Basement Dwellers

KAPP Application
Use-Case-Realization Specification

Version <1.3>

Revision History

Date	Version	Description	Author
<10/23/2022>	<1.0>	<Major Revisions> We completely changed our design and the template we used to fit more in-line with other projects.	<Troy, Tanner, Adam, Chris, Thomas>
<10/25/2022>	<1.1>	<Continuing Revisions> Began transferring out project to Github, which we decided would allow us to collaborate easier. We are also continuing to make improvements to our design documents.	<Troy, Tanner, Adam, Chris, Thomas>
<10/30/2022>	<1.2>	<Continuing Revisions>	<Troy, Tanner, Adam, Chris, Thomas>
<11/6/2022>	<1.3>	Updated headings	<Tanner>

Table of Contents

1. Introduction **6**

1.1 Purpose 6

1.2 Scope 6

1.3 Definitions, Acronyms, and Abbreviations 6

1.4 References 6

1.5 Overview 6

2. <Single Sign On> **6**

2.1 Brief Description 6

Allows the app to verify users using their KU credentials. 6

2.2 Flow of Events - Design 6

2.3 Interaction Diagrams 6

2.3.1 Sequence Diagrams 6

2.3.2 Participating Objects 7

2.4 Class Diagrams 7

3. <Login> **7**

3.1 Brief Description 7

3.2 Flow of Events - Design 7

3.3.1 Sequence Diagrams 8

3.3.2 Participating Objects 8

3.4 Class Diagrams 8

3.5 Derived Requirements 8

4. <KU Database> **8**

4.1 Brief Description 8

4.2 Flow of Events - Design 8

4.3.1 Sequence Diagrams 9

4.3.2 Participating Objects 9

4.4 Class Diagrams 9

4.5 Derived Requirements 9

5. <Tab> **9**

5.1 Brief Description 9

5.2 Flow of Events - Design 9

5.3.1 Sequence Diagrams 10

5.3.2 Participating Objects 10

5.4 Class Diagrams 10
5.5 Derived Requirements 10

6. <Homepage> 10
6.1 Brief Description 10
6.2 Flow of Events - Design 10
6.3.1 Sequence Diagrams 11
6.3.2 Participating objects 11
6.4 Class Diagrams 11
6.5 Derived Requirements 11

7. <Interactive Map> 11
7.1 Brief Description 11
7.2 Flow of Events - Design 11
7.3.1 Sequence Diagrams 12
7.3.2 Participating Objects 12
7.4 Class Diagrams 12
7.5 Derived Requirements 12

8. <Events> 12
8.1 Brief Description 12
8.2 Flow of Events - Design 12
8.3.1 Sequence Diagrams 13
8.3.2 Participating Objects 13
8.4 Class Diagrams 13
8.5 Derived Requirements 13

9. <GPA Calculator> 13
9.1 Brief Description 13
9.2 Flow of Events - Design 14
9.3.1 Sequence Diagrams 14
9.3.2 Participating Objects 14
9.4 Class Diagrams 15
9.5 Derived Requirements 15

10. <Schedule Builder> 15
10.1 Brief Description 15
10.2 Flow of Events - Design 15
10.3.1 Sequence Diagrams 15

10.3.2 Participating Objects 15
10.4 Class Diagrams 16
10.5 Derived Requirements 16

Use-Case-Realization Specification

1. Introduction

1.1 Purpose

This document provides a complex overview of the system, using multiple diagrams to represent different components of the system.

1.2 Scope

KAPP allows students to be able to access an interface with multiple different features relating to the university. KAPP will focus on providing an interactive map for students to view on campus eventsThis Use-Case-Realization Specification will provide an overview of the use cases developed in the KAPP system.

1.3 Definitions, Acronyms, and Abbreviations

See Software Architecture Document

1.4 References

1.5 Overview

The rest of this document gives a general overview of the specific use cases of KAPP, its software system, and the use cases

2. <Single Sign On>

2.1 Brief Description

Allows the app to verify users using their KU credentials.

2.2 Flow of Events - Design

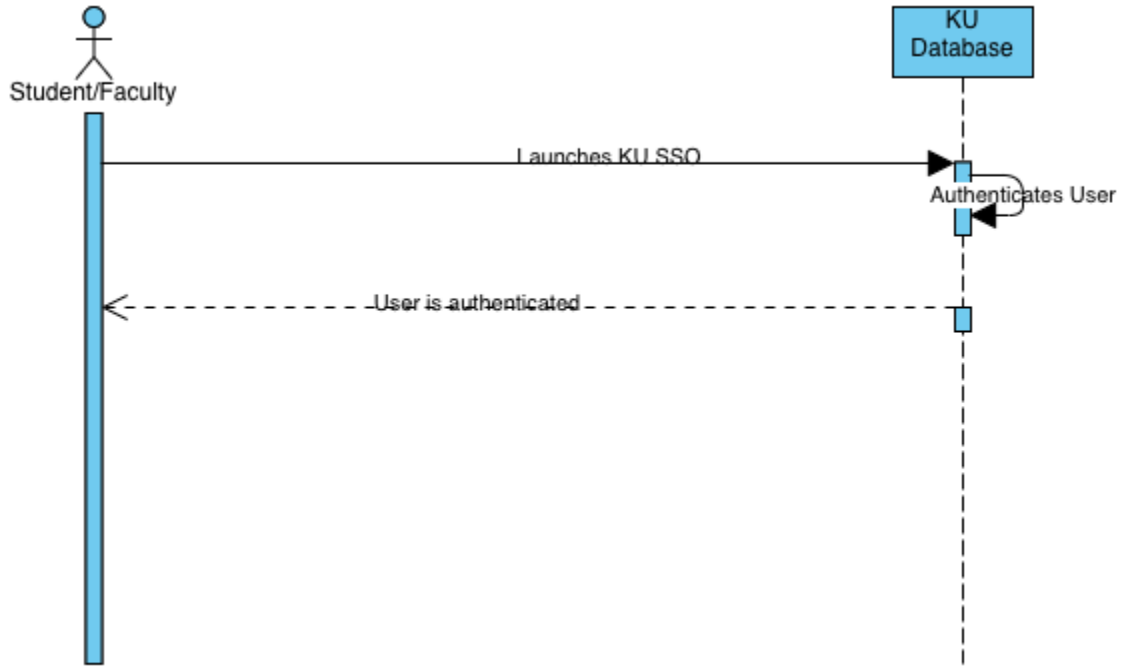
When requesting to log into KAPP a request is sent to the KU database in order to be verified to be logged in.

2.3 Interaction Diagrams

-The Faculty or Student launches the Single Sign On (SSO)

-The application received the request and redirects to the KU database

2.3.1 Sequence Diagrams



2.3.2 Participating Objects

Object	Description
KU Database	This object verifies the user's login credentials using the KU database and authenticates the user.

2.4 Class Diagrams

See Software Architecture Document

2.5 Derived Requirements

Using single sign on means that users must have credentials in the system already.

3. <Login>

3.1 Brief Description

Users will enter their login credentials to be verified via Single Sign On.

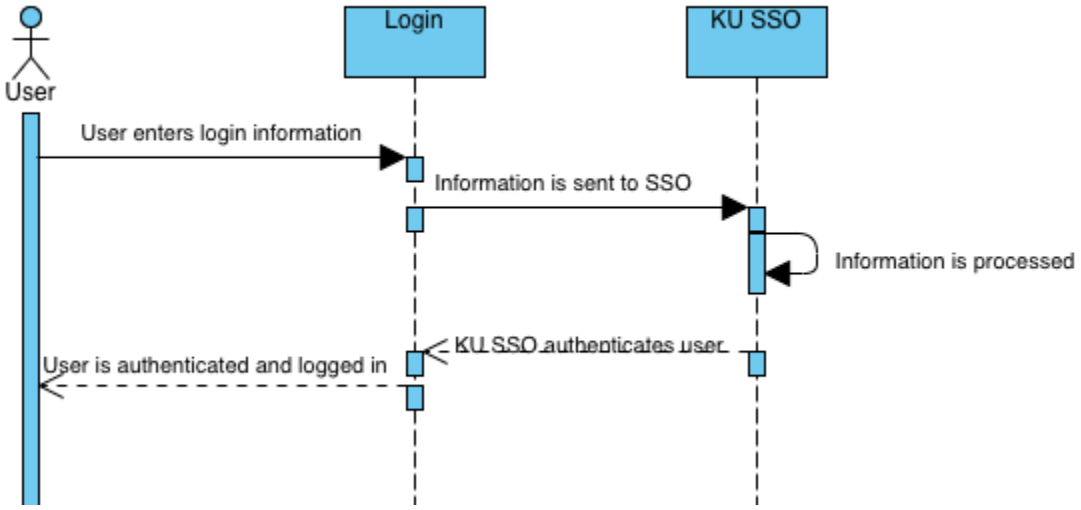
3.2 Flow of Events - Design

This section will be displayed whenever a user is not logged into their account. It will prompt the user to enter their login details and then hand this data off to the single sign on.

3.3 Interaction Diagrams

- The user enters login information to be processed
- The login information is sent to KU SSO
- It is then processed and returned if succesful
- User is then able to access KAPP

3.3.1 Sequence Diagrams



3.3.2 Participating Objects

Object	Description
Login	This object collects the user's login information and sends it to KU SSO. It will also authorize the user upon successful authentication.
KU SSO	This object verifies the user's login credentials using the KU database and authenticates the user.

3.4 Class Diagrams

See Software Architecture Document

3.5 Derived Requirements

The user will have an existing account with SSO.

4. <KU Database>

4.1 Brief Description

This is the database that is checked when login users in. It is maintained by KU, not KAPP.

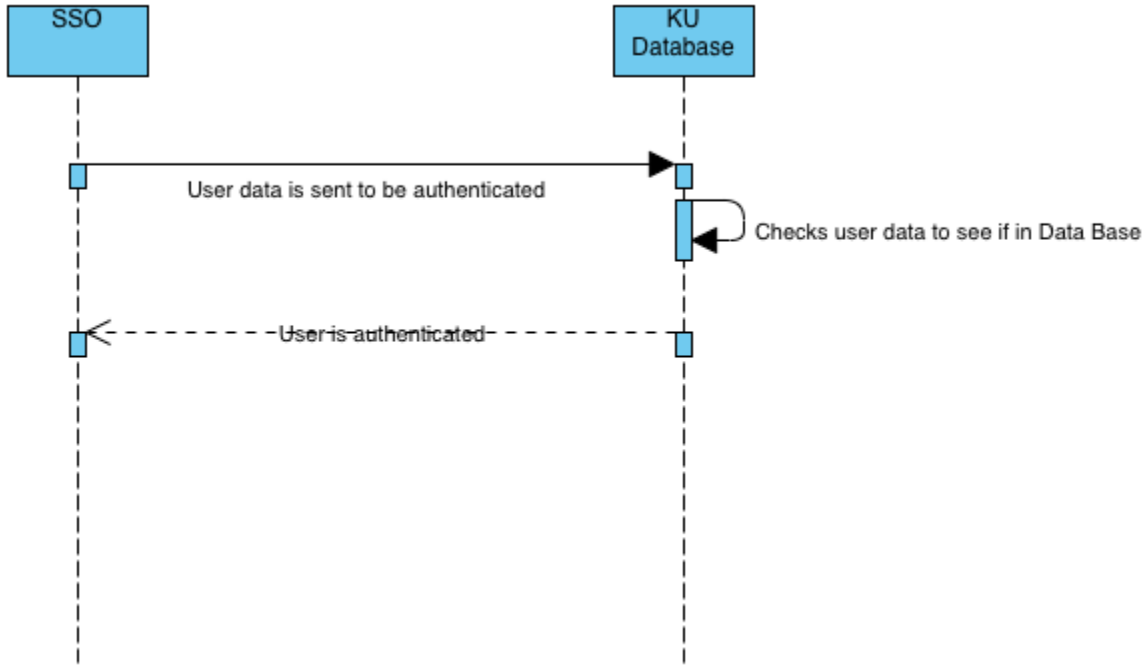
4.2 Flow of Events - Design

This is an integration into KU's database of all their users. It is funneled through the single sign on.

4.3 Interaction Diagrams

- The SSO receives login information from KAPP
- The data is sent to the KU database to be authenticated
- The information is processed and returned if succesful
- The SSO is authenticated and will return a message to KAPP

4.3.1 Sequence Diagrams



4.3.2 Participating Objects

Object	Description
KU Database	This object represents the KU database which contains login information. It will return if the user was authenticated.

4.4 Class Diagrams

See Software Architecture Document

4.5 Derived Requirements

KU continues to use and maintain this database.

5. <Tab>

5.1 Brief Description

This is a tab that is displayed as one of the main view of the app. It is accessed via the tab container at the bottom of the app.

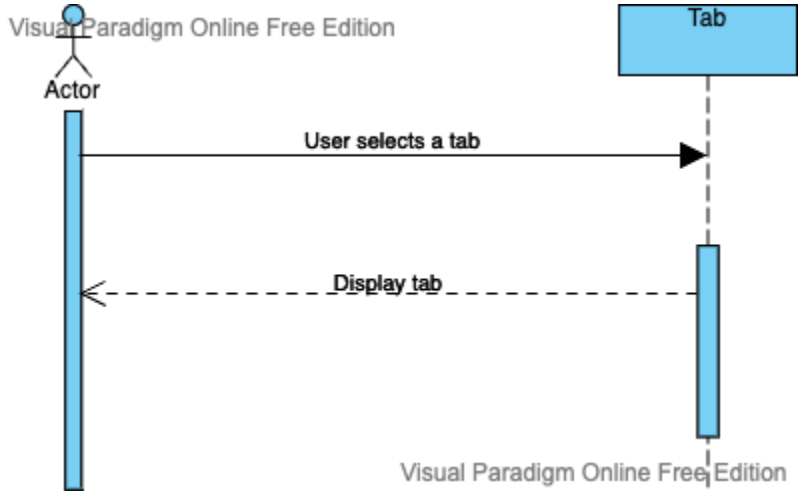
5.2 Flow of Events - Design

A tab is a page of our application. They are accessed via the tab container at the footing of the app display, and are where most functions of the app are performed.

5.3 Interaction Diagrams

- The user, now on the homepage, is able to select a tab from tab container
- If the user selects a tab, then the user will be sent to a new page based on the tab selected

5.3.1 Sequence Diagrams



5.3.2 Participating Objects

Object	Description
Tab	This object contains display information for each tab displayed to the user.

5.4 Class Diagrams

See Software Architecture Document

5.5 Derived Requirements

None

6. <Homepage>

6.1 Brief Description

This is main landing page for all users of KAPP. It will display general information to help guide the user to their correct endpoint.

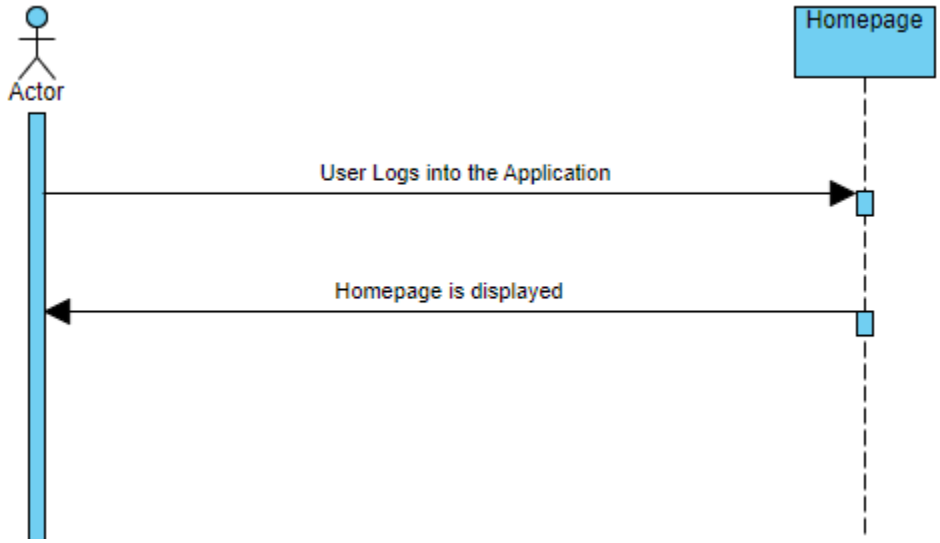
6.2 Flow of Events - Design

Once the user logs into the application, the homepage is displayed.

6.3 Interaction Diagrams

- Once the user is able to be authenticated with the login page, then they will be redirected to homepage
- The homepage contains a header, as well as the tab container for the given tabs.
- The user can then access any tabs once on the homepage

6.3.1 *Sequence Diagrams*



6.3.2 *Participating objects*

Object	Description
Homepage	This object contains the homepage which will be displayed to the user

6.4 Class Diagrams

See Software Architecture Document

6.5 Derived Requirements

None

7. <Interactive Map>

7.1 Brief Description

This is a map element that the user can interact with. It displays KU campus, and relevant information on buildings and events.

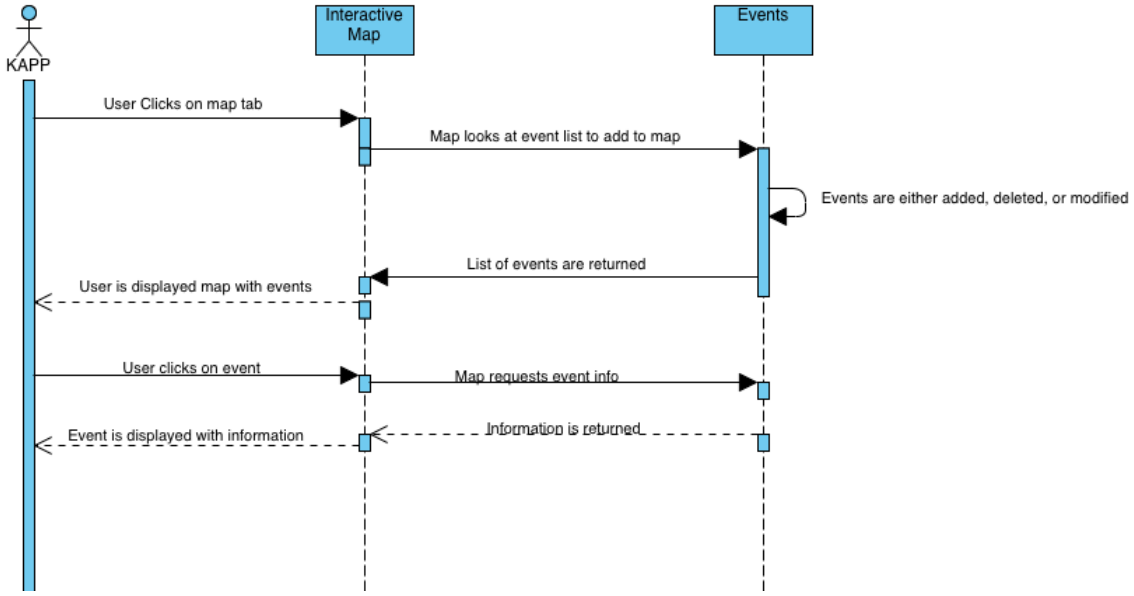
7.2 Flow of Events - Design

Once the user clicks on the map they are able to view on campus events through React JS Mapview API.

7.3 Interaction Diagrams

- The user accesses the interactive map from the tab container on the homepage
- The map accesses the event list to display events across campus
- The list of events are returned to the map
- The user is able to move around campus on the map and click on pop-up events
- Each event will then have corresponding information

7.3.1 Sequence Diagrams



7.3.2 Participating Objects

Object	Description
Interactive Map	This object will display an interactive map to the user. The user can interact by clicking on events within the map.
Events	This object contains upcoming events which will be displayed to the user and displayed on the map.

7.4 Class Diagrams

See Software Architecture Document

7.5 Derived Requirements

User's device has sufficient hardware to handle the graphical processing.

8. <Events>

8.1 Brief Description

This is a view that displays all of the events going on at KU campus in one spot.

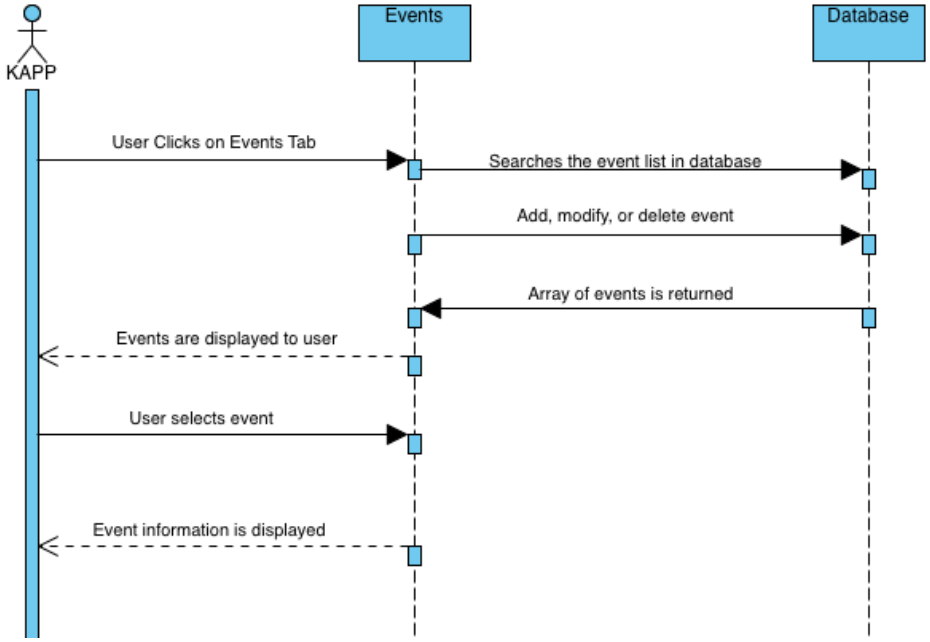
8.2 Flow of Events - Design

The user is able to navigate the events tab that will have a list of verified events across campus. Each event will be able to be modified by club admins, administrators, etc. The goal is to provide easy access to events across campus with descriptions for each.

8.3 Interaction Diagrams

- User accesses the events tab from the homepage
- From there, events are able to be added, modified, or deleted.
- A list of events is then returned to user.
- Each event will have information that is available in event tab as well as the map

8.3.1 Sequence Diagrams



8.3.2 Participating Objects

Object	Description
Events	This object contains data for upcoming events. It will display this information to the user
Database	This object contains upcoming events. It sends these events to the Events object to display to the user.

8.4 Class Diagrams

See Software Architecture Document

8.5 Derived Requirements

There must be some database of events that is populated before this can be public.

9. <GPA Calculator>

9.1 Brief Description

This is a simple information display unit. It tells the user their GPA and credit hours.

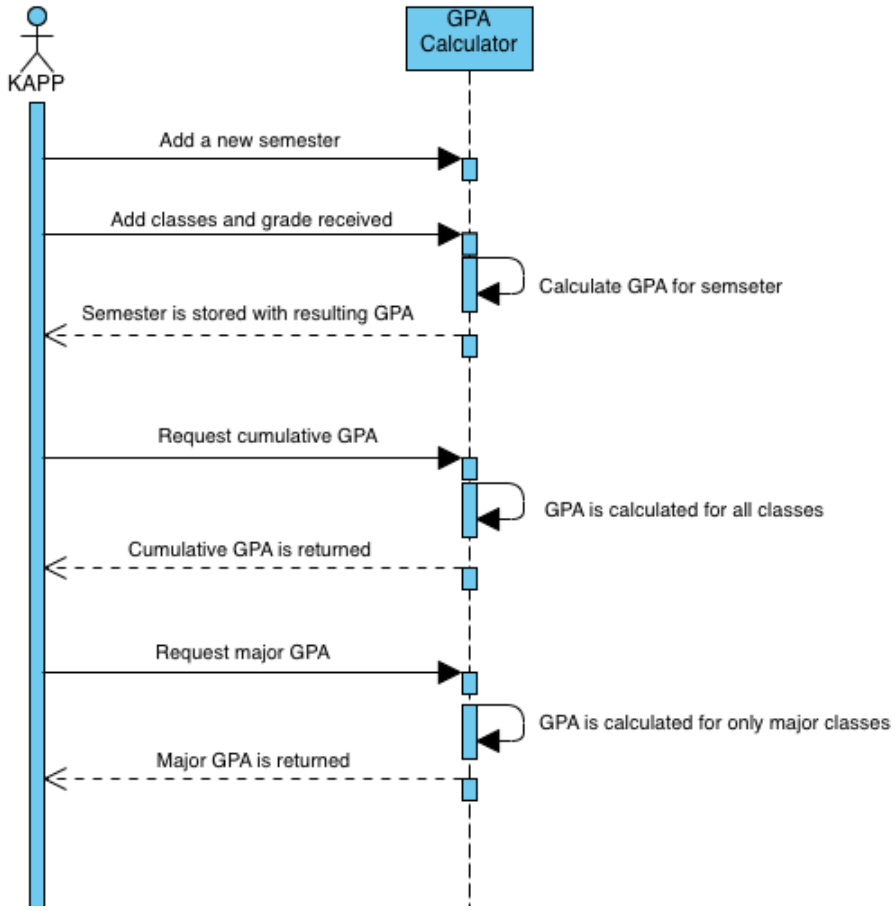
9.2 Flow of Events - Design

The user can access the GPA calculator through the homepage in the tab container. Once in the tab, the user is able calculate GPAs semester by semester. This is all done locally in the GPA Calculator tab, and does not require accessing a server.

9.3 Interaction Diagrams

- The user can add a new semester to record classes and corresponding grades.
- Once all the classes are added for a given semester, then the GPA is calculated for that semester
- This is then returned to the user.
- Once all semesters are added, the user can choose to calculate cumulative or major GPA.

9.3.1 Sequence Diagrams



9.3.2 Participating Objects

Object	Description
GPA Calculator	This object contains semesters of courses. Each course in the semester contains a grade. These grades will be used to calculate the Student's Cumulative and Major GPA.

9.4 Class Diagrams

See Software Architecture Document

9.5 Derived Requirements

User must have atleast 1 credit hour taken and at least 1 course with a grade posted.

10. <Schedule Builder>

10.1 Brief Description

This view allows the user to build their schedule and make sure it will line up with timings of other classes. It mimics the use of the schedule builder on myKU.

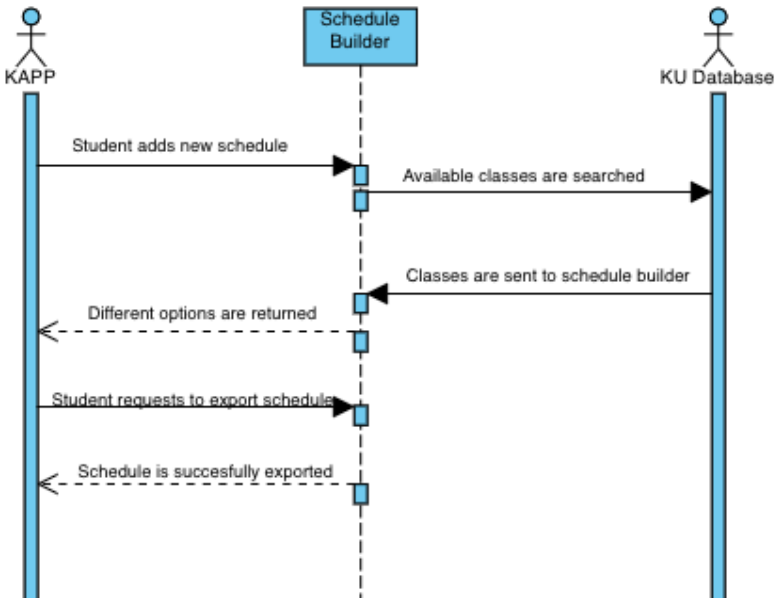
10.2 Flow of Events - Design

The user is able to access the schedule builder through the homepage. Once in the schedule builder, the user is able to create a schedule for upcoming semester. This is done through accessing KU database and returning a number of different options. The user then has the option, once the schedule is finalized, to export the schedule to their personal calendar.

10.3 Interaction Diagrams

- The user accesses the Schedule Builder through the homepage
- The user selects the option to add a new schedule
- The possible classes are searched through the KU database
- Class information is then sent back to the schedule builder to be processed.
- The user is able to then export the schedule once it is selected

10.3.1 Sequence Diagrams



10.3.2 Participating Objects

Object	Description
Schedule builder	This object contains the classes which the student added

	to their schedule. This object will also communicate with KU Database to get a list of classes for the user to pick from.

10.4 Class Diagrams

See Software Architecture Document

10.5 Derived Requirements

Next semester courses and schedules must be posted in order to use this.