**Section 1:  
1.1 Define Student Experience Objectives**

Establish measurable short-term objectives such as improving grades, managing time effectively, or increasing involvement in extracurricular activities.

Set long-term goals, such as earning a degree, obtaining internships, or preparing for a job.

***Frontend:*** Create a user-friendly interface using modern frameworks such as React or Vue.js to provide a smooth user experience.

***Backend:*** Use scalable and robust backend technologies like Node.js or Django to manage the app's logic and data.

**1.2 App Features and Learning Algorithms**

Use machine learning techniques to adapt to the user's actions and preferences.

Monitor and analyze the outcomes of the set objectives.

Provide decision-making assistance based on the user's schedule, academic deadlines, and personal goals.

* **Machine Learning Algorithms**

Implement machine learning models with frameworks like TensorFlow or PyTorch to study user behavior and personalize the app's assistance.

* **Data Tracking and Measurement**

Use a robust data monitoring system, potentially employing cloud services like AWS or Google Cloud, to measure outcomes relevant to the set student objectives.

**1.3 Requirements**

* **Usability and Utility**
* Design the software with a user-friendly and simple layout, considering the needs of university students.
* The app should be helpful in providing relevant academic and extracurricular information.
* **Robustness and Portability**
  + Design the software to be compatible with numerous devices and operating systems.
  + Ensure scalability to support an increasing number of users and evolvability for future feature additions.
  + Use an effective data monitoring system, potentially utilizing cloud services like AWS or Google Cloud, to measure outcomes relevant to the set student objectives.
* **Security and Privacy**
  + Implement robust security measures to protect user data and maintain privacy.
  + Handle sensitive academic information with the utmost integrity.
  + Use industry-standard security measures, including end-to-end encryption and secure authentication procedures, to safeguard user data and privacy.
* **Self-Evaluation Mechanism**
  + Include a *feedback loop* where the app evaluates its decision-making effectiveness.
  + Adjust algorithms based on user feedback and success in meeting objectives.
  + Develop algorithms that provide decision-making guidance based on the user's schedule, academic deadlines, and personal goals, possibly using decision trees or reinforcement learning techniques.
* **Continuous Improvement**
* Regularly collect feedback from users and statistics on usage to refine and improve the app's features and algorithms.
* Keep up with the latest developments in education technology and student demands to ensure that the app remains fresh.