# Proposed Software Architecture

[INTRODUCTION GOES HERE]

## Overview

The architectures that we chose to implement for the SOS system are the three-tier and repository architectures. One reason we chose three-tier architecture was because we are planning to deploy our different subsystems on different nodes rather than on a single node. In addition, we chose three-tier architecture because it has superior performance for medium to high volume environments. This is important because our target audiences are the university community which is always changing and growing; therefore, we need a site that could take a large number of requests simultaneously. In addition, we also chose the repository architecture because each of the subsystems that we have identified would depend on the central data structure, the data storage, which acts as a repository. The team chose this architecture because it serves as an efficient way to store a large amount of data and retrieve data from the single source. In addition, repository architecture also allows for the reduction of overhead of transient data between software components. This quality of repository architecture allows us to efficiently transfer/store data to and from the application layer of the system and the storage layer of the system. Ideally this meets the standards of the nonfunctional requirement of both performance and reliability for most of our use cases since both architectures ensure that the system will be responsive and quick to handle requests.

## Subsystem Decomposition

The subsystems that our team identified are: Data Store, SOS Website, SOS Controller, User Management, Event Management, Organization Management, Security Management and Google Maps GPS API. The Data Store subsystem will act as the central node in the repository architecture where data is both stored and retrieved. In addition, the Data Store will be part of the storage layer in the thee-tier architecture and will be responsible for keeping track of the persistent data of the SOS application. The SOS Website subsystem is the presentation layer of the three-tier architecture. The functionality that this SOS Website has is that it acts as the user interface to display the site to the member on any machine which they access it from. The SOS Controller will act as the server of the application, which will process requests from the SOS Website and create objects of other subsystems that will interact with the Data Store. The SOS Website is in charge of receiving and dispatching requests from the User Website. The User Management subsystem is a subsystem that solely executes user related functionality such as SOS22, SOS10, SOS07 and SOS02. The Event Management subsystem is a subsystem that solely executes the event related functionality such as SOS04, SOS01, SOS10 and SOS17. The Organization Management subsystem is a subsystem that solely executes the event related functionality such as SOS02 and SOS16. The Security Management subsystem is a subsystem that authenticates the actions that a particular user is attempting to perform. The use cases that are related to the Security Management subsystems are SOS02, SOS07, SOS22, SOS31 and SOS32. The final subsystem is the Google Maps GPS API and it is responsible for retrieving location coordinates for events and users and the use cases that correspond to this subsystem is SOS10 and SOS01.

## Hardware and Software Mapping

## Persistent Data Management

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Persistent Data Name: | Field Name | Field Size | Data Type | Data Format | Example | Description |
| User | date\_of\_birth | 8 chars | string | MM/DD/YY | 05/14/97 | Date of birth of a user. |
| first\_name | 20 chars | string | ----------------- | Rick | First name of the user. |
| last\_name | 20 chars | string | ----------------- | Sanchez | Last name of the user. |
| user\_name | 20 chars | string | ----------------- | TiredgeSnius68 | The username chosen by each user. |
| email | 20 chars | string | username@domainname | Szechuan808@hotmail.com | The email address the user registers with. |
| password | 20 chars | string | ----------------------------- | 3o9t23bf4180rf87b2387 | The encryption of the password chosen by the user. |
| user\_id | ------ | Int | ----------------------------- | 1 | The unique ID given to the user when they create an account. |
| privacy | 20 chars | string | PRIVATE  PUBLIC | PRIVATE | The privacy that the user has their account set to. |
| Event |  |  |  |  |  |  |
|  |  |  |  |  |  |

## Security Management