# 3. Specific Requirements

## 3.1 Use Case Reports

No matter what specific use case, our product contains two essential requirements.

*The database:*

To store and look up study sessions, there must be a database that contains all of the information about a study session. This should be implemented as a SQL database with attributes for each session such as time, class, coordinator, and any additional notes about a session.

*The web interface:*

Our database must provide some interface for end users to look at study sessions. We have decided to implement this as a web application, so there must be a reasonably intuitive website interface that implements all of our desired features.

*User Profile:*

For all users, we require a user profile. This can hold information such as major, year, email, and a messaging service. As with many of our data, this can be implemented in the form of a database of users. The web interface can then provide a way to view user details.

Additionally, there are use case-specific requirements. Going by the three-user model we described in section 2, we have therefore three sets of requirements:  
  
**The Student:**

As described above, the student uses our product to find sessions of study groups. This means that the following things need to be available to student using our product:

*Search:*

There must be an intuitive interface that allows a student to search for study sessions. This must have options to be fine-grained; students must be able to search for sessions for specific classes at specific times, but must also be able to be wide (e.g., a request for all Computer Science study sessions).

*Detailed Information:*

Once a session is selected, a user must be able to see all of the attributes of that session.

*RSVP:*

There must be some system implemented whereby a student can commit to attending a study session. This must be reflected in the study session (which says how many people and who is going), as well as be available to the student in the form of what sessions he has committed to.

**The Coordinator:**

Similarly to the user, a coordinator requires an ability to RSVP to sessions as well as view sessions he has committed to. A coordinator’s user profile should also include subjects he is proficient in, and can therefore coordinate.

**The System Admin:**

*Database Interface:*For admins, we must provide an interface to create sessions, and manage all of the attributes of that session that we have mentioned; (time, location, date, subject, etc). Additional options such as a private session that requires admin approval may be toggled.

There must also be an interface to cancel or delete sessions that an admin has started.

## 3.2 Non-Functional Requirements

### **3.2.1 Performance**

As with any web applications, users expect our product to be responsive in a short amount of time. The platform must be optimized for quick responses, and must also not take a prohibitive amount of RAM/CPU time on the client computer.

### **3.2.2 Reliability**

Our product must be reliable; users must be sure that their RSVPS, searches are working properly. If there are bugs or unexpected issues, clear, informative error messages are expected. Our product must be able to scale to a large amount of users if need be, to accommodate things like exams or papers when usage may spike sharply.

### **3.2.3 Availability**

Our product must adhere to the standard of availability people are used to. Uptime should be 99%, with any availability issues communicated clearly to the user and fixed in a timely fashion. Especial care should be taken to ensure availability during times when students need to find study sessions; midterms or finals, for example.