of a larger system. It will be developed for both iOS and Android platforms and will not require a companion website for its core functionality. The app will interact with an external, cloud-based database to retrieve and update college information, but its primary function is to serve as a user-facing tool on a mobile device.

### 2.2 Product Features

The "College Selector" app will offer the following key features to its users:

- **User Profile Creation:** Allows students to input their academic data (e.g., GPA, SAT/ACT scores) and personal preferences (e.g., major, location).
- **Advanced College Search:** Provides a robust search engine with multiple filters such as tuition range, acceptance rate, student body size, and available majors.
- **Detailed College Profiles:** Displays comprehensive information for each institution, including admission requirements, student-to-faculty ratio, campus life, and contact information.
- **College Comparison Tool:** Enables users to select and compare up to three colleges side-by-side on a single screen.
- **Favorites List:** Allows users to save colleges to a personalized list for easy access and tracking.
- **User Notifications:** The app will send push notifications for reminders and updates (e.g., when a new college is added to the database that matches a user's criteria).

## College Selection App - Class Diagram



### 2.3 User Classes and Characteristics

The application is designed to cater to the following primary user classes:

- **High School Students (Primary Users):**
  - **Characteristics:** Aged 16-18, tech-savvy, looking for an intuitive tool to simplify their college search.
  - **Needs:** Easy-to-use search functionality, clear and concise college information, and a way to organize their options.
- **Parents (Secondary Users):**
  - **Characteristics:** May have less technical proficiency than students, are often concerned with financial aspects and safety.
  - **Needs:** Access to transparent tuition and financial aid data, and the ability to compare multiple institutions.
- **School Counselors (Tertiary Users):**
  - **Characteristics:** Experienced in the college application process, may use the app as a supplementary tool for student advising.
  - **Needs:** Quick access to reliable college data and a way to view student's saved lists to provide guidance.

## 2.4 Operating Environment

The app is designed to run on:

- **Mobile Platforms:** iOS 14.0 and higher, and Android 10.0 and higher.
- **Hardware:** Modern smartphones and tablets with a minimum of 4GB RAM and a multi-core processor.
- **Network:** An active internet connection (Wi-Fi or cellular data) is required for full functionality, particularly for accessing the college database and external APIs.
- **External Systems:** The app will interface with third-party APIs for mapping services and data from educational databases.

## 2.5 Design and Implementation Constraints

- **Platform Compatibility:** The app must be developed using a cross-platform framework (e.g., Flutter or React Native) to ensure consistency and reduce development time and cost.
- **Database Management:** The backend database must be scalable to handle a large and growing number of college data entries and user profiles.
- **Data Source:** The college data must be sourced from a reputable and up-to-date provider (e.g., NCES or a licensed API).

## 2.6 Assumptions and Dependencies

- **Assumption:** The project assumes that high-quality, up-to-date college data will be available from the chosen data provider.
- **Dependency:** The successful deployment of the app is dependent on the availability and reliability of the third-party APIs used for maps and college data.
- **Assumption:** The target users (students) have access to a modern smartphone with an internet connection.

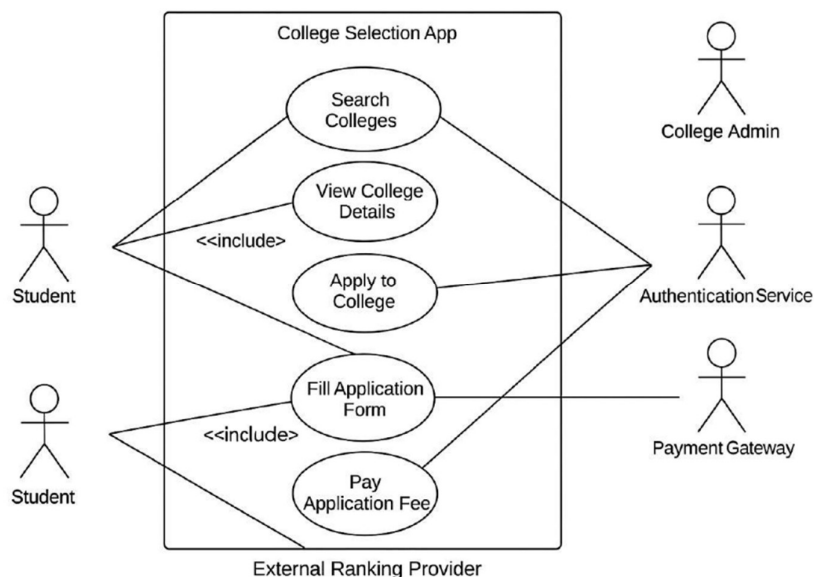
## 3. System Features

### 3.1 Functional Requirements

The following functional requirements describe the specific behaviors and functions that the College Selector app must perform. Each requirement is testable and directly related to a user's task.

- **User Registration and Login:** The system shall allow new users to register an account using a valid email address and password. It shall also allow existing users to log in with their credentials.
- **Profile Management:** The system shall enable users to create and edit their academic profile, which includes fields for GPA, standardized test scores (SAT/ACT), intended major, and a list of extracurricular activities.
- **Advanced Search with Filters:** The system shall provide a search interface that allows users to filter the college database based on specific criteria. The filters must include:

- **Location:** State, region, or city.
- **Tuition and Fees:** A user-defined price range (e.g., \$0 - \$10,000, \$10,001 - \$20,000).
- **Acceptance Rate:** A selectable range (e.g., highly selective, moderately selective, open admission).
- **Student Body Size:** Small (<2,000), medium (2,000-15,000), or large (>15,000).
- **Majors:** A list of available fields of study.
- **College Profile Display:** When a user selects a college from the search results, the system shall display a detailed profile page. This page must include, at a minimum:
  - General information (location, size, public/private status).
  - Admission requirements (GPA, test score ranges).
  - Tuition and financial aid details.
  - Available majors and academic programs.
  - Campus life and student demographics.
- **College Comparison:** The system shall allow users to select up to three colleges and view their key metrics side-by-side on a single screen for comparison. The comparison view must include tuition, acceptance rate, and average SAT/ACT scores.
- **Save to Favorites:** The system shall allow a logged-in user to save a college to a personal "Favorites" list for later review.
- **Push Notifications:** The system shall be able to send push notifications to users for updates on saved colleges or to remind them of upcoming application deadlines.



## 4. External Interface Requirements

### 4.1 User Interfaces

The user interface (UI) for the College Selector app will be designed for a seamless and intuitive mobile experience.

- **Layout and Design:** The UI will be clean and minimalist, using a modern color palette and typography. It will be responsive to different screen sizes and orientations (portrait mode is the priority).
- **Navigation:** The app will use a bottom navigation bar for core features (e.g., Home, Search, Favorites, Profile) to ensure easy access.
- **Screen Flow:** The flow from search to college profile and comparison will be linear and logical, minimizing the number of taps required to complete a task.
- **Feedback:** The system will provide clear visual and haptic feedback for user actions, such as button presses and data loading.

4.2 Hardware Interfaces

The app will need to interface with the standard hardware components of a modern smartphone to deliver its full functionality.

- **Display:** The app will be optimized for high-resolution touchscreens.
- **Processor and Memory:** It will be developed to run efficiently on standard mobile processors with a minimum of 4GB of RAM.
- **Push Notification Capability:** The app will require access to the device's native push notification system to send reminders and alerts.
- **Geolocation:** The app may require access to the device's GPS to provide location-based search results or map services for college locations.

4.3 Software Interfaces

Sure, here's the "Software Interfaces" section for the College Selection App SRS document converted into a table format for clarity and quick reference.

---

4.3 Software Interfaces		
Interface	Description	Purpose
Operating System APIs	iOS (Swift, SwiftUI) & Android (Kotlin, Jetpack Compose) SDKs.	To ensure the app integrates seamlessly with native mobile device functionalities and provides an optimized user experience on both platforms.
Database API	RESTful API communicating with a	To fetch, store, and update all college data, user profiles, and "Favorites" lists. All communication will be encrypted.

Interface	Description	Purpose
	cloud-based backend database.	
Mapping API	Third-party mapping service (e.g., Google Maps API).	To display college locations, provide directions, and enhance the location-based search feature.
Authentication Service	A third-party service (e.g., Firebase Authentication).	To handle secure user registration, login, and password management, offloading this critical security function from the app's core development.
Notification Service	Native push notification APIs for iOS and Android.	To send push notifications for reminders, updates, and alerts to the user's device.

#### 4.4 Communications Interfaces

The app's primary communication interface will be its connection to the internet to access and transfer data.

- **Network Protocol:** All communication between the app and the backend server will be encrypted using **HTTPS** to ensure data security.
- **Data Format:** Data will be transmitted in **JSON** format, which is a lightweight and widely used data interchange format.
- **Data Transfer:** The app will use a low-bandwidth data transfer model to ensure a smooth user experience even on slower cellular networks. Data will be fetched on demand rather than all at once.

### 5. Nonfunctional Requirements

Non-functional requirements specify criteria that can be used to judge the operation of a system rather than its specific behaviors. These are crucial for the user experience and overall success of the app.

#### 5.1 Performance Requirements

- **Response Time:** The app shall display search results and load college profile pages within **3 seconds** of a user request under normal network conditions.
- **Scalability:** The backend system shall be able to handle a minimum of **1,000 concurrent users** without a noticeable degradation in performance.
- **Launch Time:** The app shall launch and become fully interactive within **4 seconds** on a clean installation on supported devices.



## 5.2 Safety Requirements

The app does not have direct control over physical systems, so safety requirements are primarily related to data integrity and user well-being.

- **Data Integrity:** The system shall prevent data corruption and ensure the consistency of college information displayed to all users. All data updates from the source must be verified for accuracy before being published.
- **User Well-being:** The app shall not contain any content that could be harmful to minors, such as inappropriate advertisements or links to unsafe external sites.

## 5.3 Security Requirements

- **User Authentication:** All user passwords shall be stored using a strong, one-way cryptographic hash function (e.g., bcrypt) with a salt. Passwords will never be stored in plain text.
- **Data Encryption:** All data transmitted between the mobile app and the backend servers shall be encrypted using **HTTPS/TLS** protocols.
- **Access Control:** The system shall enforce role-based access control to prevent unauthorized access to user data. Only authenticated users can view their own saved preferences and profile information.
- **Vulnerability Management:** The software will be regularly scanned for security vulnerabilities, and any critical vulnerabilities will be patched within 48 hours of discovery.

## 5.4 Software Quality Attributes

- **Usability:** The app's user interface shall be intuitive enough for a new user to successfully perform their first college search within **one minute** of launching the app.
- **Reliability:** The app shall have a minimum uptime of **99.9%** for its core functionalities (search, profile view, and saving).
- **Maintainability:** The codebase will be well-documented and follow established coding standards to allow new developers to easily understand and modify the code.
- **Portability:** The app will be developed using a cross-platform framework (e.g., Flutter or React Native) to ensure it can be easily deployed to both iOS and Android platforms with minimal changes.