

**Student Organization Software
Software Requirements Specifications**

Version 1.0

Student Organization Software	Version: 1.0
Software Requirements Specifications	Date: 9/28/2022

Revision History

Date	Version	Description	Author
09/29/2022	1.0	First version	JetLAG Team

Student Organization Software	Version: 1.0
Software Requirements Specifications	Date: 9/28/2022

Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms, and Abbreviations	4
1.4	References	4
1.5	Overview	4
2.	Overall Description	5
2.1	Product perspective	5
2.1.1	System Interfaces	5
2.1.2	User Interfaces	5
2.1.3	Software Interfaces	5
2.2	Product functions	5
2.3	User characteristics	6
2.4	Constraints	6
2.5	Assumptions and dependencies	6
2.6	Requirements subsets	6
3.	Specific Requirements	6
3.1	Functionality	6
3.1.1	Note Taking functionality	6
3.1.2	Time management	6
3.1.3	Front end navigation	6
3.1.4	Adding and removing classes	6
3.1.5	Cloud save functionality	6
3.1.6	Cloud sync functionality	6
3.1.7	Cross platform	6
3.1.8	Account Registration	6
3.1.9	User sign in and authentication	7
3.2	Use-Case Specifications	7
3.3	Supplementary Requirements	7
4.	Classification of Functional Requirements	7
5.	Appendixes	7

Student Organization Software	Version: 1.0
Software Requirements Specifications	Date: 9/28/2022

Software Requirements Specifications

1. Introduction

1.1 Purpose

The purpose of this document is to describe the requirement specifications for a piece of software that is built to help students take notes and manage time.

The intended audiences for this document are the team members on the project, and the faculty and staff of KU's software engineering course.

1.2 Scope

The software system to be produced is a Student Organization Software, which will be referred to as "SOS" throughout this document.

The Student Organization Software will allow students to record and track their productivity through a simple user interface. SOS will allow students to easily take notes and view calendar events all within the same application.

SOS could be used by any student who is interested in improving their productivity and grades. The objective of SOS is to improve student workflow and productivity.

1.3 Definitions, Acronyms, and Abbreviations

SOS – Student organization software

1.4 References

1.5 Overview

Section 2 Describes a general overview of the SOS, Section 3 Describes Specific requirement for the actual software to be developed.

Student Organization Software	Version: 1.0
Software Requirements Specifications	Date: 9/28/2022

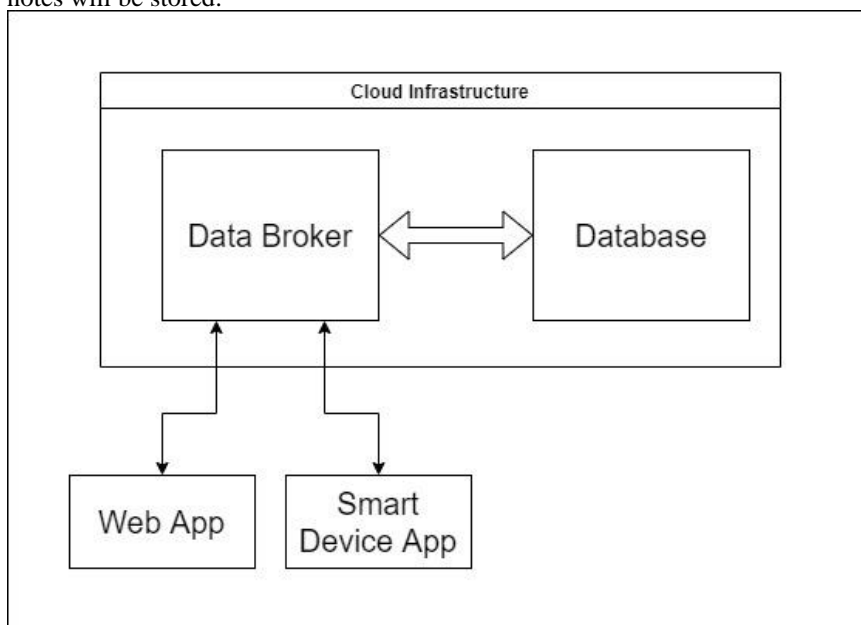
2. Overall Description

2.1 Product perspective

Students while in school find themselves having to manage information from and about classes across multiple mediums, and software products. This most commonly includes having to take notes about class contents, and manage important dates such as due dates or exam dates. Managing across multiple services can cause forgotten tasks, and increase in time spent on entering information. The Goal of SOS is to make all of this information managed from a singular app, that can be accessed from multiple devices for convenience to users.

2.1.1 System Interfaces

The SOS software is designed to work over the internet with various different front ends for user convenience during any given use, there will be 3 module active at once, a front end application, responsible for interfacing with the user, a data broker responsible for coordinating sync and save functionality, and a database where user dates and notes will be stored.



Backend Application: cloud hosted virtual machine

Front End Applications: mobile and browser based

Backend Database: cloud hosted Database

2.1.2 User Interfaces

The user interface can take a few different forms, it can either be a physical app or be accessed from a browser. In either case the functionality will be the same. Users to sign in are able to access class calendars and add and remove dates, create and remove classes, or add and delete notes. From the app the user can choose to save to the cloud or pull the last save from the cloud.

2.1.3 Software Interfaces

This software will take advantage of a few different connection types. Between the front end applications and the data broker, the UI will connect to a rest endpoints to retrieve data for the given authenticated user. the data broker will also have an open TLS connection to the database to store user information.

2.2 Product functions

This product will function as a centralized note-taking and time management app that will have the following features

Student Organization Software	Version: 1.0
Software Requirements Specifications	Date: 9/28/2022

- Cross platform saving
 - Allows for users to view and add important dates and take notes on a per class division.
 - Able to access your notes in either a browser or on a mobile device.
 - Allow users to classify note files to the Assignments they pertain to.
- The main functions of Student Organization Software are to allow students to take notes, create and view calendar events, and manage their courses.

2.3 User characteristics

The users of SOS are students who have classes in which they want to manage. Users are knowledgeable on how to navigate a simple user interface which will lead them to the functionality they aim to use.

2.4 Constraints

In this project, solutions will have to be of no cost to run, therefore things such as cloud apps must fit within the free-tier services for students on platforms like azure.

2.5 Assumptions and dependencies

Users will have an internet connection whenever they plan to use the software.

2.6 Requirements subsets

3. Specific Requirements

3.1 Functionality

3.1.1 Note Taking functionality

A user will be able to take notes in the form of .txt documents for each given class.

3.1.1.1 Note Sharing

A user will be able to share their notes with other users by inputting a user id to share with.

3.1.1.2 Note Editing

A user will be able to edit previous notes if wanted.

3.1.2 Time management

A user will be able to add events to a calendar for each given class.

3.1.3 Front end navigation

A user upon first opening the app will be able to see all classes they have added, and select one to see its notes and calendar.

3.1.4 Adding and removing classes

A user will be able to add or remove classes, when a class is removed, all of its notes will also be removed.

3.1.5 Cloud save functionality

When a user selects the save button in the user interface, it will save that current version to the cloud.

3.1.6 Cloud sync functionality

Upon signing in or when a user selects sync in the user interface, the last save will be pulled from the cloud and displace in that UI

3.1.7 Cross platform

A user will be able to view the application from any device ranging from phone, tablet, or computer. The user interface should look the same on all platforms.

3.1.8 Account Registration

A user will be able to create their own personal account in which all data from that user will be stored under. This

Student Organization Software	Version: 1.0
Software Requirements Specifications	Date: 9/28/2022

functionality includes the ability to login anywhere from the same user id as well as being able to logout when wanted.

3.1.9 User sign in and authentication

User upon accessing the app will have to enter in a username and password to access their unique notes. This will then be validated by the data broker before further access is allowed.

3.2 Use-Case Specifications

Refer to Use case specifications document.

3.3 Supplementary Requirements

Refer to symmetry requirements document.

4. Classification of Functional Requirements

Functionality	Type
3.1.1 Note Taking functionality	Essential
3.1.2 Time management	Essential
3.1.3 Front end navigation	Essential
3.1.4 Adding and removing classes	Essential
3.1.5 Cloud save functionality	Essential
3.1.6 Cloud sync functionality	Essential
3.1.7 Cross Platform	Desirable
3.1.8 Account Registration	Essential
3.1.1.1 Note Sharing	Desirable
3.1.1.2 Note Editing	Essential

5. Appendixes