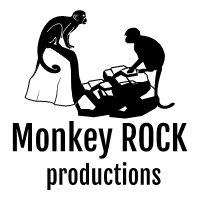
# 



Sunland Home School Co+op software Requirement Specification: Version 1.7

# **2. Table of Contents**

[2. Table of Contents 2](#_Toc6946004)

[3. Customer Statement of Requirements 2](#_Toc6946005)

[4. Glossary of Terms 3](#_Toc6946006)

[5. Functional Requirements Specification 4](#_Toc6946007)

[a. Stakeholders 4](#_Toc6946008)

[b. Actors and Goals 4](#_Toc6946009)

[c. Use Cases 5](#_Toc6946010)

[i. Use Case Diagrams 5](#_Toc6946011)

[ii. Casual Description 5](#_Toc6946012)

[iv. Activity Diagrams 7](#_Toc6946013)

[v. Use Case Descriptions 8](#_Toc6946014)

[d. Interaction Diagrams 14](#_Toc6946015)

[e. Overview Class Diagrams 15](#_Toc6946016)

[f. Potential Class Domains 15](#_Toc6946017)

[6. Nonfunctional Requirements 16](#_Toc6946018)

[7. References 16](#_Toc6946019)

[8. User Interface 17](#_Toc6946020)

[9. Effort Breakdown Table 28](#_Toc6946021)

[***Responsibility Matrix with point breakdown*** 28](#_Toc6946022)

[10. Appendix 29](#_Toc6946023)

# 3. Customer Statement of Requirements

The purpose of the Software Requirements Specification is to provide agreement on what the software product or the portion of the product to be implemented in the software is to do. Specifically, it provides a means of communication between Sunland Home school Co+op, Directors, IT Gurus, The board, system analysts, software design team, engineer, and software quality engineers.

This document will help the reader understand the many perspectives of each person involved, and will help limit confusion, and assist with asking the proper questions, or giving the proper answers as necessary. It will limit the need for direct person to person contact by acting as a reference source for the systems information.

This document will support design by acting as a go too for the software design time. It is equipped with effective diagrams to help assist with class, and function creation. Improving efficiency in this area will also assist quality control with their activities to ensure the best possible version are being released at proper milestones.

The document will enhance support and testing should issues ever occur with a marketed version of the system. If an error occurs during system operation the IT Gurus can use this as a reference for trouble shooting. Should they not be able to succeed in repairing the error they can solve the issue by passing it off to monkey ROCK productions support team, and they may use this document to assist with further support, including updating the document.

And important step in system design is a verification process, this document will assist with that by acting as, or providing a benchmark checklist for goal and timeline completion. This is best thought of as a system roadmap from beginning to project closure. Finally, this document will support system version updates and will be actively updated as the system is modified, again this will occur all the way until system discontinuation.

All of the aforementioned topic will ensure that Sunland Homeschool Co+Op not only hits their goals and targets, but has an easier time hitting futured declared targets. This document will help the system scale to Sunland’s needs as they expand into new states and even region. As challenges arise during this document can be employed for useful navigation.

# 4. Glossary of Terms

**Use case:** An action that can be implemented by users. The use case diagram is a mapping of who can perform which actions.

**Database:** The storage of information.

**User Interface (UI):** How the user interacts with the system to preform the various actions.

**Class:** An object within the system, built from various pieces of information.

**Inheritance:** The parent class (root) gives all of its attributes and functionality to the child class.

**Multiplicity:** The correlation between two objects shoeing how many each can have, the symbol \* signifies as many/infinite.

**Class Domain**: a main piece of a larger part in the class overview. The class domains are broken down and typically detailed for the programmers to easily implement the planned upon system.

# 5. Functional Requirements Specification

## Stakeholders

The stakeholders (the people -) within this system are the IT gurus, the board of directors, the director, homeschooling families/conference attendees (In this document, they are referred as members), guest speakers/vendors.

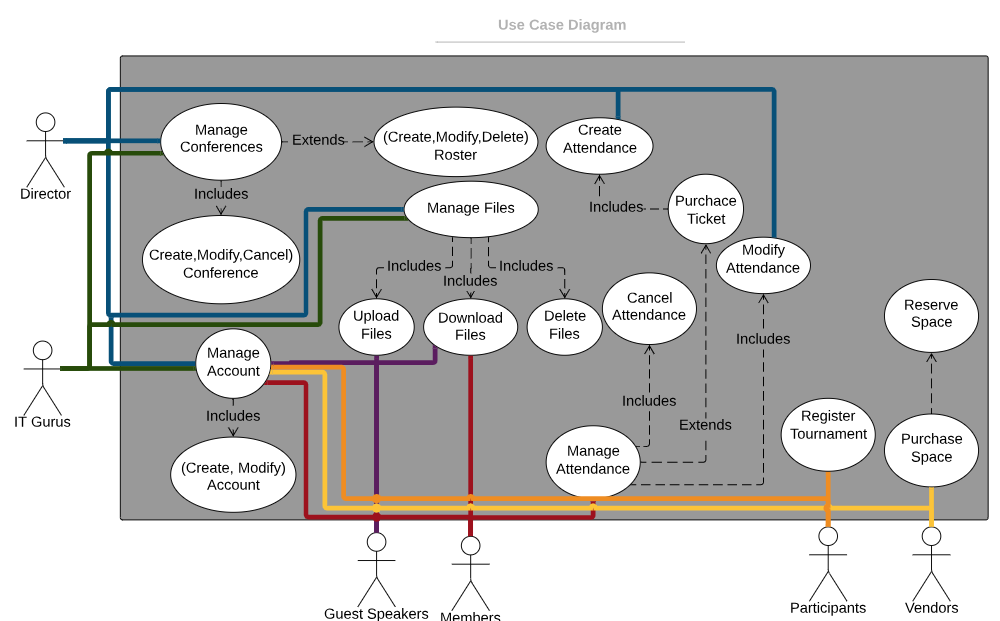
## Actors and Goals

The actors(users) the will interact with this system are as follows:

* **SYSTEM DIRECTOR**: the goal here is to all the director to easily manage conference, manage attendance schedules, and manage system files.
* **IT GURUS:** The gurus will be able to modify files, conferences, and general system maintenance or repair
* **VENDORS:** vendors will be able to reserve a space in the conference. They can also upload menus, and other pertinent data with ease.
* **GUEST SPEAKER:** speakers will be able to register for the conference and adjust their attendance with ease prior to the conference.
* **MEMBERS:** The members will be able to create an account, and once logged in they will be able to do things such as; buy tickets for a conference, modify attendance date as long as it is within the proper times. They will be able to interact with the forum and contact us pages of the system. They will be able to see and download files such as conference maps, vendor menus, and other planning tools within the system.
* **PARTICIPANTS:** The tournament participants will have all of the functionality of the members, but they will also be able to manage their tournament registry through the system.

## Use Cases

### Use Case Diagrams

Figure 1:

### Casual Description

The image above (figure 1) shows the connections on how each actor (user) can interact with the system. The diagram separates the users so that not every user interacts with the system the same way (ie. A member can’t manage conferences). The use case diagram is color coded to show the correlations and avoid confusion with crossing lines.

1. **CASUAL USE CASE DESCRIPTIONS.**

**MANAGE CONFERENCE:**

This use case will manage conferences and rosters. The director or the IT guru will be able to add, modify, or cancel conferences including the rosters of the conference. Adding a new conference will create and post a conference with the included date, times, vendors, guest speakers, and tournament participants. Once the conference is created by the director it can be modified by an IT guru if the system is experiencing trouble and it needs to be resolved. Modifying a conference can be done once it is created. Changes can be made to the date, time ranges, vendors, and rosters and necessary.

**MANAGE FILES:**

This use case will manage files for the conferences. The director or the IT Guru will be able to upload, download, and delete files from an existing conference as necessary. Files will include pamphlets, brochures, schedules, rosters that include guest speakers, tournament times and attendees, and vendors. Once the conference has been completed the Director can delete files as necessary.

\* The IT Guru should only be in contact with files if there is an error with the system and he is resolving the issue.

**MANAGE ACCOUNT:**

This use case will manage actor accounts. A user will visit the system landing page for the first time and click the create account button. They will then proceed to the essential information for necessary for account creation. Once the use has completed the form and confirmed account creation, they will then proceed to the user home screen page. From here the user will be able to interact with the system at varying levels of clearance depending on what actor the user is classified as. Interacting with the system will include the ability to modify their existing account information.

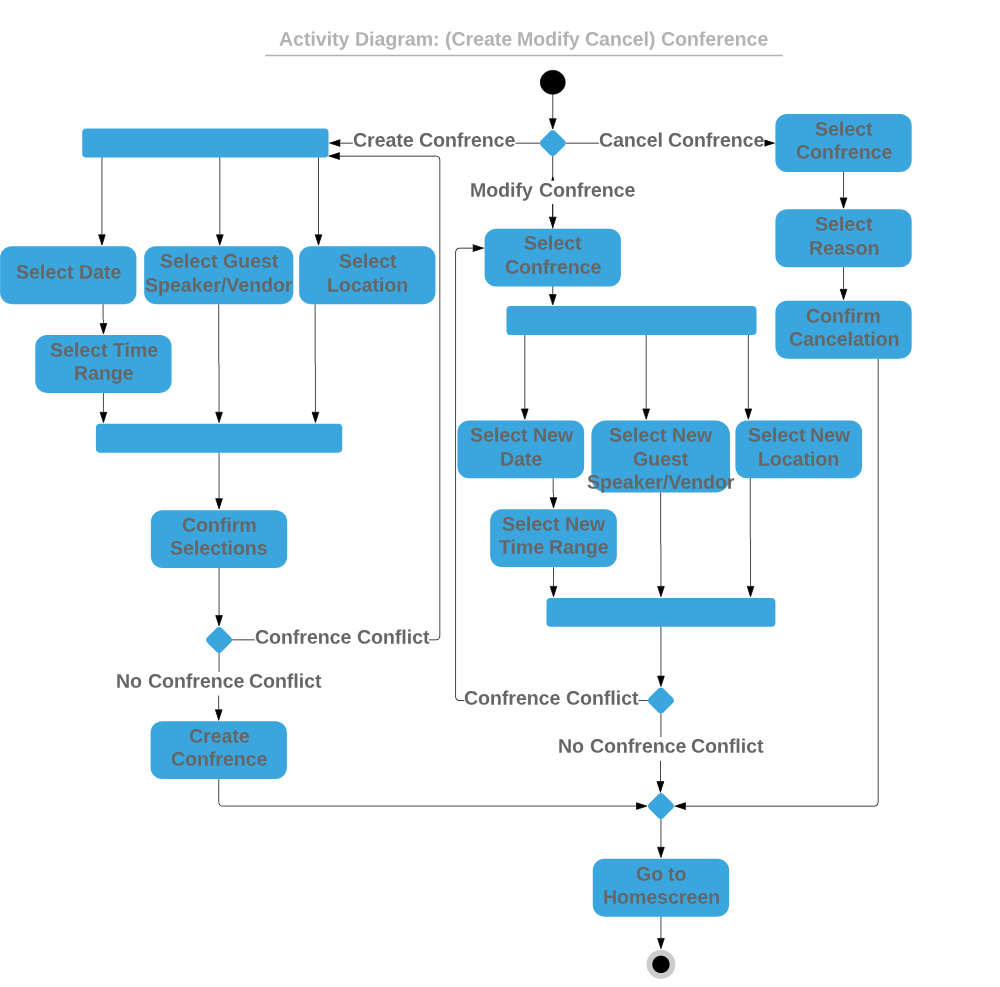
**MANAGE ATTENDANCE:**

This use case will manage actor attendance. A user will select an existing conference and click purchase ticket for desired date. They will enter payment information. The user will be allowed to cancel or transfer ticket if within the allotted time range.

**REGISTRY:**

This use case will manage registry items for participants into tournaments, and will also function similarly for vendor registration. A participant will be able to login to their account and select an existing tournament date. They can then fill out the registration form, and pay the fee. The vendor will login to their account, select vendor space registration, fill out the form, and pay the fee for an existing conference they wish to attend/serve at.

### Activity Diagrams

Figure 2:

The image above (figure 2) shows the process of creating, modifying, and canceling conferences. You may have noticed, creating and updating conferences follow similar processes, but the difference is that to update one, the user must select the specific conference on the calendar to change. The order in which the user selects the date, time, guest speaker, and location don’t particularly matter. With any changes or addition to conferences, hitting “confirm” is required.

### Use Case Descriptions

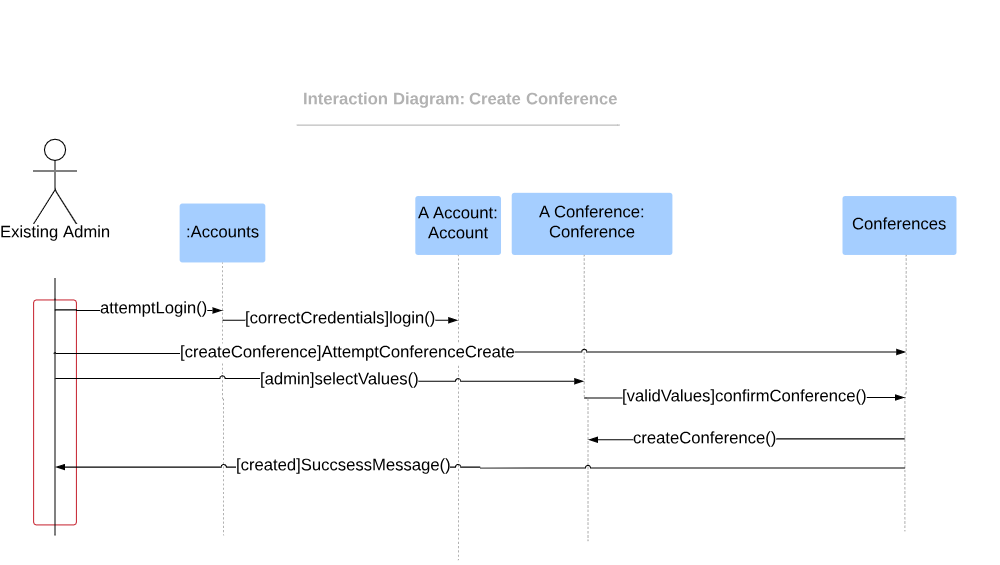
|  |  |  |
| --- | --- | --- |
| Use case name: Manage account | ID: 003 | Importance Level: 5 |
| Primary actor: Director, IT guru, Guest speaker, vendors, members, participants. | Use case type: Essential | |
| Stakeholders and interests:  Director, IT guru, Guest speaker, vendors, members, participants | | |
| Brief description:  This is a use case description for the activities that manage user accounts. | | |
| Trigger: A system actor will visit the system and create, or modify their account | | |
| Relationships:  Association:<related> Director, IT guru, Guest speaker, vendors, members, participants  Include: create, modify account  Extend:  Generalization: | | |
| Normal flow of events:   1. Visit system landing page 2. Click the create account button 3. Fill out essential information form 4. Confirm/save form 5. If an error occurs return to step 3 and correction error. 6. When complete proceed to login screen 7. End. | | |
| Sub flows:   1. Fill in username and password and click login on the system landing page 2. From user account home page click settings. 3. Edit essential information form 4. Confirm/save changes. 5. Return to user account home page 6. end | | |
| Alternate/exceptional flows:   1. fill out username and password click login 2. proceed to invalid credentials page 3. correct credentials provided or proceed with forgot username and password feature 4. if invalid 3 times lock out account 5. proceed to user account home page 6. end | | |

|  |  |  |
| --- | --- | --- |
| Use case name: Manage attendance | ID: 004 | Importance Level: 5 |
| Primary actor: members | Use case type: Essential | |
| Stakeholders and interests:  Guest speaker, vendors, members, participants | | |
| Brief description:  This use case description will cover the activities of managing user attendance. | | |
| Trigger: A member will log in and purchase a ticket for a scheduled conference, or modify attendance by transferring to a conference at a different date as long as it happens within the allotted time. The user may also cancel attendance and be refunded as long as it is within the allotted time. | | |
| Relationships:  Association:<related> Guest speaker, vendors, members, participants  Include: purchase ticket, cancel attendance, modify attendance.  Extend:  Generalization:<inheritance> create attendance. | | |
| Normal flow of events:   1. Select existing conference 2. Select quantities 3. Click purchase ticket 4. Fill out payment options. 5. Confirm purchase 6. If payment error occurs, return to step 4 7. Return to home screen 8. End. | | |
| Sub flows:   1. Select valid ticket 2. Change date 3. Confirm conference date change 4. If selected modification is not within the allotted modification time range proceed to error screen. 5. Return to user account home page 6. end | | |
| Alternate/exceptional flows:   1. select existing ticket 2. select cancel attendance 3. confirm cancelation 4. if cancelation in not within allotted modification time range proceed to error screen 5. return to home screen 6. end | | |

|  |  |  |
| --- | --- | --- |
| Use case name: Manage Conferences | ID: 001 | Importance Level: 5 |
| Primary actor: Director | Use case type: Overview/essential | |
| Stakeholders and interests:  Director, IT Guru, Vendors, guest speakers, members, participants  Directors have a need to run conferences, IT gurus manage the software problems as necessary. Vendors have a need to sell items and can only be done at a conference, therefore they need the Director to set up and manage the conferences. | | |
| Brief description:  This use case will describe the activities for managing conferences and conference rosters. | | |
| Trigger:  Manager receives validation for the need of a conference and the building and times are in place. The manager then needs to set up the conference date, times, and roster for the new conference. They continue to manage the conference date, times, and roster as necessary until the conference in completed. | | |
| Relationships:  Association:<related> director, IT gurus, vendors, guest speakers, members, participants.  Include: create, modify, cancel, conference  Extend: create, modify, delete, roster  Generalization: | | |
| Normal flow of events:  Number lists of steps <from the use case diagram>   1. Click create conference within the system. 2. Select date 3. Select guest speaker/vendor 4. Select location 5. Select time ranges for each speaker 6. Confirm and save selections 7. If conference conflict occurs return to create conference and correct the conflict at steps 2 through 5. 8. Return to homes screen 9. end | | |
| Sub flows:  Modify conference:   1. select and existing conference within the system 2. select date, speaker/vendor, location, or time range to modify. 3. Make modifications 4. Confirm and save changes 5. If there is a conference conflict, go back to step 2 and resolve conflict 6. Return to home screen 7. End.   Cancel conference:   1. Select existing conference 2. Select reason for cancelation 3. Confirm cancelation 4. Return to home screen 5. End. | | |
| Alternate/exceptional flows:  Conflict return to steps 2 then resolve conflict. Continue to home screen when resolved and confirmed. | | |

|  |  |  |
| --- | --- | --- |
| Use case name: Manage files | ID: 002 | Importance Level: 5 |
| Primary actor: Director | Use case type: Essential | |
| Stakeholders and interests:  Director, IT gurus | | |
| Brief description:  This use case to is describe the activities for uploading, downloading, and deleting files for conferences. | | |
| Trigger: Director has created a conference and needs to upload files for members to access including, brochure. Schedule, and general information. Once the conference is completed | | |
| Relationships:  Association:  Include: upload files, download files, delete files  Extend:  Generalization: | | |
| Normal flow of events:   1. Select existing conference 2. Upload files that pertain to the conference such as, schedules, rosters, brochures, and pamphlets 3. Confirm/save file additions 4. Return to home screen 5. end | | |
| Sub flows:   1. select existing conference 2. delete files that no longer pertain to the conference, or if the conference has been completed. 3. Confirm/save delete changes 4. Return to home screen 5. end | | |
| Alternate/exceptional flows:   1. select existing conference 2. if duplicate files are adding prompt with warning, return to files screen. 3. Make changes as necessary to file naming, 4. Confirm changes 5. end | | |

## Interaction Diagrams

Figure 3:

In the diagram above (figure 3), it shows the communications within the systom as to creating confrences. First the user must login and be an existing administraror (Director/IT Guru). The user selects values for a certain conference to be made, then the values are checked to make sure there are no conflicting inputs. When the user confirms the new conference, it is created and the user gets a message stating the creation was successful.

## Overview Class Diagrams

The image above shows the classes (objects) within the system. The arrowed lines show inheritance. The arrow points to the parent class, from the child class. The numbers with the dots 9in between show the multiplicity.

## Potential Class Domains

The potential class domains we would use if we were to map them are as follows: Person, Conference, and Administration.

# 6. Nonfunctional Requirements

The non-functional requirements (when following furps+) are functionality, usability, reliability, performance, and supportability.

**Functionality:** The system needs to be dynamic in transitioning to more regions (it should be scalable). It also needs to be able to protect against hackers tying to intercept card information, so the card information must be hashed (encrypted).

**Usability:** This system also needs to be intuitive and easy to learn. The aesthetics of the system should be warm and welcoming, and the overall setup of the UI needs to be familiar to the users.

**Reliability:** The system needs to be reliable and stay running as often as possible to ensure that the members and administrators, and so on receive a solid flow of use. The system should also last indefinitely and be able to require less updates upon initial launch.

**Performance:** The system needs to be moderately fast in order to give the users a smooth experience, but the severity of this system needing to be fast is not high. The system needs to be able to hold all the information of what will be added, as well as the information that was previously made outside of the system. The system also need to be scalable for expansion.

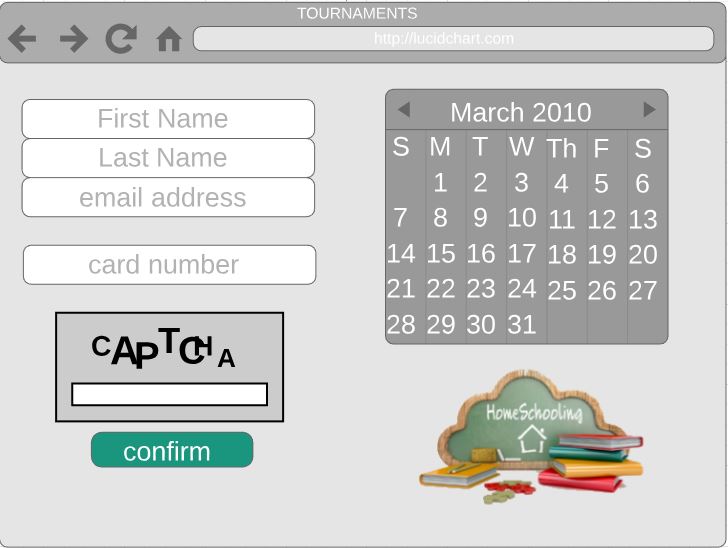
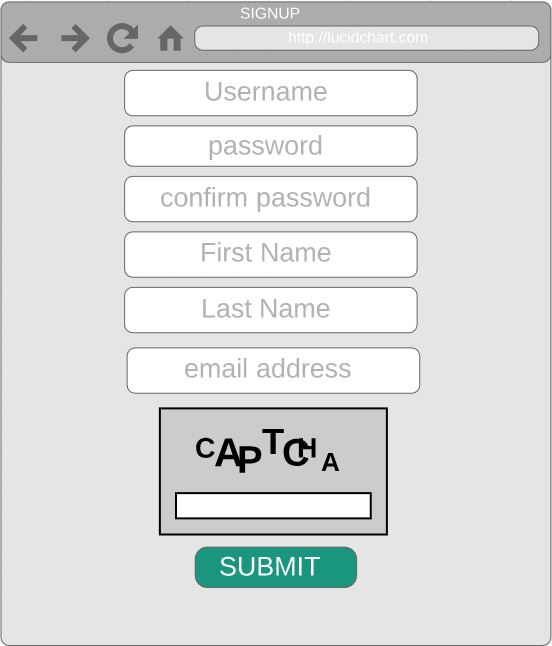
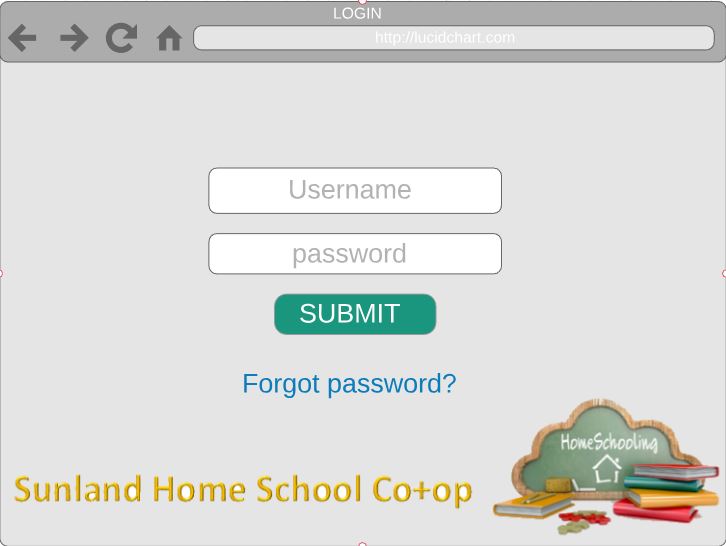
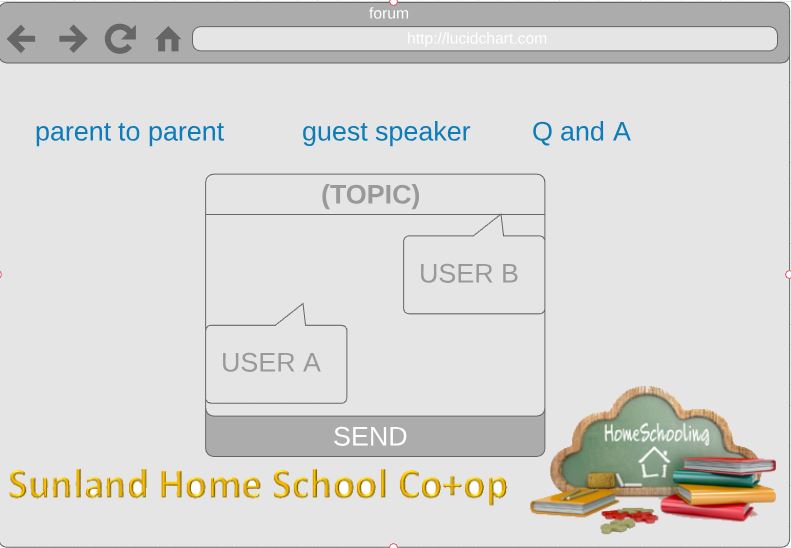
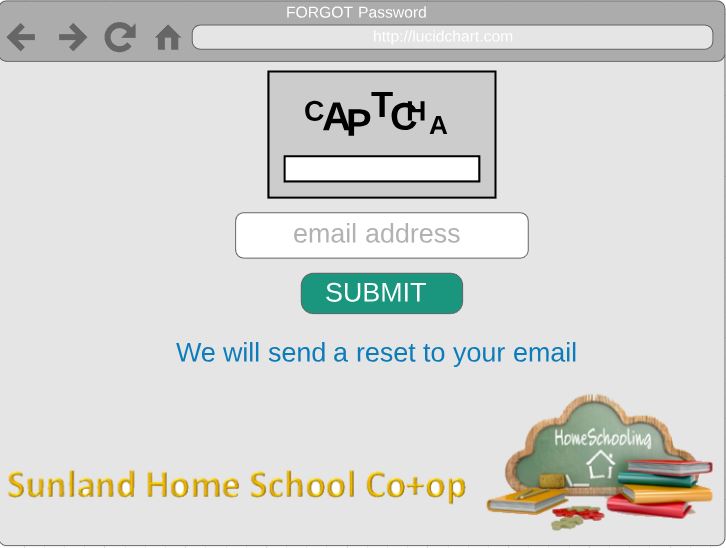
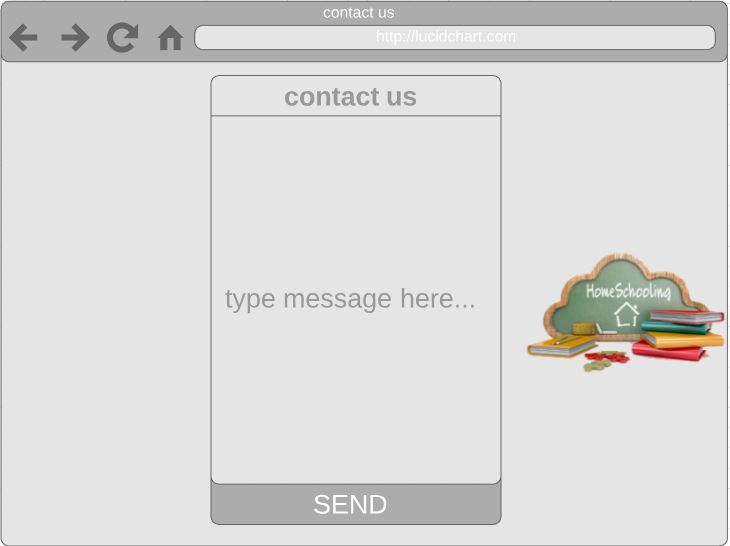
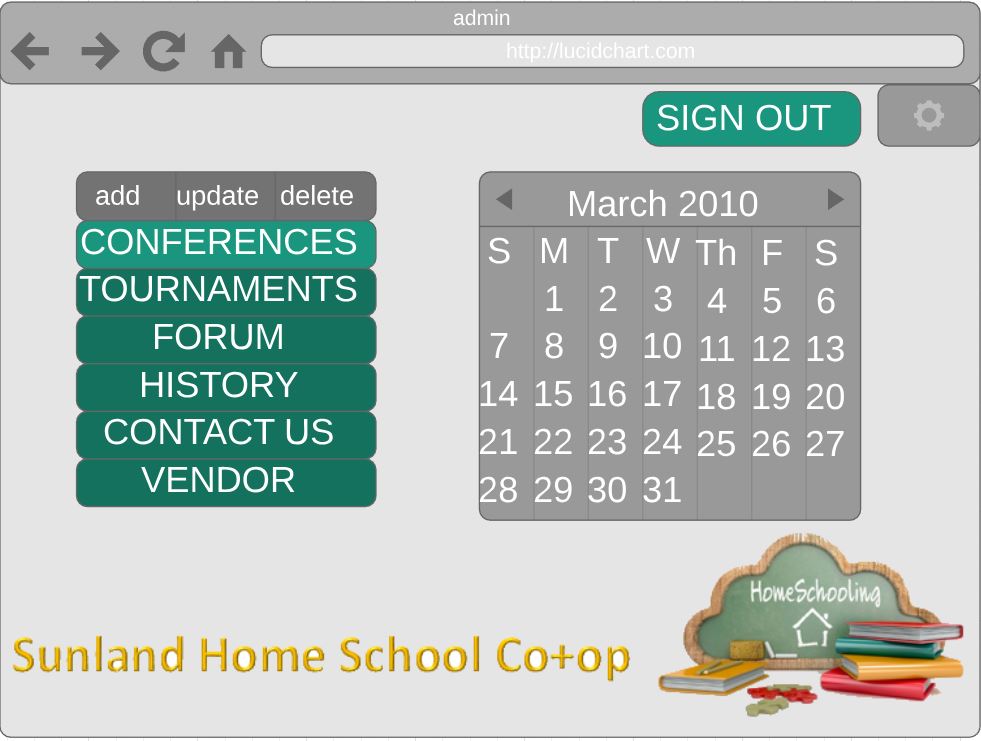
**Supportability:** The system should be simple to maintain, and it needs to be moderately simple to repair when down. It should be sustainable to last indefinitely and keep user data intact.

# 7. References

The references we used to design this system including the cost benefit analysis, the client meetings, and the system request. The system request and the cost benefit analysis can be found in the appendix.

# User Interface

This is a prototype of the System and the users and actors would see it. 



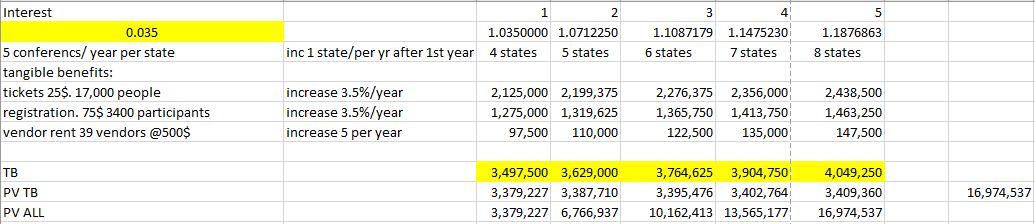
# Effort Breakdown Table

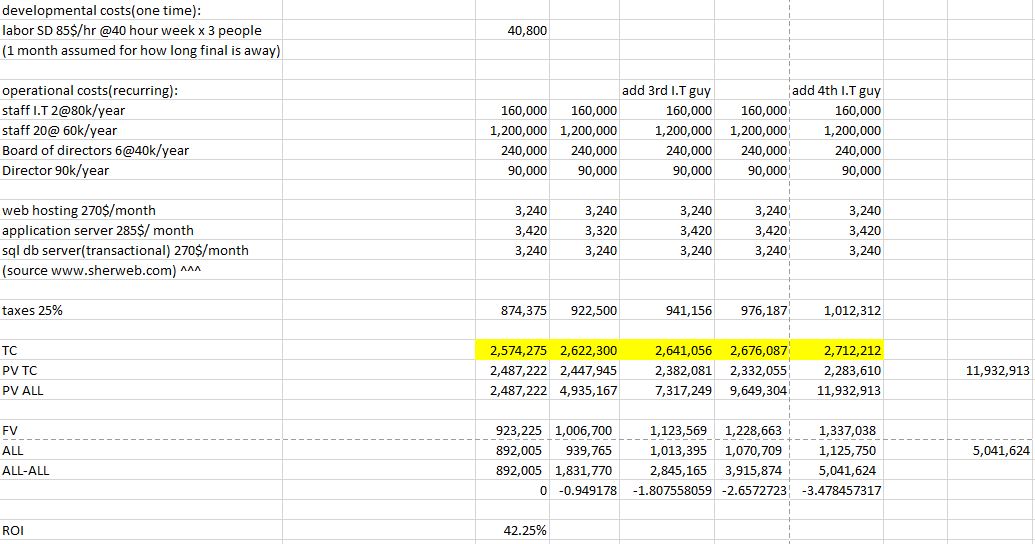
## ***Responsibility Matrix with point breakdown***

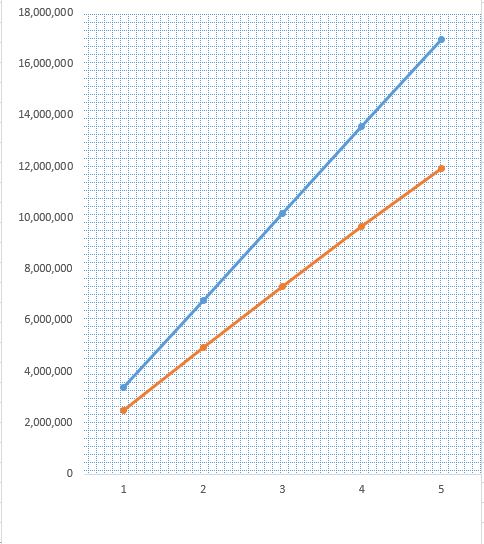
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  | | **Team Member Name:** | | |
|  | **Final Project Tasks** | points | *Skylor* | *Caden* | *Cole* |
|  | **System Request / CBA (40)** |  | 20 | - | 20 |
|  | *System Request* | *20* | - | - | 20 |
|  | *CBA* | *20* | 20 | - | - |
|  | **Presentation/Brochure (80)** |  | 13 | 33 | 33 |
|  | *Presentation* | *40* | 13 | 13 | 13 |
|  | *Brochure* | *40* | - | 20 | 20 |
|  | **SRS (125)** |  | 71 | 0 | 56 |
|  | *Customer Statement of Requirements* | *20* | *20* | *-* | *-* |
|  | *Glossary of Terms* | *2* | *-* | *-* | *2* |
| Functional Requirements Specification (61) | *Stake holder* | *2* | *-* | *-* | *2* |
| *Actors and Goals* | *4* | *4* | *-* | *-* |
| *Use case diagrams* | *10* | *-* | *-* | *10* |
| *Use case casual description* | *10* | *10* | *-* | *-* |
| *Use case full description* | *10* | *-* | *-* | *10* |
| *Activity diagrams* | *10* | *-* | *-* | *10* |
| *Interaction diagrams* | *10* | *-* | *-* | *10* |
| *Overview Class diagram* | *5* | *-* | *-* | *5* |
|  | *Non-funct. Requirements* | *5* | *-* | *-* | *5* |
|  | *References* | *2* | *2* | *-* | *-* |
|  | *User Interface design* | *35* | *35* | *-* | *-* |
|  | **Team Member Total** | | 104 | 33 | 107 |

# Appendix

**COST BENEFIT ANALYSIS**







SYSTEM REQUEST

Sponsor:

Alison Sunderland (Director of Sunland Home School Co+op)

Business Needs:

•To provide parents a centralized resource that connects to the home schooling community.

•To schedule conferences.

•To allow conference attendees to purchase tickets for the conferences.

•To have a mobile friendly app that helps buy tickets for conferences and see conference schedules.

•To keep scheduled events organized, and easily updatable.

•To let the conferences schedule be visible to all the participants to keep them all in the loop.

•To bring more people into the homeschool community

Business Requirements:

•Needs to keep track of each conferences schedule.

•Needs to designate the tickets to specific conferences.

•The guest speakers need a way to set up their conference.

•Needs the ability to watch the conference through the mobile app.

•Has to have the ability for tournament registrations.

•The mobile app needs to be intuitive.

•The employees need to learn how to access and use the new system.

•The transfer of all the current information to the new software.

•Server costs.